MOBILE PRIVACY LEAKAGE DETECTION AND PREVENTION: FROM TECHNICAL SOLUTIONS TO USER EXPERIENCE

SAEED ALQAHTANI

A THESIS
SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE
FROM THE
UNIVERSITY OF SURREY

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
UNIVERSITY OF SURREY
GUILDFORD, SURREY GU2 7XH
UNITED KINGDOM

SUPERVISORS:
PRINCIPAL SUPERVISOR: PROF LIQUN CHEN
CO-SUPERVISOR: PROF SHUJUN LI
CO-SUPERVISOR: DR IOANA BOUREANU

SEPTEMBER 2018
© Saeed Alqahtani 2018
Declaration of Originality

This thesis and the work to which it refers are the results of my own efforts. Any ideas, data, images or text resulting from the work of others (whether published or unpublished) are fully identified as such within the work and attributed to their originator in the text, bibliography or in footnotes. This thesis has not been submitted in whole or in part for any other academic degree or professional qualification. I agree that the University has the right to submit my work to the plagiarism detection service TurnitinUK for originality checks. Whether or not drafts have been so-assessed, the University reserves the right to require an electronic version of the final document (as submitted) for assessment as above.

September 2018
Saeed Alqahtani
Summary

Mobile apps feed on variety of users’ information to provide great services. Some of the features require more sensitive details such as contact list to connect with friends or a precise location to find a desired restaurant nearby. Handling personal information is vital because the user would expect them to be processed in an appropriate manner. However, research has proven that some third-party apps accidentally or maliciously leak users’ personal details. Thus, researchers made huge effort to come up with tools that detect leakage attempts. In this thesis, we targeted several gaps to improve user experience with mobile privacy leakage problem. Initially, we designed a system that evaluates mobile privacy protection tools. This system can be useful for developers to assess their tools, and users to evaluate offered solutions. 165 selected Android privacy protection apps have been tested using our system, and it was established that the most effective approach of mobile privacy protection requires modification on mobile operating system level in order to capture explicit and implicit leakages. That requirement makes it difficult to find “off-the-shelf” protection tool. Therefore, it was decided to assist the user in selecting safe apps as a precaution step before problems occur. To achieve that, it is important to understand how mobile users form their decision when selecting apps. 1,100 crowdsourcing participants have been recruited to study their perceived trust of subjective and objective ratings of mobile apps’ privacy. This experiment guided us to design new interfaces that could assist decision making towards more privacy-friendly mobile apps, which was our most recent work. A newly designed interface, which communicates objective privacy ratings to the user, has been proposed. We have also conducted several user-studies involving 300 participants to evaluate our proposed app’s efficiency, the result ultimately showed that users were more motivated to engage in privacy-related decisions.
Acknowledgements

First and foremost, I want to thank Allah, the Almighty, for giving me the strength, peace of mind and good health to accomplish this research study.

I would like to express my deepest appreciation to my supervisor, Prof. Shujun Li, who has been my real mentor, guide, and even a friend throughout my PhD journey. He committed supporting me and supervising my work to the end, although he left the University of Surrey. I really cannot thank him enough for all what he has done.

I also would like to thank my sponsor; Taibah University, Saudi Arabia, for giving me the opportunity to achieve this dream.

I am also deeply grateful for my other collaborators, Dr. Nouf Aljaffan and Dr. Haiyue Yuan for their support, constructive discussions and help.

I would like also to express my gratitude to all the great researchers who contributed fruitful ideas, materials or discussions to this work. Namely they are: Prof. Jason Hong (Carnegie Mellon University, USA), Markus Olausson & Ola Johnasson (alternativeTo founders, Sweden), Prof. Mohamed Ali Kaafar (CSIRO Data61, Australia), Dr. Panagiotis Andriotis (University of the West of England, UK), Dr. Patrice Rusconi & Dr. Sebastian E Bartos (School of Psychology, University of Surrey, UK), Dr. Abdulaziz Alanazi (Department of Physics, University of Surrey, UK) and Mr. Saleh Jalbi (Department of Civil and Environmental Engineering, University of Surrey, UK).

Last but not least, I will not forget my supervisor Prof. Liqun Chen who supervised me during my fourth year. I am very thankful for her support, time and advice during the last year of my PhD.
Dedication

To my loving
Mother & Father,
Wife,
and Siblings...

whose affection, endless love, encouragements and prayers of days and nights

make me able to get success and honour.
List of Publications

Research Papers


2. N. Aljaffan, **S. Alqahtani**, H. Yuan, S. Li, P. Rusconi and S. Bartos, “Subjective and Objective Ratings of Password Strength: What Do Users Trust More?,” *under preparation for re-submission*, former versions submitted to several venues but were not successful (with encouraging comments from reviewers).


Contents

Declaration iii
Summary v
Acknowledgements vii
Dedication ix
List of Publications xi
Contents xii
List of Figures xvi
List of Tables xviii

1 Introduction 1
  1.1 Background .............................................. 1
  1.2 Objectives and Hypotheses .............................. 2
  1.3 Contributions .......................................... 4
  1.4 Structure .............................................. 4

2 Literature Review 7
  2.1 Privacy: Overview ....................................... 7
    2.1.1 Privacy vs. Security ............................... 8
    2.1.2 Legal Aspect of Privacy ........................... 10
    2.1.3 Privacy as a Human Right ......................... 12
  2.2 User Privacy on Computers ............................. 13
    2.2.1 Data-driven Approach in Classifying Privacy Issues .... 14
  2.3 Privacy Protection Techniques ........................ 15
    2.3.1 Detective Approaches .............................. 16
    2.3.2 Protective Approaches ............................ 17
  2.4 Theoretical Privacy Concepts .......................... 19
5 PAltRoid

5.1 Introduction ......................................................... 87
5.2 Related Work ...................................................... 88
  5.2.1 Improving Privacy Awareness of Users of Mobile Devices .... 88
  5.2.2 Privacy Scoring of Android Apps .............................. 89
  5.2.3 Alternative Apps Identification ............................... 90
5.3 Design ................................................................. 91
  5.3.1 Objectives ...................................................... 91
  5.3.2 Hypotheses ..................................................... 92
  5.3.3 PAltRoid Overall Design ...................................... 92
  5.3.4 PAltRoid Components ......................................... 93
  5.3.5 Data Used .................................................... 95
  5.3.6 Collected Data ................................................ 95
5.4 Experimental Studies ............................................... 96
  5.4.1 Ethical approval ................................................ 96
  5.4.2 Study Groups .................................................. 96
  5.4.3 Lab-Based Study .............................................. 96
  5.4.4 Crowdsourcing-based Study ................................... 98
5.5 Results & Analysis ................................................ 100
  5.5.1 Participants ................................................... 100
  5.5.2 App Ratings .................................................. 101
  5.5.3 Lab-based Results ............................................ 102
  5.5.4 Privacy-Related Decisions ................................... 103
  5.5.5 Analysis of Participants’ Subjective Feedback ................ 106
5.6 Discussions ......................................................... 109
5.7 Future Work ........................................................ 111
  5.7.1 Improvement on Current Design ............................... 112
  5.7.2 New Ideas for Different Design Interfaces .................... 112
5.8 Conclusion .......................................................... 112

6 Conclusion and Future Work ....................................... 115
  6.1 Overview .......................................................... 115
  6.2 Selected Future Work ............................................ 116
6.2.1 Improving PPAndroid-Benchmarker . . . . . . . . . . . . . . 116
6.2.2 Users Perceived Trust of Mobile Apps’ Privacy Ratings: More
User Studies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
6.2.3 Extending PAItrROID . . . . . . . . . . . . . . . . . . . . . . . 117
6.2.4 Wider Directions . . . . . . . . . . . . . . . . . . . . . . . . . . 117

A Participant Information Sheet 121
B Interview Schedule 131
C Lab-Based Study Questionnaires 133
D Advertising Poster 143
E Electronic Survey Design 145
Bibliography 149
List of Figures

1.1 The stages of this PhD study ............................................. 3
2.1 Overlaps between security and privacy [1] ............................ 9
2.2 Reported problem in Skype client [2] ................................. 24
2.3 Trackers collecting personal information [3] ......................... 26
2.4 Android overall architecture [4] ....................................... 31
2.5 TaintDroid architecture [5] ............................................. 33

3.1 PPAndroid-Benchmarker’s overall design .......................... 40
3.2 PPAndroid-Benchmarker’s components and data flow map ........ 41
3.3 PPAndroid-Benchmarker in probing phase ......................... 48

4.1 A screen-shot of “anonymous” and displayed app questions and rating options .................................................. 62
4.2 Age (a) and computer skill levels (b) distribution of participants based on their gender ............................................ 67
4.3 3-D histogram of participants’ collective behaviours in terms of their choices on su and ob for plain apps ......................... 68
4.4 3-D histogram of participants’ collective behaviours in terms of their choices on su and ob for tool apps ....................... 68
4.5 3-D histogram of participants’ collective behaviours in terms of their choices on su and ob for social apps ................................ 69
4.6 2-D histograms of participant’s collective behaviours in terms of their choices on su and ob ........................................... 71
4.7 2-D distribution of participants’ responses according to their behavioural pattern in each app category condition ............. 74
4.8 3-D histograms of collective behaviours for participant’s less than 26 years ............................................................ 76
4.9 3-D histograms of collective behaviours for participant’s between 26 to 35 years ....................................................... 77
4.10 3-D histograms of collective behaviours for participant’s over 35 years .............................................................. 78
4.11 Percentages of users’ reasons behind their trust on app ratings .......... 81
5.1 PAltRoid’s overall design ........................................... 92
5.2 Screen shots of PAltRoid .......................................... 94
5.3 First time instruction fragment .................................. 99
5.4 Number of apps with different privacy scores in Groups 1 and 2 .... 102
5.5 Participant-specific number of rated apps in Groups 1 and 2 ........ 102
5.6 Average percentage of uninstalled apps for each privacy score and for Groups 1 and 2 ........................................... 103
5.7 Emotion scores for Groups 1 and 2 ............................... 108
List of Tables

2.1 Some statistics of detected leakages by mobile privacy tools [6] 25
2.2 Mobile platform usage statistics [7] 30

3.1 Testing installation-time privacy apps 49
3.2 Testing privacy apps that require root access 50
3.3 Probing apps requirement of user-intervention 51
3.4 Testing TaintDroid against PPAndroid-Benchmarker 52

4.1 Apps names used in our experiments and their privacy ratings, shown in 4-point scale 61
4.2 Predefined answers for users’ reasons behind each choice of rating 63
4.3 Results of the Stuart-Maxwell $\chi^2$ tests for analysing participants’ self-reported trust in subjective and objective ratings 70
4.4 Results of the multinomial logistic regressions conducted on the app category condition as the predictor of participants’ self-reported trust 70
4.5 Results of the multinomial logistic regression conducted on the behavioural pattern as the predictor of participants’ self-reported trust 72
4.6 Results of a multinomial logistic regression conducted on two app data subsets as the predictor of participants’ self-reported trust 73
4.7 Results of the multinomial logistic regression conducted on the age factor as the predictor of participants’ self-reported trust 75
4.8 Results of the Stuart-Maxwell $\chi^2$ tests on participants’ perception of subjective and objective app ratings for different app sessions 79
4.9 Results of the Stuart-Maxwell $\chi^2$ tests on participants’ perception of subjective and objective app ratings for different Tool app pairs 80
4.10 Results of the Stuart-Maxwell $\chi^2$ tests on participants’ perception of subjective and objective app ratings for different Social app pairs 80

5.1 App uninstallation statistics in Groups 1 and 2, assuming PE decisions are all valid 104
5.2 Two sample t-test results on uninstalled apps with different privacy scores, for Groups 1 and 2 105
5.3 $\chi^2$ test results for analysing participants’ app uninstallation behaviours 105
5.4 Wilcoxon rank sum test results on uninstalled apps with different privacy scores, for Groups 1 and 2 106
5.5 Participants’ response to the question ‘How would you assess the design of the tested app?’ 107
Chapter 1

Introduction

1.1 Background

Mobile devices ownership has increased rapidly over the recent years. The wide range of mobile devices’ possession has been accompanied by the rise of Google Play and App Store. According to Statista, the statistics portal [8], Android users can select from more than 3.8 million apps while iOS users are offered around 2 million apps, as of the first quarter in 2018. Mobile users can enjoy a variety of apps, which provide useful services and features by making use of smart devices’ capabilities. The existence of mobile apps in devices is vital for better mobile device experience. Clearly, users increasingly rely on mobile devices and apps. Mobile apps today are growingly more advanced, powerful, life-engaging and sometimes more privacy-intrusive.

Undoubtedly, advance in mobile technologies opens the door for new privacy issues. As mobile apps are getting more and more popular among users, the privacy of them also becomes more and more a concern. In some cases, malicious identities may exploit users’ data to steal or uncover personal information about them for illegal purposes. For instance, it was reported that some apps used stolen personal details and secretly made calls and sent text messages, which caused financial losses to victims [9]. Moreover, data can be used by companies to identify personal information about users without their consent which violates the law, not to forget these incidents where big companies, such as Facebook and Snapchat, had security breaches in the past [10,11]. Such security incidents could endanger users’ sensitive data. In order to detect privacy intrusive apps, mobile app stores and security software vendors have made some good efforts to develop and deploy different solutions (e.g., the permission granting mechanism of the Android OS), but comparing with security products there are relatively less advanced tools for privacy protection. Therefore, the ball is often in the user’s court to make the right decision to identify and not to use privacy intrusive apps.
To provide users with better privacy while using their mobile devices, the research community has proposed many solutions to overcome privacy obstacles. Many protection mechanisms and detective approaches have been presented to prevent mobile privacy leakages. For detective techniques the main purpose is to identify potential privacy violations as they occur and respond with corrective suggestions while preventive methods try to stop violating privacy at first place. One basic technical approach followed by many privacy leakage detection tools is to spot unauthorised or suspicious flows between sensitive information sources and sinks, where a ‘sink’ is an interface allows information to go out of the mobile device [5]. Nevertheless, many of these tools use a limited and static list of information sources leaving other sources exposed to leakages [12]. Also, there is some limited work on automatic discovery of sources. In addition, almost all mobile privacy leakage detection tools simply produce a notification for every detected leakage and leave further actions to end users who often feel annoyed or confused on how to respond [13].

For the stated concerns, a necessity of providing a way to evaluate current provided solutions is identified. Therefore, 1) a benchmarking system is needed. Such system could help in evaluating available privacy protection tools to help developers as well as users. Moreover, 2) The human perception of different app ratings need to be studied. That will help to understand how users form their decisions, and therefore 3) come out with best approaches to engage users more in privacy-related decisions. In this PhD work, the focus is to fill the gap of the above-mentioned possible directions from both technical and psychological sides. There are many motivations for that. One of them is that mobile devices ownership has been rapidly increasing [14], and malicious apps running on these devices pose risks to users’ privacy. Besides, many of the currently provided solutions for detection analysis heavily target the technical side, and we see several possibilities of improving user experience with mobile privacy protection.

1.2 Objectives and Hypotheses

The main aim of this work is to find better solutions for mobile privacy leakage problem from technical side, which will eventually help users make more privacy-aware decisions. The more detailed technical objectives are listed below.

- To design, develop and test a benchmarking system that analyses mobile privacy detection and protection tools, which will help developers evaluate privacy protection solutions and help inform users about the performance of different solutions. Previous efforts are about benchmarking privacy risks of normal apps rather than privacy protection apps, and only few focus on static analysis problems.

- To achieve a better understanding of human perception and behaviours in mobile privacy, focusing on the role of apps privacy ratings to see how they are perceived by human users to make decisions on their app choices.
1.2. OBJECTIVES AND HYPOTHESES

The expectation is users’ self-reported trust in subjective and objective ratings of apps privacy will depend on several factors. Some can be linked directly to human factors such as experience and characteristics, or to non-human factors such as the app category, and the type of feedback given (subjective/objective).

- To design, implement and test a new user interface that can potentially improve user engagement in privacy-related decisions on installation of apps on their mobile devices. We believe if mobile users are reminded of apps’ privacy level, it will help them to use more privacy-friendly apps.

However, that does not match the initial planning of this PhD work. Figure 1.1 roughly explains the stages of this PhD study. At early stages, the aim was to conduct the benchmarking experiment first (Step 1 at the Fig. 1.1) in order to identify research gaps on privacy protection tools. That step was accomplished. Yet, after testing many tools using PPAndroid-Benchmarker it turned out that the targeted improvements require some resources, technical expertise and time which are seen to be outside this work scope. Hence, the route was altered towards improving the user experience. At step 2, user perception of different app ratings was investigated to gain a better understanding of users’ behaviours about privacy ratings of mobile apps. Then, a design interface was proposed to enhance users’
privacy awareness and decision making on their mobile devices. The third work of this PhD was about that proposed design which was implemented in PAltRoid experiment (step 3 of the diagram).

1.3 Contributions

The main contributions of this thesis to the research community is summarised as follows:

- PPAndroid-Benchmarker has been developed for benchmarking the performance of mobile privacy protection apps for the Android platform (Chapter 3). The study involved testing on 165 privacy protection apps belonging to three different functional categories to demonstrate PPAndroid-Benchmarker’s effectiveness in evaluating these tested tools. The results showed that real-time dynamic monitoring tools perform better as they work on the underlying operating system level.

- To understand the perception of behaviours of users on mobile privacy better, an empirical study was designed and conducted on users’ perceived trust of subjective and objective ratings of mobile apps (Chapter 4). The study involved over 1,000 crowdsourcing participants and gives a conclusion that users’ perception of trust is influenced by several factors, such as context, personal characteristics and some others.

- A new user interface was designed and developed to enhance user awareness and assist decision making towards more privacy-friendly mobile apps. PAltRoid was designed for that and tested with some user studies reported in Chapter 5, which showed the positive impact of the presented system on participants’ privacy-related decisions on installation and uninstallation of mobile apps. PAltRoid has successfully motivated approximately 29% of participants to uninstall/substitute low privacy scoring apps when alternatives are offered, and 13% of the users when no alternatives are offered.

Moreover, A literature review has been presented to obtain a comprehensive understanding of the current issues on mobile privacy and research efforts towards solving these issues (Chapter 2).

1.4 Structure

The remainder of this thesis is organised as follows:

Chapter 2 reviews the research literature on mobile privacy detection and prevention. It starts with highlighting privacy in general. Then, it shows some scenarios
1.4. STRUCTURE

and solutions for user privacy on computers. After that, it goes deeper to mobile devices specific privacy problems and solutions. Next, it focuses on the Android platform (the most studied mobile platforms for mobile privacy research), and finishes with presenting related work from the research literature.

Chapter 3 introduces PPAndroid-Benchmarker, a mobile privacy benchmarking system. This chapter starts by reviewing related work. Then, it shows the detailed design and architecture of this system, followed by the implementation of analysing a large number of mobile privacy protection and prevention tools developed by both the research community and the industry.

Chapter 4 focuses on a crowdsourcing based user study regarding human users’ perceived trust on subjective and objective ratings on privacy-friendliness of mobile apps. This chapter starts with an introduction followed by the related work. Then, it describes how the user study was designed and conducted, followed by the detailed data analysis. The last section concludes the entire work and demonstrates directions of future work.

Chapter 5 presents PAItRoid, a new system to engage users in privacy-related decisions in the context of privacy ratings of mobile apps on the Android platform. The chapter starts by a small introduction followed by related work. Then, the detailed design, experimental studies, and a discussion on PAItRoid are presented. Finally, the chapter closes with the conclusion and future work.

Chapter 6 summarises the whole thesis and draws our future work.
Chapter 2

Literature Review

2.1 Privacy: Overview

The general understanding of privacy among people is the state where their own space is not invaded by others or not being disturbed or observed in any sense. Authors in the research have presented many definition of privacy. Clarke et al. [15] have precisely defined privacy as the interest that individual have in sustaining a ‘personal space’, free from interference by other people and organizations. Furthermore, they divided privacy into four dimensions, bodily privacy, personal behaviour, personal communication and personal data. While Westin [16] explained privacy as the claim of individuals to determine for themselves when, how, and to what extent information about them is communicated to others. According to Langheinrich et al. [17], privacy has different goals in different contexts, that means it is difficult to have a standard definition for privacy in a specific technology like computers or mobiles. However, some researchers came out with their own way of defining it which helps to understand. Montjoye et al. [18] described privacy as the guarantee that participants maintain control over the release of sensitive information that relates to them. That leads to know any sensor readings and user interaction with devices should be both protected. Another definition by Lucas et al. [19] is that privacy can be achieved when personal data are not accessed without informed consent, individuals control their personal data and have freedom from judgement by others. Another common definition offered by AICPA and CICA in GAPP 1 standard, which is “the rights and obligations of individuals and organisations with respect to the collection, use, retention, and disclosure of personal information” [20].

There are several reasons that justify why privacy is significant. From a psychological aspect, individuals need to have their own private space. This must be applied in private and public. Sociologically, everybody is free to behave, free

---

1 (AICPA) American Institute of Certified Public Accountants, (CICA) Canadian Institute of Chartered Accountants, (GAPP) Generally Accepted Privacy Principles
to socialise with others, subject to social customs, but without being threaten by surveillance, whereas economically, individuals must be free to innovate. Furthermore, from a political view, individuals need to be independent in their thoughts, arguments and actions. Being observed threatens democracy, and spoils speech and conduct [21].

Privacy protection can be described as the process of finding appropriate balances between privacy and multiple competing interests. Interests include any conflicting interest of another person, a group of people, organisation or even individual’s interests that might conflict with each other [21].

2.1.1 Privacy vs. Security

Privacy is very complicated as it overlaps with other practices and features such as information security. In this part, the overlap between security and privacy will be explained. Then, some terminologies that relate to privacy are going to be presented from ISO27000 [22]. Finally, the principle of security by design and privacy by design will be demonstrated.

Following the provided definitions of privacy in this thesis, it is possible to say that protection of privacy is only applicable to any personally identifiable information, whilst information security purpose is to protect information confidentiality, integrity, availability, authenticity and reliability, as ISO 27000 defines it. It leads to know that security applies to personal and non-personal data as well. However, some could argue that confidentiality, which is a component of security, is the same as privacy. That can be true in case of the information we want to protect is personal. Even so, confidentiality element covers both types of information.

Based on ISO definition of information security, the generally accepted elements of it are; confidentiality, integrity, availability, authentication, authorisation, reliability and accountability. Where the commonly acknowledged elements for information privacy can be taken from Fair Information Practices (FIPs), which set out by OECD [1]. These elements are:

1. Collection limitation
2. Data quality
3. Purpose specification
4. Use limitation
5. Security safeguards
6. Openness
7. Individual participation
8. Accountability
A joint report by the Information and Privacy Commissioner/Ontario and Deloitte and Touche (Canada) [1], stated that it is obvious when drawing privacy and security elements that some of privacy elements can be addressed by security controls, while others cannot. Sometimes security safeguards weaken privacy protection. Likewise, privacy measures could hinder security measures. Figure 2.1 shows where security and privacy can meet.

Anonymity

It ensures that a user may use a resource or service without disclosing user’s identity. The requirements for anonymity provide protection of user identity. Anonymity is not intended to protect the subject identity.

Unlinkability

It ensures that a user may make multiple uses of resources or services without others being able to link these uses together. That protects individuals from being profiled by knowing their habits and interests.

Traceability

The ability to verify the history, location, or application of an item by means of documented recorded identification. By being untraceable, people can have their right to be forgotten.
Confidentiality

A characteristic that applies to information. To protect and preserve the confidentiality of information means to ensure that it is not made available or disclosed to unauthorised entities. In this context, entities include both individuals and processes.

Security and Privacy by Design

The European Security Research and Innovation Forum defines the security by design (SbD) concept as “to embed security in the technology and system development from the early stages of conceptualisation and design” [23]. In other words, security should be taken into account when designing from the ground up. That applies for privacy as well. Thus, privacy should be adopted as a proactive compliance tactic rather than a reactive one. The Office of the Information and Privacy Commissioner/Ontario defines PbD as the approach to protecting privacy by embedding it into the design specifications of technologies, business practices, and physical infrastructures [24]. Cavoukian and Chanliau said: “Privacy must be pro-actively incorporated into networked data systems and technologies, by default. The same is true for security. Both concepts must become integral to organisational priorities, project objectives, design processes, and planning operations”. The objective of Privacy by Design is ensuring privacy and gaining personal control over one’s information and, for organisations, gaining a sustainable competitive advantage. That may be accomplished by practising the following 7 Foundational Principles [23]:

1. Proactive not Reactive; Preventative not Remedial
2. Privacy as the Default Setting
3. Privacy Embedded into Design
4. Full Functionality - Positive-Sum, not Zero-Sum
5. End-to-End Security - Full Life-cycle Protection
6. Visibility and Transparency - Keep it Open
7. Respect for User Privacy - Keep it User-Centric

2.1.2 Legal Aspect of Privacy

The right to privacy is grounded on the natural law resulting from the human ability to reason. Therefore, the law started to acknowledge and protect the rights of privacy regardless of the level of protection [25]. In this section, we focus on some law elements that are related to computer privacy. We take as an example the UK and EU jurisdictions and describe how they recognise those elements.
Protection of Personal Data

In the UK, the GDPR protects individuals when it comes to processing their personal data. According to the Data Protection Act 2018 [26], the GDPR protects personal data through the following:

- requiring personal data to be processed lawfully and fairly, on the basis of the data subjects consent or another specified basis,

- conferring rights on the data subject to obtain information about the processing of personal data and to require inaccurate personal data to be rectified, and

- conferring functions on the Commissioner, giving the holder of that office responsibility for monitoring and enforcing their provisions.

The GDPR even announced stricter rules for more sensitive information such as race, ethnicity, biometrics, etc. Moreover, it describes in details individuals' rights about what organisations and the government store about them and how they process their personal details.

As for the European Union, EU Data Protection Directive is the responsible body for protecting privacy and collected personal data about EU citizens. This Directive is based on seven principles which are [27]:

- Notice: subjects whose data is being collected should be given notice of such collection.

- Purpose: data collected should be used only for stated purpose(s) and for no other purposes.

- Consent: personal data should not be disclosed or shared with third parties without consent from its subject(s).

- Security: once collected, personal data should be kept safe and secure from potential abuse, theft, or loss.

- Disclosure: subjects whose personal data is being collected should be informed as to the party or parties collecting such data.

- Access: subjects should be granted access to their personal data and allowed to correct any inaccuracies.

- Accountability: subjects should be able to hold personal data collectors accountable for adhering to all seven of these principles.

In the context of the Directive, personal data means “any information relating to an identified or identifiable natural person (data subject); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological,
mental, economic, cultural or social identity” [28]. That means any data can be considered personal as long it is possible to link that data to a particular individual. It is also true even if who holds the data cannot make that link.

**Right to be Forgotten**

In 2010, the right to be forgotten was first introduced in French law. It allows deleting emails and text messages from on-line and mobile service providers at customer request after an agreed-upon length of time. In Europe, this right is part of the right of a personality who protects the moral and legal integrity of people. In that context, users can protect their identity on the Internet by preventing other parties from collecting personal information. For instance, authorities in Spain requested Google to delete news links, which are out of date and expose citizens’ privacy. In the USA, some organisations were noted to practice the right to be forgotten. They made an effort to spread the idea of protecting children private information using eraser button [29].

**Cookies Compliance**

Simply, cookies are sort of short-term memory for the web. By saving few data between visited pages, cookies make the browsing experience more personal, which is considered positive. However, it has been proved that cookies create behavioural profiles through collecting information across numerous web sites [27]. In May 2011, the EU Directive started first to adopt cookies compliance law. The user can invoke the right to refuse cookies of visited web pages to protect their on-line privacy. In the UK, Privacy and Electronic Communications Regulations have been updated following the Directive. Cookies and similar technologies are required to show clear and comprehensive information and consent of the website user [30].

**2.1.3 Privacy as a Human Right**

Although privacy is a difficult concept to be defined, it has a basic and intuitive feel to it. Throughout the history, human being expressed the feeling of the need to have a protected private space. Some argued that privacy has existed as an innate human need before any law or political society. That argument leads to the theory that privacy lays in people’s natural needs [25].

Developments in the technology world dramatically improve data sharing and communication. Still, it has come to be clear that modern technology is exposed to many malicious activities such as interception and surveillance. Such activities may threaten individual privacy rights. In the United Nations Human Rights, it is plainly stated that people should be protected on-line in the same manner as they are protected off-line. That was adopted in the ‘resolution 68/167’ that assesses individual privacy rights as it states: “Everyone has the right to the protection of the law against such interference or attacks” [31].
2.2 User Privacy on Computers

Computers, either PCs or portable devices, and any information recorded on them are parts of users’ private life that should be protected. It is universally understood that user privacy is a fundamental human right. Stamatellos [32] stated that to protect the privacy of personal data, data concerning an individual and data owned by an individual should be both protected. Research community has provided many techniques and solutions to contribute to privacy protection. However, many privacy issues and concerns are still rising due to activities of hackers, spammers, on-line merchants and organisation mistakes in handling private data [33]. In this section, we present selected computer privacy issues using a scenario-driven approach. Later, a data-driven approach in classifying privacy issues will be proposed.

Behavioural Advertising

Internet tracking collects a rich set of information about users’ usage of the Internet. Therefore, it can gather sensitive information like interests and browsing habits. Cookies are the most common tracking technique. They are proven to be useful in many ways, one of them is profiling users and serving them with related advertisement based upon their interests. Behavioural advertising on the other hand, is done by advertising networks. They are responsible for effectively distributing their ads to the users. That happens through controlling tracking technologies and compiling user profiles.

Nevertheless, on-line tracking technologies raise concerns about user privacy. It is somehow vague to the users who tend not to favour profiling and advertising [34]. One of the concerns is advertising parties are able to link individual profiles with identifiable information. On-line users’ activities can be matched with their names and addresses. Furthermore, on-line profiling can be used against users by rising prices based upon their needs [35].

Accidental Data Leakages

Shabtai et al. define data leakage as “accidental or unintentional distribution of private or sensitive data to an unauthorised entity”. There are numerous incidents of data breaches that were reported. In 2009, Verizon Business RISK team analysed 90 data breaches, which compromised around 285 million records. Another example, privacy Rights Clearinghouse reported 227,052,199 records with sensitive information in the US were exposed between 2005 and 2008 [36]. Moreover, the massive hack incident against Sony in 2013 when it suffered from millions of leaked records and huge losses that were estimated around 15 million [37]. Actually, there are a lot of similar reported incidents that are made available for the public on the Internet (e.g., datalossdb.org).
Geolocation

Web sites try to acquire visitors’ physical location by examining their IP addresses. Once obtained, websites determine what to display and allow users to do. For instance, a company may display the prices of its products in different currencies based on visitor’s location, or it might block users from some countries because of legal restrictions. People views about using geolocation against them differ. Salomon stated that many users think that not all geolocation are bad as they cannot identify individual users. Even though geolocation schemes assist in providing relevant services and complying with different countries’ laws, privacy advocates count it unethical. They believe it could lead to confusing information and less privacy. Salomon quoted Jason Catlett as “the technical possibilities do allow a company to be two-faced or even 20-faced based on who they think is visiting” [33].

Medical Privacy

The National Committee of Vital and Health Statistics in the United States adopted a definition of medical privacy as “Health information privacy is an individual’s right to control the acquisition, uses, or disclosures of his or her identifiable health data”. In order to protect patients’ privacy, patients must have the right to identify how their data is handled. In that sense, misusing patient data or collecting data without patient consent might end with risking their privacy and health. For instance, that could result in medical fraud, fraudulent insurance claims and even endangering patient life [38].

Anonymous Web Browsing

Anonymous web browsing refers to the usage of the World Wide Web while hiding users’ personal information from visited websites. There are many proposed and applied techniques to achieve anonymity in browsing, such as VPN, proxy servers and anonymity programs. However, these programs are susceptible to traffic analysis and users may be uniquely identified using cookies, browser plug-ins or others. One of the identified problems is the device fingerprinting that can be resistant to anonymisations. According to Eckersley, “It has long been known that many kinds of technological devices possess subtle but measurable variations, which allow them to be ‘fingerprinted’. Cameras, typewriters, and quartz crystal clocks are among the devices that can be entirely or substantially identified by a remote attacker possessing only outputs or communications from the device” [39].

2.2.1 Data-driven Approach in Classifying Privacy Issues

Due to the large variety of computer technologies, privacy issues and concerns are difficult to be covered. A similar work to be mentioned is by Chen and Zhao [40]. They provided a concise analysis on data security and privacy protection issues
2.3 Privacy Protection Techniques

Privacy protection has gained a considerable attention from both academia and industry introducing a wide selection of technical means for protecting user privacy. In this section, many of these techniques are reviewed. The solutions can be seen across data life cycle stages in cloud computing. These are generation, transfer, use, share, storage, archival and destruction of data.

As stated earlier, privacy has different goals in different contexts. Accordingly, privacy can be found in many aspects of computer technologies. While reviewing privacy related issues in this report, it has been identified that there is a need to classify them following a specific approach. Hence, the data-driven approach is proposed. It is believed that looking at the problem from a data flow angle will help in demonstrating it. This approach will be used later in 2.5 in mobile privacy which we believe that it will present the contents in a systematic manner. In the following, data-driven approach stages are explained:

1. **Prior to Data Creation:** The first stage to think of is before creating any data on the computer device. Computer applications normally need to obtain data from the user. At this level, data sources are investigated.

2. **During Data Creation:** All data with which a user is interacting are of concern in this phase. An example for that to think of is when a user making a call using a PC, how it handles his recorded information. Furthermore, while writing a text or an email how the system controls them and prevents any interference from other parts. In this level, the focus will be on access control and how the system distributes permissions and privileges.

3. **Dynamic Data Collection:** Programs collect data in run-time as they require to function. The privacy concern here is how these programs treat collected data. Moreover, another concern is the data that is going to be shared and for how long it is going to be kept.

4. **Data in Transmission:** A similar term is data-in-motion, which includes any data that is transferred inside or outside the computer. That includes transferring data between programs or system parts internally, uploading it to the web or cloud, or sharing it with other devices. Privacy concerns at this stage are communication channels, used protocols in transfers and whether transmitted data is encrypted or not.

5. **Residual Data:** At this level, the privacy concern is about residual data that is no longer required whether it has been completely destroyed or it can be restored. Additionally, how temporary and similar data are managed in the computer. These types of data may result in disclosure of sensitive information.
as a detective or protective approaches. In the detective side, binary-code and program analysis methods are some examples. In the protective side, researchers used to describe methods as privacy-enhancing technologies (PET) or privacy-preserving computing approaches. Blarkom et al. defined PET as “a system of ICT measures protecting informational privacy by eliminating or minimising personal data thereby preventing unnecessary or unwanted processing of personal data, without the loss of the functionality of the information system” [41]. In this section, many examples of privacy protection techniques which categorised as detective or protective are presented. Many proposed work derive their ways to achieve privacy concepts. As these theoretical concepts are important and relevant to this work, we will demonstrate some of the famous theoretical privacy concepts from the research in 2.4.

2.3.1 Detective Approaches

These types of approaches are designed to detect privacy incidents as they occur and then respond to them accordingly. That can be achieved through application analysis. In general, dynamic analysis is utilised to detect program flaws during the run-time, while static analysis inspects errors of applications’ source code. The light will be shed on different methods of analysis, which are static analysis, dynamic analysis and binary-code analysis.

Binary Code Analysis

One of the key aspects of cyber security is to detect program flaws and malware. The program static analysis cannot be applied, unless the binary-code of the software is available. Binary code analysis can be also static or dynamic. Static method involves checking the software without running it to derive its main features. That is done in two steps, disassemble the binary-code and then statically analyse resulted code. Researchers have proposed various methods to analyse the assembly code such as control flow analysis and reverse engineering. That means to understand how all parts of the code are built, what are their functionalities and how these parts interact and relate to each other [42]. On the other hand, dynamic method involves examining the software based on particular execution traces. It includes deriving the binary executable properties that hold for executions using run-time data.

Binary code analysis is not an easy task to accomplish. One of the reasons behind that is the code and data are identical and difficult to be distinguished. Nonetheless, there are many binary code analysis techniques that have been presented in the research community. Still, there is a big opportunity for future work to fill the gap between the two approaches. Having a hybrid static-dynamic method could make binary code analysis more accurate and complete [43].
2.3. PRIVACY PROTECTION TECHNIQUES

Static and Dynamic Analysis

Along with the increased importance of programs on computers, the negative effect of security vulnerabilities has rapidly increased. Security flaws which may lead to the exposure of private information are constantly reported. There are many reasons for that to happen, such as lack of programming skills, time and financial constraints and limited security awareness of developers. One of the methods to mitigate these threats, is statically or dynamically analysing applications. Static analysers scan the program source code for vulnerabilities, whereas dynamic tools search for attacks during the run-time of the audited application.

In static approaches, the goal is to identify if tainted data reaches sensitive sinks without being properly sanitised. To achieve that, data flow analysis (DFA) is applied by computing determined data for every program point. DFA operates on the application’s control flow graph that is produced for the tested program. Hence, static analysis is carried on a non-runtime environment. One of its advantages is being more comprehensive than dynamic method. Furthermore, it can be more cost-efficient by detecting bugs at early stages of software development life cycle with no run-time overhead. It also discovers future errors that dynamic analysis does not.

On the other hand, dynamic analysis adopts the opposite method and is implemented during application run-time. This approach is utilised through what is called taint checking. Dynamic analysis can expose any hidden flaws and complex vulnerabilities that static analysis cannot reveal. Nevertheless, it will only detect vulnerabilities in the executed part of the code and adds overhead to current processes [44], [45], [46].

2.3.2 Protective Approaches

Preventive mechanisms are meant to prevent potential privacy incidents before they occur. As privacy is considered a wide topic and can be seen from different angles, there is no technique to solve all privacy-related problems. However, researchers and privacy advocates have done an outstanding effort in presenting various solutions and techniques. In this part, some of the privacy protecting techniques are demonstrated.

Anonymity Techniques

Being anonymous means that the user cannot be tracked on-line. One of the approaches to achieve anonymity is to “strip identifying headers and resend” method. That has been used in email re-mailers and web browsing tools. An example of these tools is Anonymizer which is a web proxy that removes any identifying information from the web browser. Onion routing is another way that was derived from the idea of mix network. Simply, a mix network is a chain of proxy servers. In onion routing, messages are encrypted to each mix node using public key cryptography. The
encryption can be described as the outer layers and messages are the inner layers which look like an onion. Each mix node removes its own layer of encryption to determine what is the next receiving mix node. If all mix nodes are not compromised, the message cannot be traced. One of the concrete onion routing system is Tor. A third major method is based on k-anonymity concept, which will be described later in 2.4. That concept involves publishing individuals’ data without exposing their identifying sensitive information [47].

Authentication and Identity Management

The idea of authentication is ensuring that an individual is really the identity he/she claims to be. To fulfill that a combination of user name and password is applied. The user name represents individual identity, and the password represents his/her authentication. Currently, there are many sophisticated ways of authentication by adding more factors such as something the user has e.g., ATM card or something the user knows e.g., memorable word.

In contrast, identity management can enhance user privacy by allowing to have multiple digital identities. For instance, users can use different Google accounts in their different applications. Moreover, Microsoft’s CardSpace is another example where users are allowed to have many virtual ID cards where each card contains the minimum amount of information. CardSpace authenticates users through identity cards, such as a driving license and a credit card [47].

Homomorphic Encryption

Homomorphic encryption is the operation of ciphering a plain-text so it is possible to analyse it as if it is still in its original form. Homomorphic encryption involves performing complex mathematical operations on encrypted data without having to decrypt or compromise the encryption. This technique shows the transform of one data set into another while preserving relationships between elements in both sets. Hence, data in this type of encryption scheme keeps the same form which means mathematical operations will give the equivalent results, whether they are performed on plain or cipher-text [48].

Authorisation and Access Control

Authorisation includes controlling access rights by allowing or rejecting attempts. There are three types of access control: classic way, role-based and directory-based access control. In the classic method, an access matrix will be used to control the access to system resources. While role-based access control, permissions are assigned based on roles but not subjects. In a directory-based access control model, every directory has its subjects, and permissions are assigned to directories. Some privacy policy languages have an access control aspect, e.g., P3P and XACML [47].
2.4 Theoretical Privacy Concepts

Secure Multi-Party Computation

Secure multi-party computation (MPC) can be described as the problem of computing in a secured way an agreed function of certain identities’ input, guaranteeing the privacy of it and the correctness of the output. A famous example is the Yao’s millionaire’s problem where two millionaires want to know who is richer without revealing each one’s wealth [49].

Privacy Policy Languages

Privacy policy is a detailed description of an organisation’s information practices, that can be accessed from the organisation’s website. Its objective is to make users aware of any of the website’s privacy related practices. Hence, they are targeting human readers. On the other hand, privacy policy languages are directed to be machine-readable. They can be classified into two external and internal policy languages. The former is to depict websites’ public privacy policies or preferences. The latter assigns websites’ internal rules of privacy practices. Generally, “external privacy policy languages are declarative without enforcement mechanism, while internal privacy policy languages are normative with support for enforcement” [47].

2.4 Theoretical Privacy Concepts

In this section, some of the privacy models are highlighted due to their relevance to this work as mentioned in 2.3. The effort is not to cover all of them; yet important models will be focused on. The majority of these models utilises some transformation on the data to preserve the privacy. Usually, they minimise the granularity of representation, so the privacy can be reduced. That leads to losing some efficiency of data mining or management. Hence, there is a possibility of trading-off between privacy and information loss [50]. Here are some examples of privacy models:

- **K-Anonymity**: The reason of developing k-anonymity privacy model is the potential indirect identification of public databases’ records. That is due to using series of record attributes that possibly help in identifying people’s records. There are several techniques to minimise the granularity of data representation in k-anonymity method, e.g., suppression and generalisation. Using such techniques adequately makes any single record maps onto k other records in the data set [50].

- **L-Diversity**: l-diversity model was presented to deal with the weaknesses in k-anonymity model. For instance, if there is a similarity in personal values within a group, l-diversity involves protecting the corresponding personal values rather than only protecting records of k-identities. Thus, sensitive attributes in each data set must be diverse [50].
2. T.Closeness: Li et al. show that there is still few limitations on l-diversity and it is unnecessary and insufficient to restrain attribute revelation. Consequently, they propose the t-closeness model and suggested that “the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table” [51].

• Differential Privacy: In general, differential privacy guarantees that removing or adding a single record will not influence the result of any analysis of the database. That means no risk is caused by joining different databases [52]. Differential privacy “assures record owners that they may submit their personal information to the database securely in the knowledge that nothing, or almost nothing, can be discovered from the database with their information that could not have been discovered without their information” [53]. Thus, this model offers rigorous mathematical guarantees against what attackers can infer from using the results of some randomised algorithm.

2.5 User Privacy on Mobile Devices

Mobile devices are simply small computers. The main function of mobiles is making calls and receiving texts while moving around. Many features in the mobile environment make mobile privacy very important. ICO [54] reported some of them which are portability, frequently used, being consistently on, collecting many users’ information, and many others. Apps can misuse data in background and user desires of convenience that contradict with their privacy. As long mobiles are meant to receive calls all the time, they need to be always on when they are carried, unlike PCs. Moreover, advances in mobile technologies that look for user comfort and needs bring new sensors to the market, such as motion and position sensors. These developments allow mobiles to collect more sensitive data about users than in normal computers.

In this section, we present the classification of issues in mobile privacy research. Then, many mobile privacy specific issues are demonstrated in a data-driven order. After that, we end this section with mobile privacy protection techniques. Before describing these subsections, we first provide preliminary information about mobile devices’ data, which are significant to understand user privacy on mobile devices.

2.5.1 Data on Mobile Devices

Data here can be referred as any piece of information on the mobile device that relates to mobile users. It may contain sensitive details about the user which could be exploited to identify him/her. Otherwise, it may not contain sensitive details, yet it is possible to infer sensitive data from it through analysis [6]. Thus, data on mobile falls in one of the following types: raw data, data inferred from raw data and data inferred from aggregated data which possibly combined with non-mobile data.
2.5. USER PRIVACY ON MOBILE DEVICES

Theoharidou et al. [55], came out with their own way of classifying mobile data. Their taxonomies are based on data source and information type. Based on the data source, they classify mobile data into messaging data, device data, SIM Card data, application data, usage history data, sensor data and user input data. Based on the information type, they classified data as personal data, business data, government data, financial data, authentication data and connection/Service data. This research divides mobile raw data into three types:

1. Device identifiers such as IMEI, IMSI, Android ID, Android Advertising ID, UDID (aka universal device ID in iOS), IDFA (aka identifier for advertising in iOS) and Windows Advertising ID

2. Sensors data such as generated picture and video files by camera, audio files generated by microphone, acceleration and rotation forces along three axes by motion sensors, GPS location coordinates and environmental parameters like temperature, humidity and pressure

3. Personal communication data which are messages, contacts, email, browsing history and calendar data

For the second and third type of mobile data, data inferred from raw data and data inferred from aggregated data, there are a lot of work proved that they exist. Further examples and description will be provided later in 2.5.3 in data inference part.

2.5.2 Classification of Issues in Mobile Privacy Research

According to Haris et al. [6] mobile privacy is often investigated either on software level or hardware and communication level. The former is precisely the operating system and application level which concern with privacy models, data flow, sources and sinks of privacy, privacy solutions’ effectiveness and user attitudes toward privacy. The latter is sensors and communication level which are of interest for those looking at mobile sensors leakages, privacy against sensor data inference and leaks through mobile communication protocols. Briefly, Haris et al. categorised mobile privacy issues into four groups. They are issues related to mobile applications, mobile sensors, mobile users and mobile connectivity.

2.5.3 Selected Mobile Privacy Specific Issues

Many researchers made effort in investigating mobile privacy issues. In this section, mobile privacy concerns will be represented based on the proposed data-driven approach in 2.2.1. Related privacy issues are going to be organised based on the data state and when privacy concerns take place.
Application Markets

In 2008, Apple was the first to provide a centralised market that offers a variety of apps. This idea helps in removing the burden on both users and app developers. Users can easily search for an app and download it. On the other hand, developers do not worry about publishing and distributing their apps. Currently, all mobile platform companies provide apps through their private application stores. However, there are third party companies that entered this competition with their own markets such as Amazon, Firefox Marketplace, Opera and many others [56].

According to Enck [13], application stores and markets can provide three security and privacy utilities. In order to understand them, this research will compare them between App Store and Google Play as they are the most commonly used markets by developers [57]. Firstly, implementing the walled-garden model to have control on what apps the user can install. Apple applies this model on all its iOS devices. However, Android gives the freedom of installing apps from anywhere, which can be considered as an advantage and security drawback at the same time. Secondly, markets can check on application security certification. Apple applies a vetting process on any app wants to be placed in the App Store. However, the vetting procedure is unclear. On the other hand, Google Play uses automated analysis in scanning new candidate apps, namely Bouncer [56]. Lastly, application stores can offer remote software management. Accordingly, Google used that utility to remotely remove malware apps from handsets and patch damaged OS [58].

User Decisions before Installing Apps

There are many factors influence users’ decisions when choosing mobile apps. While selecting an app, users are often motivated by reasons such as whether the app fits their needs, preference of the interface, being trustworthy and privacy issues [59]. However, mobile users are not accurately aware of the threats associated with installing an app [60]. This lack of understanding leads to choosing apps that may compromise user privacy.

Inferring from Raw, Aggregated and Sensor Data

Many researchers proved that various personal information can be inferred from mobile raw data. There are many examples that fall in this category. For instance, Seneviratne et al. [61] managed to predict user’s gender using a supervised learning technique by knowing installed apps. Chittaranjan et al. [62] applied data mining and machine learning techniques on mobile data to infer user’s personality. Likamwa et al. [63] predicted user’s mood by using many logs that have been collected from visited websites, SMS, voice calls and emails.

Another way is inferring data from aggregated data that is possibly combined with non-mobile data. Seneviratne et al. [64] show how to predict user traits from a single snapshot of installed applications on the user device. They manage to
infer information such as religion, language, countries of interests and others using supervised learning methods and minimal external information. Another example, Pan et al. [65] showed that by combining collected information such as affiliation and friendship with many mobile usage logs, it is possible to infer future app installation behaviour of users.

Another way of inferring data has been revealed, which inferring from sensor data. Simon and Anderson [66] have explained in their paper how to retrieve PIN numbers by launching a side-channel attack. They used the microphone and video camera to detect touch events and estimate the mobile device orientation, respectively. Owusu et al. [67] used the accelerometer readings to spy on keystroke information when the user key in a password. Their result indicates that 6 of 99 six-character passwords can be broken in as few as 4.5 trials.

### Access Control and File Permissions

Following the comparison between iOS and Android operating system, both implement access control mean to maintain privacy and security [68]. In Android, access control is applied through Android’s Sandbox that keeps apps’ data in an isolated environment and inter-process communications are controlled by access permissions. However, it is not the ultimate solution. Once apps get permission, they might share data with third parties without user approval. On the other hand, iOS isolates each mobile app in sandbox policies [13]. Even so, Apple mobile devices use a different type of permission, which is time-of-use permission. Therefore, the user will be prompted when apps execute sensitive operations. Nevertheless, iOS devices suffer from privacy information leakages same as Android [69].

Another weakness can be exploited is the privilege escalation problem. Although sand-boxing and access control offers security for Android, the privilege escalation can be launched against it. Davi et al. [70] prove that either a malicious app or a genuine one exploited at run-time could escalate granted permission. In their experiment, they escalate some privileges granted to the app’s sandbox and send a number of short messages to a selected number without corresponding permissions.

### Data Storage in Mobiles

At this stage, data storage needs to be focused on. In Android, multiple data-storage facilities have been made available. Namely, they are shared preferences, SQLite databases and plain old files. Apps can access them in several ways, either using managed and native code or using content providers. In the book “Android Hacker’s Handbook” [2], authors explicitly stated three privacy and security issues related to those storage facilities. The most common mistakes are plain-text storage of sensitive data, unprotected content providers and insecure file permissions.

In April 2011, Justin Case [2] reported a similar problem in Skype client for Android. He found that Skype was creating a lot of SQL and XML files in plain-text, with read and write permissions. Fig. 2.2 displays what Case has found.
CHAPTER 2. LITERATURE REVIEW

Figure 2.2: Reported problem in Skype client [2]

Apps and Ad Libraries Data Collection

The research community has proven that many mobile apps leak variant types of data [5,69,71–73]. Some of these leaks can be justified where others stand doubtful. Apps often need to send data outside the device to provide better services for users. Nevertheless, suspicious apps leak data with no relevance to its nature. Enck [13] stated that there is a need to identify whether a leak is intended or not in order to decide if privacy is violated. However, there is a related work that tried to answer that question. Keng et al. [74] correlated leaks to user action. Nevertheless, the problem still exists. Table 2.1 shows some statistics of detected leaks by privacy leakage detection tools.

On the other hand, advertising and analytic servers are harvesting many data about mobile users. Seneviratne et al. came out with PrivMetrics to help users choosing trusted apps in terms of privacy protection. They mentioned in their website that there are more than 25 companies who could identify and know the location of the mobile user [3]. Figure 2.3 demonstrates how these ad and analytic servers collect users’ data. These trackers or servers have access to many mobile third party apps, which allows them to build accurate profiles about users. On top of that, trackers are able to infer very sensitive information from collected data such as user’s gender, health condition and political interests, as stated earlier.

There are some reasons that help to spread this problem of third party ad libraries. Some mobile app developers tend to distribute their apps for free following users’ preference. These developers include advertisement libraries within their apps to gain some profit [6] as mentioned earlier. Another reason is due to installing over-permission apps. Over-permission means apps ask for permissions more than what
Table 2.1: Some statistics of detected leakages by mobile privacy tools [6]

<table>
<thead>
<tr>
<th>Tool/Framework</th>
<th>Platform</th>
<th>Technique</th>
<th>No. of Tested Apps</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scandal</td>
<td>Android</td>
<td>Static Data Flow</td>
<td>90</td>
<td>6 apps leak loc to ad-servers, 5 apps leak loc to analytic server, 1 app leaks IMEI to its server.</td>
</tr>
<tr>
<td>PiOS</td>
<td>iOS</td>
<td>Static Data Flow</td>
<td>1407</td>
<td>656 apps leak device ID, 36 apps leak GPS location, 5 leak address book information, 1 leaks browser history and photos.</td>
</tr>
<tr>
<td>AndroidLeaks</td>
<td>Android</td>
<td>Static Data Flow</td>
<td>25976</td>
<td>Approximately 57,299 leaks are found in applications; 63.51% leaks are found in ad code. Moreover, 92% leaks are related to phone data. 5.94% leaks are of location data and 0.46 and 0.61% leaks of WiFi and Audio.</td>
</tr>
<tr>
<td>DroidTest</td>
<td>Android</td>
<td>Dynamic Data Flow</td>
<td>50</td>
<td>Most apps leak model no., subscriber ID and mobile no.</td>
</tr>
<tr>
<td>ProtectMyPrivacy</td>
<td>iOS</td>
<td>Crowdsourcing</td>
<td>685</td>
<td>48.43% apps access Identifiers, 13.27% access locations, 6.22% access contacts and 1.62% access music library.</td>
</tr>
<tr>
<td>AppIntent</td>
<td>Android</td>
<td>Static Data Flow</td>
<td>1000</td>
<td>It is found that 140 apps have potential data leaks, 26 apps leaks data unintentionally, 24 apps leaks Device ID, 1 app leaks contacts and 1 app leaks SMS.</td>
</tr>
<tr>
<td>IecTA</td>
<td>Android</td>
<td>Intra-Component Analysis</td>
<td>3000</td>
<td>425 apps leak information directly about device and location data.</td>
</tr>
<tr>
<td>IntentFuzzer</td>
<td>Android</td>
<td>Dynamic capability leak</td>
<td>2183</td>
<td>50% of applications leak capabilities or permissions related to network state, phone state, location and internet connection.</td>
</tr>
<tr>
<td>AGRIGENTO [75]</td>
<td>Android</td>
<td>Differential black-box analysis</td>
<td>750</td>
<td>AGRIGENTO detected 278 leaking apps from the ReCon [76] dataset more than any other compared tool.</td>
</tr>
</tbody>
</table>
they really require to function. Leontiadis et al. [77] reported that 73% of the free apps they studied ask for at least one over-permission, which may exploit user private data.

**Data Transfer in Mobiles**

Even though data transmission security, in general, receives large attention, there is a lack of it in the mobile application world. That is possibly due to improperly applying SSL and TLS. It has been stated that app developers make mistakes in securing sensitive data while in transit. Some of the issues in this regard are: unencrypted data or not strong enough encryption, having certificate validation errors, using plain-text after failures and inconsistent use of transfer security per network type, e.g., cell versus Wi-Fi [2].

Researchers from the University of Ulm reported that few apps on Android sent Google authentication credentials over HTTP in plain-text. Calendar and Contacts are two of the apps they reported. If this sensitive information is obtained by malicious identity, the information owners will be impersonated. Moreover, many
tools are available to conduct a man-in-the-middle attack, so it will ease intercepting those tokens [2].

Residual Data

Mobile users have an incorrect assumption about residual data. They think resetting mobile devices to factory default setting would delete all data from memory. An experiment has been reported from PR Newswire that proves the misleading assumption. In 2011, a live experiment was performed on that matter. CPPGroup assigned an ethical hacker to review 35 second-hand mobile phones and 50 SIM cards within the UK. 247 pieces of sensitive data were left unhandled on 19 mobile devices and 27 SIM cards. The sensitive data contained log in details, passwords, bank account and credit details and company information. Moreover, ICM has randomly interviewed a sample of people to see their feedback regarding residual data. It has been reported that 81% of them claimed they wiped all data before selling their devices, with 6 in 10 are confident that no private data left. Conversely, ICM reported that 54% of their mobile devices and SIM cards included personal information [78].

On the other hand, uninstalled apps may leave vulnerable data behind in the device. In a recent work published by Zhang et al., authors explained what they called data residue attacks. They have shown through their analysis and experiments how deleted apps can remove sensitive data unattended. Such details could be exploited to steal users’ account credentials, access their private data, escalate privileges and few other vulnerabilities [79]. Same authors show in a different work how data can reside in the phone and leads to data leakage and privilege escalation attacks [80].

2.5.4 Mobile Privacy Protection Techniques

In this section, some mobile privacy protection techniques are presented following the same categorisation from computer privacy protection techniques in 2.2.1.

Application Analysis

The concept of application analysis for mobile devices is similar to the one for computers, which is described earlier in 2.3.1. The research community has provided various techniques of application analysis. In this part, application analysis techniques for mobile devices are presented.

Permission Analysis:
Some mobile platforms apply a permission framework which will be described later in 2.6.1. Researchers have illustrated their method of identifying suspicious apps by analysing app requested permissions. Enck et al. were the first in that by proposing Kirin. If an app uses permissions or action strings that are defined as a dangerous
functionality, Kirin detects it. Out of 311 tested apps from Google Play top free apps, 10 apps were flagged based on Kirin rules. Another work by Felt et al. is Stowaway. It identifies apps that ask for over-privileges. They analysed 956 apps from Google Play where they found that most common unnecessarily requested permissions are accessing the Internet and reading phone state. Even though risky apps can be identified by analysing their permissions, this approach is limited and still requires static or dynamic ones [56].

Static Analysis:
In this type, the analysis is run with (static source code analysis) or without source code (binary-code analysis). Currently, almost all mobile apps that apply this approach retrieve apps’ source codes using decompilers. Although the static method offers an automated comprehensive apps analysis, its precision may greatly rely on the decompiler performance or the quality of used coding. Additionally, this approach is known to generate false positive or false negative whenever there is a dead code in the retrieved source code. Moreover, static analysis is not able to decide if privacy behaviour of an app is justified or not from the user point of view [68].

Dynamic Analysis:
Unlike static analysis, dynamic ones can run without source codes. It detects data leaks once they are executed. By performing Data Flow Analysis in mobile platform, dynamic tools monitor sensitive sources. Users are notified in case there are flows between sources and sinks. One of the remarkable dynamic analysis tools is TaintDroid [5] (TaintDroid is described in details in 2.6.). The developers examined 30 Android common apps. They reported that half of them leak device location with ad servers and 7 of them frequently leak device IDs. Many tools are built on top of TaintDroid as extension for effectiveness evaluation, such as MockDroid [81], TISSA [82] and AppFence [83]. Nonetheless, TaintDroid authors discussed some of its limitations. First, it only detects explicit flows leaving opportunity for attackers to use implicit flows. Kang et al. defined implicit flows as: “parts of a program where tainted data values affect control flow, and then the control flow variation affects other data. This can lead to under-tainting, a type of error in which values that should be marked as tainted are not, and so for instance could cause an analysis to fail to detect a leak of sensitive information” [84]. Also, TaintDroid reports the data leakages without providing further information whether it is a privacy violation or not. Above than that, a group of researchers proved the possibility of bypassing TaintDroid which will be described later in 2.6.2 [85].

Hybrid Analysis:
Some other authors introduced tools that offer a combination of dynamic and static analysis together. The idea is to have the advantage of both approaches in order to
2.5. USER PRIVACY ON MOBILE DEVICES

improve privacy leakage detection. There are a number of works propose this kind of analysis such as, AppScanner, Gort, SmartDroid, and AppsPlayground [56]

Cloud-based Analysis:
Owing to mobile platforms have restrictions in resources, running the detection analysis on them can be difficult. The community proposed cloud-based analysis method to overcome that problem. Paranoid Android implements this approach and will be described later in 2.6.2. Shabati et al. [86] used this method by uploading user activity and resource usage to detect intrusions. Still, Enck [13] has shown how Miettinen et al. have shown the limitation of applying cloud-based analysis. They argued that there are administrative and technical boundaries.

Others:
Research community provided other proposed techniques to enhance current detection analysis approaches. One work proposed using user comments from official app markets. Another work used machine learning technique to generate automatically a list of sinks and sources. Other developers inserted a forensics element to provide causes of data leaks. Some tools utilise crowdsourcing to identify app vulnerabilities. Lastly, researchers have gone beyond detection and tried to help users defending their private information. AppFence, which is built on TaintDroid, gives the user the choice of blocking data leaks by using two methods, ex-filtration blocking and data shadowing. Another example is MockDroid and TISSA, where they let user substitute data to be leaked with a false data [6].

Protection Mechanisms

In this part, some of the proposed protection mechanisms by the research community are presented.

Rule-driven Policy Approaches:
According to Enck “The often-cited limitation of smartphone protection systems is insufficient policy expressibility”. Therefore, some researchers suggested policy languages to analyse mobile apps such as xJ2ME, SxC, Kirin, Saint, CRePE, XMan-Droid. These languages support their requirements to integrate new proposed policy into targeted OS. However, there is a limitation in applying the rule-driven approach which is the difficulty of accurately defining and maintaining created rules [13].

High-level Policy Approaches:
These approaches relate to the general high-level security aims in mobile devices. As an example, TrustDroid focuses on the isolation between applications. That is applied by dividing apps into system trusted and untrusted domains. Then, it controls the interaction between domains by allowing or isolating domains following its policy. Other works followed high-level methods in preventing some attacks
such as cross-service attack in Windows phones and confused deputy attack in Android [13].

Faking Sensitive Information:
A number of leakage detection tools tried to provide more protection to the user by giving control over sensitive information. Several works provide techniques to block leakage attempts or fake the capabilities of the device or the data provided to receivers. MockDroid is one example that provides mock data to requesting third parties. TISSA, which is another tool, gives the user more options in providing empty, anonymised or fake data [56].

2.6 Android as an Example

This PhD work focuses on Android platform for several reasons. First of all, Android is an open-source OS that allows researchers and developers to access it more than any other platform. This lets researchers build their own prototypes to approve new ideas. Another reason is being the most widely used mobile OS. According to IDC (International Data Corporation), Android is the dominant platform among mobile devices’ market share. Table 2.2 shows a recent usage statistics of mobile platforms.

<table>
<thead>
<tr>
<th>Year</th>
<th>2016</th>
<th>2017</th>
<th>2018</th>
<th>2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>84.6%</td>
<td>85.1%</td>
<td>84.8%</td>
<td>85.2%</td>
</tr>
<tr>
<td>iOS</td>
<td>14.7%</td>
<td>14.7%</td>
<td>15.1%</td>
<td>14.8%</td>
</tr>
<tr>
<td>Others</td>
<td>0.7%</td>
<td>0.2%</td>
<td>0.1%</td>
<td>0.1%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Additionally, Android is the only platform that has a flexible interaction between apps that brings many security issues for study, said by Enck [13]. To express more, it has a rich inter-application message passing system which poses many security problems. For instance, messages can be sniffed, altered or removed ending with compromising mobile user privacy.

2.6.1 Android Security Design and Architecture

Before going deeply into Android security, abridged explanation about Android is presented. The best short description for Android is Java apps built on a Linux-based OS. That explanation is a very concise, and it does not show the complex design of Android. In fact, it is divided into five layers, including Android apps, the Android Framework, the Dalvik virtual machine, user-space native code and the Linux kernel. Figure 2.4 simplifies these layers. Its applications are all written in Java and executed within the Dalvik virtual machine interpreter. Each of the
2.6. ANDROID AS AN EXAMPLE

Figure 2.4: Android overall architecture [4]

Interpreter instances is executed with a unique user identity to ensure that each app runs in isolated Linux subsystem. Whenever an app needs to communicate with another app, it can use IPC mechanism (Inter Process Communication) [6].

In the following paragraphs, some of the applied security components and measures in Android OS are briefly demonstrated, which serves privacy protection. They are Android’s sandbox, permission model, apps signing and encryption.

Android’s Sandbox

Android is based on Linux core as mentioned earlier which brings design principles from Unix such as least privilege and process isolation. Apps are run in an isolated environment where processes cannot interfere with each other. Sandboxing in An-
droid is predicated on three key concepts: standard Linux process isolation, unique user IDs and restricted file system permissions [2].

Android Permission Framework

Android platform is built with a permission framework. Apps are only able to access a limited range of system resources. For resources that can adversely impact the user data, experience or network, Android controls access to these resources via permissions. There are three types of permissions, API permission, file system permission and IPC permission. For the first type, protected APIs are camera functions, location data, Bluetooth functions, Telephony functions, SMS/MMS and Network/data connections.

This Android Permission Model protects users from two sides: control mobile apps access to sensitive locations and help users to make a proper decision prior to app installation [2]. Accordingly, app developers must declare needed permission in the manifest file which will help to explain to the user what resources will be accessed. That information will help the user to make decision whether to trust an app or not.

Once a permission is granted, it will be applied to the granted app as long as it is on the device. To avoid confusing the user regularly, he/she will not be notified whenever an app tries to access granted permission/s. If the user uninstall an app, the system will remove the associated permissions. Using the system settings, users can view, grant or block permissions for their installed apps. Alternatively, the user is able to globally turn off functionalities when they want (e.g., disable GPS, radio and WiFi [87].

Application Signing and Encryption

Android apps must be digitally signed with a certificate before they go for installation. Such signatures do not need a certificate authority. Android will use them to identify the author of the app. If an app requires any update, the certificate’s private key of the author will be matched to the existing version. Another use is to allow apps signed with similar certificates to run in the same process. Thus, apps can expose functionality between each other if they are signed with the same certificate. That secures app sharing of code and data.

In Android, all created data by users is encrypted before delivering it to disk, and all reads decrypt data before sending them back to the calling process. All that automatically occur by the kernel. It uses 128-bit AES algorithm operating under the CBC (cipher-block chaining) mode and Encrypted salt-sector initialisation vector (ESSIV): SHA256 [58].
2.6.2 Selected Android Privacy Protection Systems

While presenting privacy protection techniques for both computers and mobile devices, we categorised them into detective and protective approaches. These detection methods’ purpose, eventually, is to find a way of protecting users’ privacy. Therefore, we use for both categories the term ‘privacy protection’ systems. In the research and industry, there are quite a lot of tools and systems that have been designed to protect user privacy. In this subsection, selected privacy protection systems are illustrated. All presented works are related to Android platform, with purposely chosen different analysis methods.

**ScanDAL (Static Analysis) [71]**: It implements a static analysis to detect data leakages by converting app package from Dalvik byte-code to a defined intermediate language. Using abstract interpretations, suspicious flows will be detected. The authors of ScanDAL analysed 90 free apps and they found 11 of them leak private data. They reported many location leakages to remote ad servers, namely AdMob and AdSenseSpec. Besides location data, apps are found to leak IMEI to their app servers.

![TaintDroid architecture](image)

**Figure 2.5: TaintDroid architecture [5]**

**TaintDroid (Dynamic Analysis) [5]**: This tool uses dynamic analysis to monitor apps’ behaviour. Sensitive information sources are identified and tainted. Using DFA, TaintDroid monitors how apps handle tainted information and notifies the user if private data is leaked outside the device. Figure 2.5 explains how TaintDroid works. First, data is tainted in trusted apps and then tag markings are stored in a virtual taint map. When apps use tainted information, the Dalvik interpreter propagates taint tags. After that, taint tags with information are encompassed in
the parcel as trusted apps use modified IPC library. Next in step 5 of Fig 2.5, third party apps will receive tags transparently through the kernel. The altered IPC library removes the tags at the receiving end and assigns it to their values using the virtual taint map and tags are propagated to the app. If the third party app invokes taint sink library, the trusted library removes tags and sends a notification to the user.

**ScrubDroid (Anti-TaintDroid) [85]**: The authors of this paper investigated the limitation of TaintDroid, in tracking privacy-sensitive information on Android mobile devices. They presented multiple attacks on TaintDroid using generic classes of anti-taint methods. Therefore, they proved that taint tracking is not effective enough in case of attackers applying one of the methods that they showed.

**SmartDroid (Hybrid Analysis) [88]**: It combines static and dynamic analysis. This tool works at two levels. At the first level, it utilises a static path selector to mine for activity switch paths through analysing activity and function call graphs. At the second level, SmartDroid implements a dynamic analysis to traverse each UI element and explore the UI interaction paths towards the sensitive APIs.

**SuSi (Machine Learning Approach) [12]**: All dynamic, static and hybrid analysis methods share one idea, which is detecting flows between sources and sinks. Nevertheless, many of them use a static list of information sources and sinks. Lately, a group of researchers has proposed ‘SuSi’. It analyses apps’ source code and automatically generates a list of sources and sinks using supervised machine learning technique.

**Paranoid Android (Cloud-based Analysis) [86]**: This tool runs a synchronised replica of the tested phone on a cloud-based server. The main reason to use cloud is to avoid the constraints of detection analysis on normal devices. That allows to perform complex analysis. In the phone, a tracer will be installed to record required data which will be used to re-run app executions. Then, collected information will be transmitted via secured channel to the cloud-based server. Afterwards, a re-player re-runs apps in a smart emulator to detect privacy leaks.

**CrowdDroid (Crowd Sourcing) [89]**: It applies crowdsourcing in order to differentiate between reliable and unreliable apps that share similar name and versions. Crowd-sourcing tells behavioural traces from any unusual execution of the same app. By comparing both traces, CrowdDroid can detect fake copies of applications.

**Mobile Forensics of Privacy Leaks (Correlate User Actions to Leaks) [74]**: Many authors proposed privacy detection methods to identify data privacy leakages. Nevertheless, there is a lack of information about leak causes. By correlating user actions to leaks, authors demonstrates the causes from a user point of view. This
information may help end users to take preventive measures and understand the authenticity of the leak.

**PCLeaks (Static Inter-Component Analysis)** [90]: PCLeaks was designed to identify possible capability leakages. It is based on ICC vulnerabilities to perform DFA on Android apps. This tool has been tested against randomly selected 2000 apps. Authors have reported that 986 potential components leaks in 185 apps. They presented PCLeak Validator which finds an app that tries to exploit the detected leak. Out of the reported leaks, 75% of them are exploitable.

**IntentFuzzer (Dynamic Capability Leak)** [91]: Is a fuzzing tool dynamically that analyses apps for capability leaks. Capability leaks happen when an app lacks a permission to sensitive sources and tries to manipulate another app that has the permission by performing access on its behalf. The authors stated that out of 2000 tested apps, 161 have this vulnerability.

**VetDroid (Permission Analysis)** [92]: Is a tool that reconstructs sensitive apps behaviours from a permission use’s perspective. It applies a dynamic analysis approach and presents a systematic framework in constructing permission use behaviours. VetDroid helps researchers and developers to study apps’ internal behaviour.

As part of this PhD work, a benchmarking environment is designed and implemented. The reason behind that is the lack of a benchmarking tool for mobile privacy leakage protection tools. Hence, we aim to provide a real-time dynamic benchmarking system which helps in identifying possible improvements in related areas. In the following chapter, our system will be demonstrated in details.
Chapter 3

PPAndroid-Benchmarker: 
Benchmarking Privacy Protection Systems on Android Devices

3.1 Introduction

This chapter presents the first work of this PhD which is about designing a benchmarking environment for mobile privacy leakage detection and prevention apps. The idea came after reproducing some of the currently available mobile privacy detection tools, namely TaintDroid and Anti-TaintDroid ‘ScrubDroid’. TaintDroid is used as a base to build many other tools with different flavours of security and privacy features. It has been noticed that there is no real-time dynamic benchmarking system for mobile privacy leakage detection tools.

For each new privacy protection solution, there is always a problem of how to evaluate its performance against existing solutions. Similarly, given a number of candidate solutions, a user has the need to know which solution is the best for her specific needs. For researchers, a proper benchmarking system is also desired so that insights about how to improve exist solutions can be gained and the performance of any new solution can be properly evaluated. While such benchmarking systems are very important, surprisingly, there is no such system in the research or commercial worlds, not mentioning open-source tools. Instead, currently researchers and developers either depend on bespoke performance evaluation apps or collections of test apps to conduct such benchmarking tasks.
In this chapter, PPAndroid-Benchmarker is presented, a system for benchmarking mobile privacy protection systems on Android devices, which to the author best knowledge is the first of this kind. PPAndroid-Benchmarker is designed to be oblivious to details of mobile privacy protection systems, and can detect performance of combined mobile privacy protection apps in run time – this is why the term “systems” is being used rather than “apps”. It is effectively a system 1) simulating different kinds of tester-configurable privacy leakage activities; 2) capturing what leakage attempts were successful; 3) supporting a high level of automation for the whole benchmarking life-cycle; and 4) providing a good user interface for testers to configure benchmarking tasks. Moreover, this chapter also highlights two additional components of PPAndroid-Benchmarker at the design level (which have not been implemented in this prototype): 1) an Automatic Test Apps Generator for benchmarking static analysis based privacy protection systems; and 2) a Reconfigurability Engine allowing PPAndroid-Benchmarker to be reconfigured such as adding and removing information sources and sinks. Although the prototype system that has been implemented for Android OS only, the framework is generic enough to be applied to other mobile operating systems such as iOS. The work in this chapter targets Android platform for the reasons explained at 2.6. PPAndroid-Benchmarker has been tested with 165 selected Android privacy protection apps, and report the findings and some insights about current status and future directions of mobile privacy protection and prevention tools.

The rest of the chapter is organised as follows. Section 2 reviews the related work. Then, Section 3 discusses the design and implementation of the presented system. Section 4 presents the followed experimental set-up and Section 5 illustrates the results and analysis. Section 6 discusses the implications of the findings and some limitations. Lastly, Section 7 concludes the chapter and illustrates some future work.

3.2 Related Work

Performance evaluation has been an area of research in many research fields including computer science in general. Here we focus on performance evaluation issues around mobile privacy. Many researchers have presented their benchmarking systems and tools for mobile devices. Some are designed to test the validity of security and privacy protection tools, while others are built to benchmark normal apps. In this work, the focus is on benchmarking mobile privacy protection tools. In this section, we present related work from both types; evaluating privacy in normal apps and mobile privacy protection tools.

3.2.1 Privacy Metering and Scoring of Android Apps

Many researchers tried to help users to understand the potential impact of privacy caused by mobile apps. The reason is that users do not fully comprehend these
effects and thus behave in an insecure manner, such as ignoring warnings about requested permissions in installation [93]. Kang et al. [94] proposed a privacy metre, which clarifies the potential risk of requested permissions. Another framework proposed by Seneviratne et al. which is called ‘PrivMetrics’ [3]. It aids mobile users to make informed decisions before installing apps by showing users an analyses on apps privacy leakages. Moreover, PrivMetrics offers a recommendation of other apps with a same functionality as an alternative, yet better in terms of privacy level. Another work in this category is PrivacyGrade. The aim of its authors is to raise awareness of the behaviours that many mobile apps have which may affect users’ privacy. It uses a privacy model that measures the gap between an app’s actual behaviour and users’ expectations of that app’s behaviour. Then, PrivacyGrade shows the result of applying its model in a form of grade to make it easier to be understood by users [95].

3.2.2 Evaluating Mobile Privacy Protection Tools

As stated earlier, it is required to validate privacy protection tools in order to maintain user privacy. Several works presented systems to benchmark mobile privacy protection tools. In this subsection, we present DroidBench as an example of such works as a background for the presented benchmarking environment in this Chapter.

DroidBench

DroidBench is one of the provided benchmark testing tools which contains test cases for some static-analysis problems (e.g., field sensitivity, object sensitivity, tradeoffs in access-path lengths) to test both static and dynamic taint analysis tools. It only focuses on Android. DroidBench is a collection of different apps from various categories that pose data leaks. Also, other different apps without data leakages are included. DroidBench test security tools if they could identify the leaks as they occurred [96].

Stanford SecuriBench [97] is another example from the community. Although it focuses on Web-based applications written in Java, we mention it for its relevance. It comprises of eight actual open-source Web-based applications that have intentional security flaws. These apps are written in Java in a medium size. They deliberately suffer from several vulnerabilities, including SQL-injection, cross-site scripting, HTTP splitting and path traversal attacks. This tool is meant to serve as test cases for practitioners from both research and industry.

3.3 Design and Implementation

This section illustrates the design and some implementation details of the proposed benchmarking system.
3.3.1 PPAndroid-Benchmarker Overall Design

The main purpose of PPAndroid-Benchmarker is to evaluate privacy leakage detection applications in an automated manner. As stated earlier, the Android platform is targeted in this work. PPAndroid-Benchmarker is composed of three basic components, the Benchmarker App, the Drop-in Server and the PC-based mobile device manager (MDM). Figure 3.1 shows the architecture of PPAndroid-Benchmarker. Firstly, the benchmarker is programmed to simulate leakages of different types of private information. It has a profile creator that allows the user to define different benchmarking tasks. Secondly, the Drop-in Server is used to receive leaked information from the benchmarker. Lastly, the MDM handles automatic installation and uninstallation of tested apps during the benchmarking process.

Figure 3.1: PPAndroid-Benchmarker’s overall design

3.3.2 PPAndroid-Benchmarker Components

Figure 3.2 shows how PPAndroid-Benchmarker’s different components interact with each other as data flows between them, where the user is also shown as a “component” as his/her interactions with several components are needed. In the following, the three key components with greater details are explained.

The **Profile Creator** allows a user (who wants to tests some privacy protection apps) to create the actual benchmarking task (i.e., a benchmarking profile). Any profiles created can be stored in a profile database so that they can be reused in future. The **Profile Database** can also retrieve app-related information from an **Apps Repository** which will also work with the **MDM** (to be explained be-
3.3. DESIGN AND IMPLEMENTATION

Figure 3.2: PPAndroid-Benchmarker’s components and data flow map

The Profile Creator interacts with the user to collect information about a benchmarking profile and feed the created profile to the **Benchmarker App** for execution. The profiles are stored as XML files to make them accessible from other components and external applications more easily. The Profile Creator can be made part of the Benchmarker App or be implemented as a PC-based application which communicates with the Benchmarker App via USB or a wireless channel.

The **Benchmarker App** is the core of PPAndroid-Benchmarker and its main purpose is to simulate leakages of different types of private information. It is programmed to collect a variety of private information from the hosting mobile device. Thus, this app needs to be granted with all required permissions in order to access all data sources. This is not an issue as the benchmarker is used for testing purposes only, and can run on a dedicated testing device or within an emulator.
The Benchmarker App is programmed to leak information to a Drop-in Server, which is a web server set up to simulate an attacker’s information collection server. The Drop-in Server is designed to collect all needed information to create results of each benchmarking profile, which are stored in a Results Database for further (off-line) analysis. The Benchmarker App is connected to the MDM in order to facilitate the automation of the benchmarking process.

Furthermore, the Benchmarker App is facilitated with many anti-tainting tricks. Many designed dynamic analysis tools apply taint tracking technique, e.g., Taint-Droid, MockDroid, AppFence and many others. Therefore, several tricks were added, which are constructed to bypass dynamic taint tracking. In the current implementation, the following obfuscation techniques reported in Anti-TaintDroid [85] are included, which were verified to be still valid for the Android and TaintDroid versions that have been tested.

- **Simple encoding attack** is an array indexing attack where the tainting operation can be tricked by propagating taints of the array (X-Tainted is used to index an array of untainted variables to assign to Y-Untainted) and the index to the assigned variable.

- **Shell command attack**: The idea is to simply limit system commands to remove the mark off the variables. The aim here is to alter a system utility so the value of X-Tainted can be printed in some storage area in its output stream and then Y-Untainted becomes taint-free.

- **File+shell hybrid attack** takes place by separating read/write operations required to acquire a taint-free variable. A file can be created somewhere in the system, along with the tainted information as its content, and then can be read. If one of the operations wrongly propagate taint markings, the output variable is taint-free.

- **Time keeper attack** relies on the side channel created by the time it lasts to accomplish a task. This can work if a system clock readable without being tainted is available. The difference of time reads prior and post to a waiting period, which duration is based on a tainted variable value, is untainted, and can be used with a taint-free output variable.

- **Count-to-X attack**: Rather than traversing arrays to find out the value related to X-Tainted, this attack remakes the value one incremented at a time, until Y-Untainted finds the matching X-Tainted.

- **File length attack**: File meta-data can be exploited as a carrier to evade tainting process. Some data can be written to a file until the size of this file equals the value of X-Tainted. The resulted size can then be read without being marked.

- **Clipboard attack**: The same technique of the previous attack can be followed using a clipboard if it is available for applications use.
• *Exception/error attack* occurs by inserting executing paths that contain deliberate exceptions. Exception handlers will help to make taint-free variables to the values belong to the recognised value of X-Tainted leading to the exception.

• *Remote control attack*: The idea is to remotely taint-free the tainted variable X-Tainted and assign a known untainted value to Y-Untainted. This attack is carried out by sending commands using http communication with the designated server.

• *File last modified attack*: If a tainted variable is written to some file, the whole file will be marked as tainted too. By using any system command attack such as (cat /path/X-Tainted), the file can be modified with a malicious application to produce a Y-Untainted. Eventually that will break the taint propagation chain.

The **Mobile Device Management (MDM)** is a PC-based Android device manager handling automatic installation and uninstallation of each tested privacy protection app during the testing process. The MDM has been implemented to communicate with the Benchmarker App via a TCP port, although other communication channels can also be used. When the MDM receives an app download link, it downloads it and installs it to the mobile device using ADB (Android Device Bridge). After the benchmarking process ends at the Benchmarker App’s side, the MDM will receive a request of uninstalling the tested app.

**Drop-in Server**: In order to simulate the complete information leakage process, a sink is required to allow information to go out of the mobile device. In PPAndroid-Benchmarker, the Internet connection is used as the sink. To simulate the case of information leaked through the Internet, it is needed to set up a server that receives leaked information. A number of server-side scripts (written in PHP in this implementation) are used to handle received leakage information, some are used to receive leaked files, and some others to create the results as XML files.

### 3.3.3 Data Management

A prototype of PPAndroid-Benchmarker has been implemented including all the above components. At the time of this writing, this prototype supports the following information sources:

• **Device IDs**: IMEI, IMSI, Android_ID

• **Personal data**: SMS messages, contacts, call logs and browsing history

• **Sensor data**: camera, microphone, accelerometer axes data, last known geolocation by GPS device or the ISP (Internet service provider)

• **Files** on the mobile device’s external memory storage
The above list is not supposed to be complete, but used as a representative start of the presented prototype system. Adding more sources is a matter of improving the tool itself. To simplify this prototype, any advanced processing of information leaked were not applied (e.g. encryption and steganography) other than some tricks specially added to circumvent taint tracking techniques. Adding more advanced information processing operations to information leaked will not be difficult, but require a proper interface to allow easy reconfiguration (see below).

The presented design also considers two other major conceptual components, the **Automatic Test Apps Generator** and the **Reconfigurability Engine**, which have not been implemented in the current prototype system yet but will be added in future versions of PPAndroid-Benchmarker prototype.

### 3.3.4 Automatic Test Apps Generator

PPAndroid-Benchmarker is designed to benchmark privacy protection apps more in a dynamic way. To support benchmarking of static analysis tools, an automatic Test Apps Generator can be developed to allow generation of apps with different privacy leakage capabilities. The Test Apps Generator will take the source code of the Benchmarker App as the source and the user’s descriptions of the test apps wanted, and generate a number of apps with requested privacy leakage capabilities. The process of generating test apps can contain random factors so that a large number of test apps can be generated, which will produce much more test cases for static analysis based tools than other solutions can provide. The generated apps (in the form of apk files) can be used to benchmark any static analysis based privacy protection systems. This component can be achieved in several ways such as embedding a compiler that can automatically convert the source code of the Benchmarker App into a subset representing the needed privacy leakage profile and then compile the resulting source code to a mobile app. The compiler may be implemented as part of the MDM as well.

### 3.3.5 Reconfigurability Engine

Any instance of PPAndroid-Benchmarker can only cover a limited number of sources and sinks and limited settings for benchmarking profiles. To allow extension of supported features and reconfiguration of the system (including removing some unwanted features), a Reconfigurability Engine can also be developed.

A major part of the Reconfigurability Engine is addition and removal of sources and sinks. This can be achieved by defining a dynamic list of sources and sinks for PPAndroid-Benchmarker to process. The dynamic list needs to support both descriptions of sources and sinks and also code for accessing the sources and sinks. One way of supporting such a dynamic list is to have an XML file for the sources and another one for the sink, and the binary code for accessing each source and sink is provided in the form of an executable plug-in following a defined API. Another
way of achieving this is to provide source code of new sources and sinks directly with description files, and a compiler is used to re-compile the whole system into a new instance of PPAndroid-Benchmarker.

Another part of the Reconfigurability Engine is changing how the system behaves e.g. how to automatically configure some privacy protection apps requiring human intervention, which can be achieved by defining other configuration files or APIs so that different types of plug-ins can be added.

Another major part of the Reconfigurability Engine is to add and reconfigure more information processing operations and tricks against static and dynamic analysis techniques. This prototype has included a number of tricks mainly for testing TaintDroid. Adding more will require a different type of API and plug-in system so that any operation can be added between any pair of source and sink, which will need to work along with the API/plug-in systems for sources and sinks.

### 3.3.6 Interaction between Components

The interactions between different components can be explained by telling how a typical benchmarking task looks like. At the beginning, the user will fill a test profile to tell the benchmarker about details of the test. It will include information such as types of leaked data, mobile device specifications, privacy apps to be tested and others. The profiles are written in XML as stated. The profile creator keeps XML files in the Profiles Database in order to be used for analysis. Then the Benchmarker App starts communicating with the MDM to request installing each tested app. The MDM searches for the required app and installs it. MDM keeps records of installed apps in the Apps Repository for future use. Once the Benchmarker App receives a signal of starting the test, it will initiate leakage attempts to the Drop-in Server. The latter keeps the results as XML profiles in the Results Database.

### 3.4 Experimental Setup

In this section, the method that has been followed to set up the experiment is explained. It starts with describing how tested apps were collected, followed by how special apps were handled. Lastly, an explanation of variant settings and implementation is provided.

### 3.4.1 Selection of Privacy Protection Apps

The first step of the experiment was to identify and collect Android apps with some real-time privacy protection features. These tools were collected from variant sources. Many of them were gathered from Google Play store as it is the main source for Android apps. The following steps were taken to collect the apps. Firstly, Google’s search engine was used to look for related privacy apps. Many keywords were used in this step. For example, “privacy”, “security”, “private”, “protection”,

“leak”, “dynamic analysis”, “static analysis”, “leakages” and several others were used. Secondly, some major third-party Android markets have been explored such as Amazon Appstore, GetJar, Slide ME, F-Droid, Samsung Galaxy Apps, AppsLib, Mobogenie and a few others. Thirdly, many related apps have been collected from cyber security product vendors and service providers. The list of these vendors were taken from AV-Comparatives website [98]. There are around 50 mobile security companies like AVG, AegisLab, Bitdefender, etc. Lastly, some data leakage protection and Android forensics tools are included such as TaintDroid, NowSecure forensic tool [99], PrivacyProtector app reported in [100]. All privacy protection apps that have been found were targeted to be tested but not all of them are available or provide real-time protection. At the time of this writing, in total 165 privacy protection apps have been collected and tested. According to the sources, these tools can be categorised as follows:

- Apps dedicated for privacy protection (from Google Play),
- security apps with privacy protection features (from Google Play),
- apps from third-party markets,
- security vendors’ apps (those not covered in the above categories) and
- privacy related apps developed by researchers.

Functionally speaking, those tools can be categorised into three different groups:

1. apps that try to detect privacy violations at installation time,
2. apps that detect privacy violations based on blocking access to sensitive information sources,
3. real-time dynamic monitoring tools requiring changes to the Android system.

3.4.2 Testing Procedure and Settings

In this experiment, the testing procedure covers three different scenarios: fully automated testing without user intervention, semi-automated testing with user intervention, and testing access-related analysis apps. For the fully automated scenario, the tester (as a user) is involved to select target apps for testing and define the benchmarking tasks only. For the semi-automated scenario, the tester is also involved in the process of installing process because some apps require manual configuration before they can run properly. For the last scenario, the Benchmarker App will attempt to access private sources first. Then, if the access is granted, the app will proceed with the actual leakage.

PPAndroid-Benchmarker has been equipped with some options to increase the configuration power. For instance, the user can set time-outs for the evaluation test.
Moreover, the user can set wait times between data acquisition by the benchmarker and data leaking attempts.

The architecture of PPAndroid-Benchmarker is intended to work in an automated manner as much as possible including automatically downloading and installing a target app. However, download links of some apps cannot be automatically determined, so the apk files must be provided manually by the user. Moreover, some apps are marked for manual configuration and initialisation. For instance, some apps require manual registration, accepting terms or connecting to the cloud. To ensure all target apps were tested with the most appropriate configuration settings, settings with and without user intervention were manually tested and then labelled each app with the best setting accordingly.

While testing apps which require Root access, it has been noticed that there is a need of different settings. In this case, PPAndroid-Benchmarker is programmed to also record if access to each target data source is blocked. That allows PPAndroid-Benchmarker to know if a recorded failure of a privacy leakage attempt was blocked at the access level or at a later stage.

Most apps can be tested with PPAndroid-Benchmarker without a special treatment. However, there are a few tools that must be tested using a different procedure or special settings. For example, TaintDroid does not work as a stand-alone application. It involves building a custom-built operating system on the tested device/emulator. Thus, a customised version of Android was built to test TaintDroid against PPAndroid-Benchmarker in an emulator. Accordingly, Anti-TaintDroid (ScrubDroid) was tested in the same environment, too.

In this experiment, it has been decided not to test apps based on traffic analysis. Testing such apps requires some significant changes to the architecture and procedures of PPAndroid-Benchmarker, and this was left as a future work.

### 3.4.3 Special Benchmarking Profiles

In this subsection, some special benchmarking profiles are discussed which were used in the experiment.

#### Baseline Benchmarking Profile

The Android system itself may already have some privacy protection mechanism so that some privacy leakage attempts can be detected and blocked at the operating system level. In this experiment, a baseline profile was always ran without any third-party privacy protection app first. When each privacy protection app was tested, only those new successes in detecting privacy leakages were counted. For the Android version that has been ran in this experiment, none of the privacy leakage attempts were detected by the Android system, but this may change in future versions.
Probing Phase

As mentioned above, some apps may require user intervention. A probing phase was therefore added to test if an app behaves differently with and without user intervention. Figure 3.3 shows how this phase was conducted.

![Figure 3.3: PPAndroid-Benchmarker in probing phase](image)

Access-Based Apps

Some apps block privacy leakages by blocking access to sensitive information sources. Therefore, a special benchmarking task is added to test if a privacy leakage attempt is blocked at the source access level or afterwards. The Benchmarker App will access each source in the benchmarking profile and check if there is a response from the tested app, and if the access goes through it will proceed with the actual leaking attempt.
3.5. RESULTS & ANALYSIS

Table 3.1: Testing installation-time privacy apps

<table>
<thead>
<tr>
<th>Sensitive Information Source</th>
<th>Installation-time apps</th>
<th>Device IDs</th>
<th>Accelerometer</th>
<th>Contacts</th>
<th>Location</th>
<th>SMS</th>
<th>Files</th>
<th>Web History</th>
<th>Call Log</th>
<th>Camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation-time apps</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

"F" means the tested app failed in detecting the leakage attempt.
### Table 3.2: Testing privacy apps that require root access

<table>
<thead>
<tr>
<th>App Name</th>
<th>Vendor/Developer</th>
<th>Sensitive Information Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECUREit</td>
<td>Lenovo</td>
<td>B++</td>
</tr>
<tr>
<td>X Privacy</td>
<td>Marcel Bokhorst</td>
<td>B+</td>
</tr>
<tr>
<td>LBE Privacy Guard</td>
<td>Lamian</td>
<td>B</td>
</tr>
<tr>
<td>360 Mobile Safe</td>
<td>Qihoo 360</td>
<td>B</td>
</tr>
</tbody>
</table>

- 'F' means the tested app failed in detecting the leakage attempt.
- 'B' indicates that the privacy system blocked access to the information source.
- 'B+' refers that the privacy system is able to go beyond blocking access by faking the leaked information.
3.5 Results & Analysis

This section presents the analysis results gathered from testing collected apps and tools. It starts by demonstrating the probing phase results. Then, the findings from the three functional categories of privacy protection apps are reported, respectively, since the behaviours of apps in each category are different.

3.5.1 Probing Phase Results

During the study, some apps were noticed that they need to be configured before using to ensure all included privacy protection means and settings are enabled. A small experiment was conducted for nine apps, selected from distinct categories. Seven out of nine asked for user input in variant ways. For example, some asked the user to accept terms, to slide a few ad pages, to register on-line or to wait for configuration. The time spent to set up each app after installation were also calculated. Table 3.3 shows the findings of this small pilot study. As a consequence, the actual benchmarking tasks were conducted in two approaches depending on if user intervention is needed: fully automated without user intervention and semi-automated with user intervention.

Table 3.3: Probing apps requirement of user-intervention

<table>
<thead>
<tr>
<th>App No.</th>
<th>Set-up time in seconds</th>
<th>Interaction required</th>
<th>Type of interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.1</td>
<td>Yes</td>
<td>Accept terms/ Connect to cloud</td>
</tr>
<tr>
<td>2</td>
<td>26.5</td>
<td>Yes</td>
<td>Accept policy/ Configurations</td>
</tr>
<tr>
<td>3</td>
<td>8.5</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>22.5</td>
<td>Yes</td>
<td>Slide some ads/ Configurations</td>
</tr>
<tr>
<td>5</td>
<td>8.8</td>
<td>Yes</td>
<td>Accept policy/ Configurations</td>
</tr>
<tr>
<td>6</td>
<td>13.0</td>
<td>Yes</td>
<td>Accept terms/ Upgrade offer</td>
</tr>
<tr>
<td>7</td>
<td>76.3</td>
<td>Yes</td>
<td>Accept terms/ Set a code/ Register</td>
</tr>
<tr>
<td>8</td>
<td>57.4</td>
<td>Yes</td>
<td>Click start/ Setting/ Slide ads</td>
</tr>
<tr>
<td>9</td>
<td>18.0</td>
<td>No</td>
<td>-</td>
</tr>
</tbody>
</table>

3.5.2 Benchmarking Results

In this subsection, results of the three functional categories are shown. These are static analysis apps, apps require Root and dynamic analysis apps.

Static analysis based apps

A majority of the mobile privacy protecting applications collected from different app markets belong to the first category. Apps in this group are only capable of inspecting privacy-related features of an app at installation time. They either
Table 3.4: Testing TaintDroid against PPAndroid-Bencmarker

<table>
<thead>
<tr>
<th>Anti-TaintSensitive Information Source</th>
<th>Trick Device IDs</th>
<th>Accelerometer</th>
<th>Contacts</th>
<th>Location</th>
<th>SMS</th>
<th>Files</th>
<th>Web History</th>
<th>Call Log</th>
<th>Camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple encoding</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>shell command</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>file+shell hybrid</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>time keeper</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>simple encoding</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>False</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Note: ‘F’ means the tested app failed in detecting the leakage attempt.

'S' means the tested app succeeded in detecting the leakage attempt.

Anti-Taint tricks cannot be applied to camera photos, as they are designed to process strings only.
apply a permission analysis, statically analyse installed apps or upload examined app to a sandbox to test it dynamically. Some tools notify the user of reported malicious apps if installed. Nevertheless, none of them reported the Benchmarker App of PPAndroid-Benchmarker as a malicious app. Therefore, all tested apps did not block or detect any of the leaking attempts. Table 3.1 clarifies that where ‘F’ indicates the failure of tested apps to detect privacy violation. The results are expected and demonstrated that PPAndroid-Benchmarker worked as designed.

In this group, approximately %75 of the apps were tested semi-automatically with user intervention and the rest was tested in a fully automated manner.

**Privacy apps requiring Root access**

The second category covers apps that block access to defined sensitive data sources on mobile devices. All apps in this group require the Root access in order to reject other apps’ requests of accessing sensitive information sources. In PPAndroid-Benchmarker testing, five commercial apps were found to provide such a functionality: **SECUREit**, **X Privacy**, **PDroid Privacy Protection**, **LBE privacy guard** and **360 Mobile Safe**. The last two apps are only available in Chinese. One research app developed by Li et al. [100] also falls into this category.

Table 3.2 shows the result of testing apps in this category. Those apps are different in terms of how much they protect the user. As illustrated, each app has its own fixed list of predefined private information sources. The table shows how each app responded to privacy leaks of variant sources. ‘F’ indicates that the app did not react against that leakage, where ‘B’ means the attempt was blocked. **X Privacy** clearly is more diverse than others in covering many sensitive sources. It is also capable of faking the mobile device identifier IMEI (represented as ‘B+’ in Table 3.2). This app is the only one in this group able of providing an option beyond blocking attempts. Hence, **X Privacy** gives three options when the IMEI is being accessed: block, fake, and allow. All the apps in this group were tested in the semi-automated with user interaction mode on a rooted device. The reason is that they need to be granted with root access to function. Additionally, most of them require the user to set up the app and interact with some interfaces before they are ready.

**Real-time privacy monitoring apps**

The last category covers only a few apps developed by researchers. For this category TaintDroid was tested since it is the basis of many other solutions. Table 3.4 illustrates the result of benchmarking TaintDroid with the presented system. It shows that TaintDroid succeeded in detecting privacy leakages in real time, as ‘S’ shows in the table, for its predefined list of sensitive information sources. Each time that it has been tried to leak a piece of sensitive information, TaintDroid triggered a notification showing the tainted tracked data alongside with the leaking app and some other details. Some information sources, notably ‘files’, ‘call log’
and ‘camera’, are not included in TaintDroid’s list, so the privacy leakage attempts were not detected. When the information leakage was attempted using anti-tainting tricks, most of leakage attempts were not detected, except for for the ‘direct buffer’ and ‘lookup table’ attacks.

TaintDroid in this work was tested in a customised emulator since it requires a modified version of Android operating system to run. TaintDroid was tested in a fully automated manner since no user intervention is required in the benchmarking process.

3.6 Discussions

In brief, this experiments proved that PPAndroid-Benchmarker is efficiently and correctly working as designed. The overall results from the benchmarking experiment can be summarised as follows: static analysis based apps when tested all failed to detect any real-time leakage attempts, privacy apps that require Root access could block access to some sensitive information sources, and TaintDroid (representing dynamic analysis based methods) could detect most of the leakage attempts except for three information sources tested but it failed when most anti-tainting tricks were applied.

As a general observation, the more “intrusive” a privacy protection tool is, the more powerful it can do to detect and prevent privacy leakage attempts. What is meant by “intrusive” is how much changes a tool requires from the operating system, ranging from the most intrusive to the least: rebuilding the operating system, requesting Root access to the operating system and hooking into the operating system for checking new apps at the installation time. However, even for the most intrusive tool, TaintDroid, there are still privacy leakages that cannot be detected. This implies that the best approach is for the operating system itself to provide native support on privacy leakage detection and prevention, so that the intrusiveness will not be an issue any more. If that happens, PPAndroid-Benchmarker will still be able to benchmark the built-in privacy protection mechanism since it simply simulates what malicious apps are doing. As a matter of fact, in the current implementation, a baseline benchmarking profile is always run without any third-party privacy protection apps so it is easy to identify what privacy leakage attempts can be detected by the Android system itself.

Some could argue, why PPAndroid-Benchmarker does not give a numeric or categorical rating for tested apps. Having a way of rating privacy protection mobile apps in terms of their general performance can be very useful for end users, and is a topic for the future research. A possible way of giving scores or ratings through PPAndroid-Benchmarker is to count how many sensitive sources are protected by the tested tool, but some issues need to be carefully considered. First, not all sensitive information leakages are considered privacy violation [101]. Some sensitive information is transmitted out of the device for the sake of providing better services for the user. For instance, collecting users’ precise location can help an app give more
3.7 Conclusion and Future Work

In this chapter, PPAndroid-Benchmarker is presented, a benchmarking system for evaluating performance of mobile privacy protection apps. This system allows to benchmark privacy protection apps designed for the Android platform. To the
best of the author knowledge, PPAndroid-Benchmarker is the first of its kind. PPAndroid-Benchmarker has been tested on 165 privacy protection apps belonging to three different functional categories to demonstrate PPAndroid-Benchmarker’s effectiveness in evaluating their performance in detecting privacy leakage attempts. It is believed that the experiment that has been conducted is also the first of the kind as many previous efforts are about benchmarking privacy risks of normal apps rather than privacy protection apps. The results showed that real-time dynamic monitoring tools like TaintDroid is the best approach, which is not a surprise since such tools require the most changes to the underlying operating system.

For future work, there are a number of improvements that can be made on PPAndroid-Benchmarker. The most important components to add to the presented prototype are the Test Apps Generator and the Reconfigurability Engine discussed in Sections 3.3.4 and 3.3.5. Adding these two components is not technically difficult, although it is necessary to decide carefully how to make them more usable to end users of PPAndroid-Benchmarker. Another interesting feature to add is to incorporate the system with an Off-line Analyser. This component can be designed to collect both testing profiles and benchmarking results with the aim of producing more visualised results to facilitate understanding of tested privacy protection apps and comparing their performance. It can also produce one or more ratings to reflect the level of privacy protection of each tested app. Furthermore, another future work is to add a benchmarking profile for dynamic behaviour apps. Some privacy protection apps have dynamic behaviours. There are some privacy protection apps connected to a cloud and update their data every while. Other apps may have an intelligent way of adapting their behaviours, e.g. it may use machine learning techniques and improve its responses to privacy leakage attempts of malicious apps. To properly benchmark such apps, PPAndroid-Benchmarker need to run the benchmarking task for a significantly long period of time and capture a number of snapshots of the tested app’s behaviour, and then see if some changes can be observed.
Chapter 4

User Perceived Trust of Mobile Apps’ Privacy Ratings

4.1 Introduction

Mobile apps are the soul of mobile devices that are the sources of functionalities and features. Variety and sophisticated functions that apps offer today require access to sensitive sources of information on the phone. However, apps can be poorly or intentionally designed to violate the user’s privacy, so mobile users must pay attention when selecting apps. On the other hand, it is hard for app stores keeping track of all apps to identify malicious and legitimate apps especially that their statistics are remarkable. The number of available apps in Google Play Store from December 2009 to December 2017 has been dramatically increasing. According to Statista [102], it was very recently placed at 3.5 million apps in December 2017, after actually surpassing one million apps in 2013. The same website announced that during May 2016 there were 65 billion apps downloaded from Google Play, reported from the online company. Over and above, a report in 2017 showed that daily US users usually spend 2.3 hours with digital media on mobiles [103].

Earlier in the literature review of this thesis, privacy risks of mobile devices have been presented. One of the risks is the consequences of private information leakages of malicious apps. Based on the fact that apps are ubiquitous and essential, users need to be informed about security and privacy issues of apps. To solve that, one of the elements to look at is the ratings associated with apps that can help users to make decisions about apps to install and use. App ratings mainly aim to provide users with feedback about the app either in terms of security, privacy, performance or any sort of experience that can be encountered in use.

Ratings of apps in general can be classified into subjective and objective ratings according to the extent of the direct human involvement in the evaluation process,
although the distinction between both classifications can be more complicated. In this thesis, the focus is on mobile privacy leakage issue. Hence, this work concentrates on privacy ratings in studying users’ perceived trust. Moreover, the following definitions are used for a given app: 1) an objective rating is an automated (i.e., machine-generated) evaluation of the app in terms of privacy; 2) a subjective rating is human-judged evaluation of the app’s privacy level.

Although there is some evidence of the usefulness of objective app privacy ratings in motivating users to select more secure apps, how human users perceive objective ratings and how such ratings influence their behaviours (especially when the objective ratings contradict with subjective judgements of their own or what they heard from human experts) remain largely unstudied. Since any contradiction can potentially cause a loss of trust of users, understanding users’ trust in both objective and subjective app ratings could offer ways of improving the design and evaluation of the related systems. The determinants of trust in the human decision-making process have been studied in various contexts [104,105], however, to the best of the author’s knowledge, they have not been systematically investigated in the context of mobile apps research.

To fill the above-mentioned gap, a user study on users’ perceived trust in subjective and objective privacy ratings of mobile apps has been conducted. Data collected from over 1,000 crowdsourcing workers revealed that: 1) the user’s own subjective perception of app privacy could heavily influence their perceived trust in subjective and objective ratings of mobile apps; 2) there are different user-specific behavioural patterns in the reported trust in subjective and objective ratings of mobile apps; 3) users’ trust in app ratings is dependent on the app category; 4) no sufficient evidence that users’ trust in app privacy ratings is influenced by demographic factors such as skill-level and gender except the age which has proven to play a significant role on users’ self-reported trust.

The rest of the chapter is organised as follows. Section 4.2 presents selected work closely related to the work reported in this chapter. A detailed description of the user study is given in Section 4.3, which is followed by data analysis and results in Section 4.4 and more discussions on the main findings in Section 4.5. In Section 4.6, some limitations of this study are presented. The last section concludes the chapter with future work.

### 4.2 Related Work

In this section, related work is reviewed. It starts by describing the role of trustworthiness, followed by showing evidence on the impact of subjective and objective feedback on decision making.
4.2. RELATED WORK

4.2.1 The Role of Trustworthiness

Trust in the mobile apps context can be important in predicting to what extent ratings will affect users’ decision-making processes related to mobile app privacy. Huang et al. [106] indicated that low perceived security may cause users to reject the use of IT systems, while high perceived security may result in engaging insecure practices. This would imply that risk perception, which can be in relation to trust, can greatly affect users’ decisions and behaviours. To the best knowledge of the author, no work touched the effect of trust in human decision-making in the context of mobile apps ratings. However, in social sciences, trust is often explained as an individual’s readiness for a vulnerable circumstance as a result of a positive expectation of others’ actions [105], where “others” can be other people or automated systems. It is a general understanding that the interpretation of trust is context-dependent and research on it requires a multi-dimensional approach in the right context [107].

Trust characterisation, factors and influences have been also studied. Often, researchers attribute trust to three factors of a trustee: 1) expertise or ability: to what extent a trustee is considered to be competent, 2) trustworthiness: the level of cooperation and kindness that a trustee is seen to be, and 3) honesty: the integrity level that the trustee is believed to be [105, 108]. Trust can be also influenced by other factors such as conditional reasons and personality traits [104, 105, 107]. Moreover, another work has shown when judging others’ trustworthiness through Facebook profiles, observers rely on some meaningful cues e.g., number of friends, number of comments and likes, smiling profile photographs, etc [109].

4.2.2 Impact of Subjective and Objective Feedback on Decision Making

To the best of the author’s knowledge, very limited work has investigated the influence of subjective and objective feedback and ratings on users’ decision making processes. Some limited research has been done in the areas of marketing, business and health care. Lynn [110], as an example, explained how subjective messages are trusted more than objective messages in relevance to the rise of venereal diseases.

In another related work, some researchers from the marketing research community examined the influence of message board persuasion in terms of subjective messages and/or objective messages. They found that using objective messages is more efficient than subjective ones. Yet, combining the two types did not improve the result from using either subjective or objective alone [111].

The closest related work in the area of cyber security was done by Chen et al. [112]. A user study was conducted to investigate the impact of risk (negative) and safety (positive) information in mobile apps’ summaries on users’ decisions on app installations. They suggested developing a validated risk/safety index for mobile apps. That could be used to improve users’ decisions when they install new apps, specially when that index is framed in terms of safety.
4.3 User Study Design

Mobile users’ trust in mobile apps appears to play an important role in predicting either they depend on subjective or objective rating to form their decisions about apps. As a consequent, this work sheds light on how trust affects human users’ decision making processes in the context of mobile apps’ privacy ratings. In fact, this experiment tends to find out which is the most trusted evaluation source of apps privacy, objective (from automated computer programs), subjective (human experts opinions) evaluation of app privacy level or the user own judgement. Ultimately, the user study was designed to answer research questions in the following subsection.

4.3.1 Research Hypotheses

This user study has been designed to investigate users’ trust in subjective and objective privacy ratings of mobile apps. The expectation was that users’ self-reported trust in this context is dependent on several factors. These factors can be either linked directly to human aspects such as their characteristics and experience, or to non-human aspects such as the app category, and the type of feedback given ‘subjective/objective rating’. In this piece of work, the following five hypotheses were formulated to be tested:

- **Hypothesis 1 (H1):** users’ own subjective judgement on app rating plays a significant role in users’ self-reported trust in subjective and objective ratings.

- **Hypothesis 2 (H2):** users’ self-reported trust in subjective and objective ratings is user specific.

- **Hypothesis 3 (H3):** users’ self-reported trust in subjective and objective ratings is dependent on app category.

- **Hypothesis 4 (H4):** demographic factors such as age, gender and skill level play a significant role in users’ self-reported trust in subjective and objective ratings.

- **Hypothesis 5 (H5):** users’ self-reported trust in subjective and objective ratings differs based on the context.

4.3.2 Procedure

To test the above hypotheses, a within-subjects crowdsourcing-based experiment was conducted. At the beginning of the user study, a brief overview was given to the participants explaining the meanings of subjective and objective ratings. The study was designed to be completed within 30 minutes. The study was structured
4.3. USER STUDY DESIGN

in three sessions\(^1\). Participants' demographics including age, gender, and their computer skill levels were collected in the first session. Then, six apps with their subjective and objective ratings (see Section 4.3.3) were presented to participants without revealing any details about the presented apps. The six apps were displayed in a completely “anonymous” form (shown as “App 1”, “App 2”, etc.) in the second session. In the third and fourth sessions, app type was revealed such as “Anti-virus 1”, “Anti-virus 2”, “Messenger 1”, “Messenger 2”, etc. Each participant was asked to complete 18 questions (one question per app), a third of the questions on “anonymous” apps and the other two-thirds are evenly on Tool and Social apps (Section 4.3.3 describes how categories were chosen). The apps were shown to the participants in a random order and participants were asked to focus on a single app at a time.

Table 4.1: Apps names used in our experiments and their privacy ratings, shown in 4-point scale

<table>
<thead>
<tr>
<th>Hidden</th>
<th>Mobile apps</th>
<th>Privacy ratings</th>
<th>Social</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Objective</td>
<td>Subjective</td>
</tr>
<tr>
<td>App1</td>
<td>Antivirus1</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>App2</td>
<td>Antivirus2</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>App3</td>
<td>Antivirus3</td>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td>App4</td>
<td>Antivirus4</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>App5</td>
<td>Antivirus5</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>App6</td>
<td>Antivirus6</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

\(^a\) The 4-point scale of app privacy rating was chosen to be in line with the strength rating scales used for the password sessions (see Footnote 1).

For each app, participants were asked which rating they trust more. Dummy app names was presented to users to avoid complicating the study with other factors (see Section 4.3.3). Users were informed that objective scores are generated by an algorithm developed by a group of human security experts, whereas subjective scores are generated by a group of human security experts. Participants were offered four choices of answers ("Subjective Rating", "Objective Rating", "Neither", and "Undecided"), and they had to select one of the four. The Neither option implies that a user disagrees with both objective and subjective privacy scores, where "Undecided" means that the participant is unable to make a choice.

All apps with their ratings were shown to participants in the same order as shown in Table 4.1. The order was not randomised for two reasons: 1) to get an even randomisation of app orders, this would require a larger sample size of participants

\(^1\) In our user study, we actually included two more sessions on password strength ratings, which is reported in a separate thesis (Chapter 3 [113]). To counterbalance the learning effects, participants were randomly divided into two groups, one was asked to do the password tasks first and the other the mobile app tasks first. Our analysis showed that the data from both groups were consistent so we used all participants for our data analysis reported in this thesis.
CHAPTER 4. USER PERCEIVED TRUST

(a) Hidden Apps  
(b) Displayed Apps ‘tool’  
(c) Displayed Apps ‘social’

Figure 4.1: A screen-shot of “anonymous” and displayed app questions and rating options

and additional expenses can be avoided if the apps order is fixed; 2) only dummy names of apps were displayed to avoid complicating the study with many factors that might affect user responses. The only information revealed to the user is the app type in the second sessions. Therefore, it does not make sense to show the user “Messenger 6” before “Messenger 3” for instance. However, participants were allocated to different orders of displaying password and app sessions to exclude the potential influence of the order of displayed sessions. Furthermore, to minimise the potential influence of the order of answer options in each question, the order was randomised.

After a participant made a choice for one app, another follow-up question was asked for the participant to justify his/her decision. A set of predefined justifications were displayed to help guide the participant. These presented reasons are shown according to user previous selection, meaning that there is a set of reasons for each choice of trust. Table 4.2 depicts the reasons that were shown to participants. Moreover, there was a space for the users to construct his/her own reasoning in a free-format text area. This part of the design was added to investigate how participants made their decisions of the perceived trust.

Like typical crowdsourcing based user studies, we did not collect any personal or sensitive data. The user study was reviewed by the University of Surrey’s University Ethics Committee (UEC) and a favourable ethics opinion was secured before running the user study.

4.3.3 Our Choice of Mobile Apps

In this experiment, eighteen pairs of objective and subjective rating combinations were used. There were six apps displayed to participants in each session. Table 4.1 shows the apps that were used along with their privacy scores. In the following, the considerations of how apps’ privacy scores were determined are discussed.
### Table 4.2: Predefined answers for users’ reasons behind each choice of rating

<table>
<thead>
<tr>
<th>Choices</th>
<th>ID</th>
<th>Reason Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>O1</td>
<td>Automated algorithms can detect hidden things better than humans.</td>
</tr>
<tr>
<td></td>
<td>O2</td>
<td>Humans often make mistakes while automated algorithms always do what they are designed for.</td>
</tr>
<tr>
<td></td>
<td>O3</td>
<td>I don’t believe human experts can always agree with each other.</td>
</tr>
<tr>
<td></td>
<td>O4</td>
<td>The objective rating matches my expectation.</td>
</tr>
<tr>
<td></td>
<td>O5</td>
<td>The objective rating is lower and I want to be at the safer side.</td>
</tr>
<tr>
<td></td>
<td>O6</td>
<td>It is just my feeling and I can’t explain it.</td>
</tr>
<tr>
<td></td>
<td>O7</td>
<td>Others.</td>
</tr>
<tr>
<td>Subjective</td>
<td>O8</td>
<td>Automated algorithms cannot predict new form of attacks.</td>
</tr>
<tr>
<td></td>
<td>O9</td>
<td>Automated algorithms are designed by human experts but they won’t do as well as human experts.</td>
</tr>
<tr>
<td></td>
<td>O10</td>
<td>Computer programs can contain errors and bugs so the results may be misleading (although not intended by human designers).</td>
</tr>
<tr>
<td></td>
<td>O11</td>
<td>The subjective rating matches my expectation.</td>
</tr>
<tr>
<td></td>
<td>O12</td>
<td>The subjective rating is lower and I want to be at the safer side.</td>
</tr>
<tr>
<td></td>
<td>O13</td>
<td>It is just my feeling and I can’t explain it.</td>
</tr>
<tr>
<td></td>
<td>O14</td>
<td>Others.</td>
</tr>
<tr>
<td>Neither</td>
<td>O15</td>
<td>Neither ratings match my expectation.</td>
</tr>
<tr>
<td></td>
<td>O16</td>
<td>It is just my feeling and I can’t explain it.</td>
</tr>
<tr>
<td></td>
<td>O17</td>
<td>Others.</td>
</tr>
<tr>
<td>Undecided</td>
<td>O18</td>
<td>I need more information to make a proper choice.</td>
</tr>
<tr>
<td></td>
<td>O19</td>
<td>Others.</td>
</tr>
</tbody>
</table>

### Different App Categories

The study was designed in a way that represents different apps categories and variant privacy levels. Three categories were chosen to allow studying user perception in variant contexts. In a pilot study, we decided to use three categories: anti-virus tools, instant messaging apps, and mobile games. However, games were later removed in the formal study as they contain too many sub-categories and it was hard to classify them consistently (which we left for future research). The remaining two categories represent two representative classes of apps: security tools like anti-virus apps are supposed to be trusted as they meant to protect security of mobile devices, while social tools like IM apps allow users to share information including personal and sensitive information so they can be perceived privacy sensitive by many users.
Different Subjective and Objective Ratings

The subjective and objective privacy ratings are simply allocated to apps in a way fulfils the study goals. Dummy app names were used to avoid potential impact of factors such as app name, user’s familiarity with the app, attractiveness of the icon, etc. Therefore, any privacy ratings can be given to apps without limitations.

The allocated ratings are believed to be reasonable to some extent, although there might be some disagreement of how they were selected. In early stages of the design, limited cases were tested which represent differences between subjective and objective privacy scores. In that case, participants were asked to select what they trust based on the rating value and type. Yet, a few cases were added, where subjective and objective ratings are made equal to study user behaviour in this special condition. In addition, the difference between objective and subjective ratings were made logical, so the participant does not feel it is not realistic. Therefore, that difference was one- or two-point difference on the 4-point scale for each rated app. In this way, the potential effect of illogical ratings can be controlled, to avoid having trust issues of the user-study. Furthermore, the design has taken into account having a variant pair of subjective and objective ratings for each app to provide a more uniform distribution of apps’ ratings.

4.3.4 Recruitment

The CrowdFlower crowdsourcing platform [114] was used to recruit participants in this study (in April 2018 CrowdFlower was renamed to Figure Eight). Each participant was rewarded $0.6 for the whole study (including the password strength tasks which are not covered in this thesis)\(^2\).

The recruitment only involved the most trusted workers rated by CrowdFlower (the so-called ‘Level 3’ workers) to maximise data quality. The whole user study is split into a number of parts and ran on different days and at different times to recruit participants from different geo-locations and working times.

4.3.5 Quality Assessment

Some strategies are needed to ensure the quality of the collected data due to the reason that crowdsourcing based studies can incur a lot of noise in the data. Therefore, outliers, as anomalies, inconsistencies and deviants in data, must be detected to achieve high reliability. The following paragraphs explain the participants pre-screening step, followed by how outliers were detected.

In the pre-screening step, as mentioned in the previous subsection, the recruitment only accepted Level 3 workers. To elaborate, CrowdFlower provided a feature that allowed selecting workers based on trust annotations. These annotations are

---

\(^2\)Note that on crowdsourcing platforms rewards for such micro tasks are mostly very cheap. $0.6 was agreed based on similar tasks on the same platform to avoid under- or over-motivating participation.
acquired from groups of other trusted annotators. Also, each worker (participant) has a trust level or reputation which is built based on the previously accomplished jobs.

Post the data collection, three detection rules were constructed to detect outliers. These are built based on workers’ actual responses. The following set of rules were followed to ensure the reliability of approved answers.

1. Unfinished tasks were excluded from the results. This was done to ensure that all questions are answered in both sessions.

2. Only tasks that were completed within a reasonable time were accepted. First, we calculated the average duration to complete the task from a small pilot study included ten participants and the actual average duration from our crowdsourcing experiment (approximately 8.5 and 11 respectively). Using k-means clustering, we split the participants into 3 groups based on their task completion time. The smallest group where participants spent less time than others are treated as outliers. Specifically, we sat a threshold value based on the clusters and the average spent time to accept participants work. Precisely, the threshold we used was 4 minutes.

3. Conflicting answers from the same participant were considered from outliers, and therefore excluded. In this rule, outliers were identified whenever their given reasoning deviated from the selected rating they chose to report trust. This step was manually done.

These approaches would produce large errors if it is applied too rigidly. In other words, the above approaches could exclude participants who are not necessarily outliers since a honest participant may still produce a few errors unintentionally. Such unintended human errors would be expected and should not degrade the trustworthiness of their responses as such mistakes would happen without intention and can be neutralised by other participants' answers. Therefore, it would be better to not exclude those who made only a few conflict errors. Then, this would require calculating the average human error rate in a separate study and use this to determinate the threshold value of acceptable human errors.

4.4 Results

In this section, results of this crowdsourcing based experiment are illustrated. This section starts with analysis of the participants. Afterwards, the observed behavioural analysis is described on different levels: app condition, behavioural patterns, demographic factors and app dependencies. Finally, the section ends with elaboration of reasons behind trust analysis.
4.4.1 Participants

Based on a statistical power analysis, the required sample size for this user study is 293. This sample size is the outcome of an acceptable power of 0.95, given an expected medium effect size of 0.5 and \( p < 0.05 \). Ultimately, the sample size \( (N = 1077) \) exceeds the desired number of subjects which in turn would have a significant power.

In total, 1,100 participants took part in the presented user study. 23 participants were excluded because they took the experiment more than once. Therefore, the data that have been analysed were based on \( 1100 - 23 = 1,077 \) participants from 68 countries. The male-female ratio was 66% to 34%. The participants had a reasonably wide age range: 26-35 (39.4%), 18-25 (30.6%), 36-45 (20.8%), over 45 (8.6%), and below 18 (0.6%) (they were excluded as it is against the ethics approval of this study). Due to the relatively small number of participants in the last two age groups, participants were re-grouped to have a more even distribution: 25 or below (336 participants), 26-35 (424 participants), 36 or older (317 participants). Figure 4.2 shows participants distribution by age, computer skill levels according to their gender.

In the following subsections, the main results of the presented user study are illustrated. For the four answer options for each app, 2-character short names are defined: su for subjective rating, ob for objective rating, ne for neither and ud for undecided.

4.4.2 Behavioural Analysis

In this subsection, an analysis of participants’ behaviours is demonstrated at variant levels. First, it shows how the app category, when it is revealed, affected users trust. Then, it illustrates the observed behavioural patterns of participants. After that, this part continues with showing the effect of demographic factors on users’ self-reported trust. Lastly, it investigates if participants’ behaviours are influenced by different rating combinations.

App Category Condition

First, the impact of the display condition of the app category (apps category hidden or shown, as a binary independent variable) on participants’ self-reported trust in subjective and objective ratings is reported. Here, the dependent variable is the 4-valued answer of each participant (i.e., the self-reported trust). Among the four values, our main interest is on su and ob, but we will also look at ne and ud since they can reflect how participants felt about the shown subjective and objective ratings.

In order to visualise the collective behaviour of all participants, 3-D histograms of participants’ behavioural patterns for both hidden- and displayed-app conditions have been produced as shown in Figs. 4.3, 4.4 and 4.5, where the bin in the 3-D
4.4. RESULTS

Figure 4.2: Age (a) and computer skill levels (b) distribution of participants based on their gender

histogram at position \((i, j)\) indicates the number of participants who chose subjective \(su\) \(i\) times and objective \(ob\) \(j\) times \((0 \leq i + j \leq 6)\). Each bin in the 3-D histogram denotes a specific behavioural pattern of a number of participants in terms of a participant’s tendency to trust subjective or/and objective ratings.

The data represented in the 3-D histograms has different dimensionalities. For the reason that this study aims to compare subjective vs. objective factors, the graphs in Figs. 4.3, 4.4 and 4.5 focus on two dimensions which are subjective and objective app ratings. However, the data can be seen from different angles such as representing the collective behaviour based on given apps’ scores. This will be for future work as it goes beyond the scope of this study.

From Figs. 4.3, 4.4 and 4.5 it can be seen that the participants’ collective behaviour was not significantly affected by the app category display condition. Despite the apps category information was hidden or displayed, most participants tended to

\footnote{The maximum value of \(i + j\) is 6 because the sum cannot exceed the number of apps shown to participants, which is 6.}
Figure 4.3: 3-D histogram of participants’ collective behaviours in terms of their choices on `su` and `ob` for `plain` apps

Figure 4.4: 3-D histogram of participants’ collective behaviours in terms of their choices on `on` and `ob` for `tool` apps
4.4. RESULTS

Figure 4.5: 3-D histogram of participants’ collective behaviours in terms of their choices on \textit{su} and \textit{ob} for social apps

follow one of four typical behavioural patterns (i.e., four peaks in the 3-D histogram – see Section 4.4.2).

For participants who reported full trust in objective ratings (79 participants), less than 6% of them (3 in Tools apps session and 6 in Social apps session out of 79) changed their reported trust for at least one app after the app category was revealed to them in the following session. On the other hand, 58 participants reported full trust in subjective ratings, less than 10% (8 in Tools session and 3 in Social session out of 58) of them shifted their answers for one app at minimum. This suggests that a great majority of subjective and objective ratings believers have a consistent reported trust in all sessions. One more thing to note is that around 7% (75/1077) of the participants reported trust in neither subjective nor objective ratings in both conditions, suggesting some human users may have an intrinsic disbelief on ratings given by others (machines and other people).

Although the pattern in the three 3-D histograms is clearly visible, a Stuart-Maxwell \(\chi^2\) test and a multinomial regression were conducted to see if there are any statistically significant differences. The Stuart-Maxwell \(\chi^2\) tests showed that there is an observed difference between plain and other app category sessions as seen in Table 4.3. The table shows that participants tended to trust subjective and objective ratings more when the app category was displayed. It deserves noting that participants trusted subjective rating more than objective in tool session. On the contrary, the multinomial regression results in Table 4.4 show a low McFadden’s
Table 4.3: Results of the Stuart-Maxwell $\chi^2$ tests for analysing participants’ self-reported trust in subjective and objective ratings

<table>
<thead>
<tr>
<th>Distributions Compared</th>
<th>$\chi^2$</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>su (Plain) vs. su (Tool)</td>
<td>23.163</td>
<td>0.0007</td>
</tr>
<tr>
<td>su (Plain) vs. su (Social)</td>
<td>15.205</td>
<td>0.019</td>
</tr>
<tr>
<td>su (Tool) vs. su (Social)</td>
<td>8.5759</td>
<td>0.199</td>
</tr>
<tr>
<td>ob (Plain) vs. ob (Tool)</td>
<td>14.9</td>
<td>0.021</td>
</tr>
<tr>
<td>ob (Plain) vs. ob (Social)</td>
<td>13.443</td>
<td>0.036</td>
</tr>
<tr>
<td>ob (Tool) vs. ob (Social)</td>
<td>5.2312</td>
<td>0.514</td>
</tr>
<tr>
<td>su (Plain) vs. ob (Plain)</td>
<td>3.745</td>
<td>0.711</td>
</tr>
<tr>
<td>su (Tool) vs. ob (Tool)</td>
<td>15.78</td>
<td>0.015</td>
</tr>
<tr>
<td>su (Social) vs. ob (Social)</td>
<td>9.5279</td>
<td>0.146</td>
</tr>
</tbody>
</table>

For $p$-value, "< $\varepsilon$" means that the exact $p$-value could not be obtained but it drops below the precision limit (which is $2.22 \times 10^{-16}$ for R, the language we used for statistical tests). The same notation will be used for other tables below.

Table 4.4: Results of the multinomial logistic regressions conducted on the app category condition as the predictor of participants’ self-reported trust

<table>
<thead>
<tr>
<th>Predictor</th>
<th>b</th>
<th>SE</th>
<th>$p$-value</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool su</td>
<td>-0.098</td>
<td>0.042</td>
<td>0.0216</td>
<td>0.907</td>
</tr>
<tr>
<td>Tool ne</td>
<td>-0.015</td>
<td>0.058</td>
<td>0.7974</td>
<td>0.985</td>
</tr>
<tr>
<td>Tool ud</td>
<td>-0.124</td>
<td>0.0498</td>
<td>0.0129</td>
<td>0.884</td>
</tr>
<tr>
<td>Social su</td>
<td>-0.047</td>
<td>0.043</td>
<td>0.2738</td>
<td>0.954</td>
</tr>
<tr>
<td>Social ne</td>
<td>0.0302</td>
<td>0.058</td>
<td>0.6059</td>
<td>1.031</td>
</tr>
<tr>
<td>Social ud</td>
<td>-0.034</td>
<td>0.049</td>
<td>0.491</td>
<td>0.966</td>
</tr>
</tbody>
</table>

$\chi^2 = 9.8746 \ (p < \varepsilon)$, McFadden $R^2$: 0.0002. The baseline of the independent variable (app category condition) is “Plain”. The pivot outcome is “ob” and OR means the odds ratio (the same hereinafter).

$R^2$ value, suggesting that the condition as a binary variable does not have a good predictive power. Regression result shows there is no significant difference at most as majority of $p$-values are higher than 0.05. That suggests the differences were better tested by other statistical tests (i.e., Stuart-Maxwell $\chi^2$ test).  

**Behavioural Patterns**

As it has been mentioned earlier in 4.4.2, four peaks can be observed in the 3-D histogram of collective behaviour of all participants, each referring to a distinct

---

4In the table, each row represents a prediction model where the predictor variable $x$ is the display condition (1 = display, 0 = hidden) and the predicted variable is $\ln \left( \frac{p(y)}{p(ob)} \right)$, and $y \in \{ su, ne, ud \}$. The predictor is a linear equation: $\ln \left( \frac{p(y)}{p(ob)} \right) = \beta_0 + \beta_1 \times x$. 

---
4.4. RESULTS

Subjective

- **P1** (N=246 (22%), centre = {0.7, 0.6})
- **P2** (N=391 (36%), centre = {1.8, 2.8})
- **P3** (N=281 (26%), centre = {4.3, 0.7})
- **P4** (N=159 (15%), centre = {0.2, 5.3})

Figure 4.6: 2-D histograms of participant’s collective behaviours in terms of their choices on subjective and objective behaviour. This appears to suggest that each participant may have some intrinsic behavioural style that can influence her/his self-reported trust in subjective and objective ratings. Therefore, by knowing which behavioural style a person has, the trust in subjective and objective app ratings can be predicted which will allow to test H2. Therefore, a k-means algorithm has been performed to cluster all participants of the user study into four behavioural clusters: P1 (disbelievers, 246 participants, centre={0.7, 0.6}), P2 (balanced believers, 391 participants, centre={1.8, 2.8}), P3 (subjective rating believers, 281 participants, centre={4.3, 0.7}) and P4 (objective rating believers, 159 participants, centre={0.2, 5.3}). Figure 4.6 depicts the four generated behavioural groups from k-means clustering.

Another multinomial logistic regression was applied by using the behavioural cluster of each participant which is obtained from running the k-means clustering. This regression tests whether the behavioural cluster label is a good predictor of participants’ perceived trust. The results are given in Table 4.5, which indicate that the effect is statistically significant with significant odd ratios. The result shows that subjective rating believers (P3) are more likely to select subjective ratings over objective ratings comparing to the balanced believers (P2). The odds ratio of (P3) shows that subjective rating believers are predicted to select subjective ratings over objective ratings more than those who belong to the other behavioural styles. Unsurprisingly, the disbelievers (P1) are more likely to trust neither ratings or select undecided more than any other group according to high odds ratios.
Table 4.5: Results of the multinomial logistic regression conducted on the behavioural pattern as the predictor of participants' self-reported trust

<table>
<thead>
<tr>
<th>Predictor</th>
<th>b</th>
<th>SE</th>
<th>p-value</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 su</td>
<td>0.325</td>
<td>0.060</td>
<td>$7 \times 10^{-8}$</td>
<td>1.383</td>
</tr>
<tr>
<td>P1 ne</td>
<td>1.809</td>
<td>0.063</td>
<td>$&lt; \varepsilon$</td>
<td>6.103</td>
</tr>
<tr>
<td>P1 ud</td>
<td>2.039</td>
<td>0.057</td>
<td>$&lt; \varepsilon$</td>
<td>7.686</td>
</tr>
<tr>
<td>P3 su</td>
<td>1.281</td>
<td>0.045</td>
<td>$&lt; \varepsilon$</td>
<td>3.6</td>
</tr>
<tr>
<td>P3 ne</td>
<td>0.301</td>
<td>0.071</td>
<td>$2.4 \times 10^{-5}$</td>
<td>1.351</td>
</tr>
<tr>
<td>P3 ud</td>
<td>0.271</td>
<td>0.064</td>
<td>$2.2 \times 10^{-5}$</td>
<td>1.312</td>
</tr>
<tr>
<td>P4 su</td>
<td>-1.996</td>
<td>0.073</td>
<td>$&lt; \varepsilon$</td>
<td>0.136</td>
</tr>
<tr>
<td>P4 ne</td>
<td>-1.939</td>
<td>0.115</td>
<td>$&lt; \varepsilon$</td>
<td>0.144</td>
</tr>
<tr>
<td>P4 ud</td>
<td>-1.395</td>
<td>0.08</td>
<td>$&lt; \varepsilon$</td>
<td>0.248</td>
</tr>
</tbody>
</table>

For p-value, "$< \varepsilon$" means that the exact p-value could not be obtained but it drops below the precision limit (which is $2.22 \times 10^{-16}$ for R, the language we used for statistical tests). The same notation will be used for other tables below.

The above analysis may be seen as circular reasoning as the personality labels are obtained from the data and then used to predict the data. To further validate whether the personality labels obtained from running the $k$-means clustering are reliable, another analysis was applied where the data was split into two non-overlapping subsets. Each subset contains users’ responses on a different subset of three apps. Then, $k$-means clustering algorithm was run on each data subset to derive the personality label for each participant and then used the label as an independent variable to predict the reported trust in the other subset. Next, a multinomial regression on each data subset was conducted, as can be seen in Table 4.6. The results showed that the odds ratios observed in the new analysis are mostly aligned with the findings in the first analysis, suggesting that most users behave consistently for different apps in how they reported their trust in subjective and objective ratings.

Another interesting element to study is the behavioural changes of participants with different behavioural patterns when the app category condition changed from plain to tool or social. Figure 4.7 shows a comparison between the distribution of users’ responses in terms of their choices on su and ob for the four behavioural patterns. There are four green sub-figures, each referring to a particular behavioural group style (P1, P2, P3, or P4) and highlighting the distribution of users’ responses when the app category was hidden. The number of participants in a particular behavioural group style is shown at the top of each green sub-figure.

The four yellow sub-figures highlight how users with a particular behavioural style changed their behaviour when the app category was shown as Tool. At the top of each sub-figure, the number of participants who did not change their behaviour is highlighted in dark grey while the sum of participants who completely shifted to
Table 4.6: Results of a multinomial logistic regression conducted on two app data subsets as the predictor of participants’ self-reported trust

(a) Model 1 based on Dataset 1

<table>
<thead>
<tr>
<th>Predictor</th>
<th>b</th>
<th>p-value</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1: ne</td>
<td>0.289</td>
<td>0.0511</td>
<td>1.335</td>
</tr>
<tr>
<td>P1: sb</td>
<td>0.623</td>
<td>$2.19 \times 10^{-13}$</td>
<td>1.865</td>
</tr>
<tr>
<td>P1: ud</td>
<td>-0.45</td>
<td>0.0018</td>
<td>0.636</td>
</tr>
<tr>
<td>P3: ne</td>
<td>0.725</td>
<td>$6.72 \times 10^{-11}$</td>
<td>2.065</td>
</tr>
<tr>
<td>P3: sb</td>
<td>1.121</td>
<td>&lt; ε</td>
<td>3.068</td>
</tr>
<tr>
<td>P3: ud</td>
<td>0.302</td>
<td>0.0012</td>
<td>1.353</td>
</tr>
<tr>
<td>P4: ne</td>
<td>1.839</td>
<td>&lt; ε</td>
<td>6.292</td>
</tr>
<tr>
<td>P4: sb</td>
<td>0.66</td>
<td>&lt; ε</td>
<td>1.936</td>
</tr>
<tr>
<td>P4: ud</td>
<td>1.798</td>
<td>&lt; ε</td>
<td>6.038</td>
</tr>
</tbody>
</table>

χ² = 1629.5 (p < ε), McFadden $R^2$: 0.063. The dataset includes users responses on App1, App3, and App5.

(b) Model 2 based on Dataset 2

<table>
<thead>
<tr>
<th>Predictor</th>
<th>b</th>
<th>p-value</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1: ne</td>
<td>0.114</td>
<td>0.325</td>
<td>1.121</td>
</tr>
<tr>
<td>P1: sb</td>
<td>-0.303</td>
<td>$2.72 \times 10^{-5}$</td>
<td>0.738</td>
</tr>
<tr>
<td>P1: ud</td>
<td>-0.198</td>
<td>0.0486</td>
<td>0.819</td>
</tr>
<tr>
<td>P3: ne</td>
<td>1.519</td>
<td>&lt; ε</td>
<td>4.571</td>
</tr>
<tr>
<td>P3: sb</td>
<td>-0.37</td>
<td>$5.03 \times 10^{-6}$</td>
<td>0.687</td>
</tr>
<tr>
<td>P3: ud</td>
<td>1.505</td>
<td>&lt; ε</td>
<td>4.505</td>
</tr>
<tr>
<td>P4: ne</td>
<td>-0.601</td>
<td>$4.54 \times 10^{-8}$</td>
<td>0.548</td>
</tr>
<tr>
<td>P4: sb</td>
<td>-1.003</td>
<td>&lt; ε</td>
<td>0.367</td>
</tr>
<tr>
<td>P4: ud</td>
<td>-0.69</td>
<td>$2.6 \times 10^{-14}$</td>
<td>0.502</td>
</tr>
</tbody>
</table>

χ² = 1626.9 (p < ε), McFadden $R^2$: 0.0628. The dataset includes users responses on App2, App4, and App6.

Another behavioural style is highlighted in dark red. The number of participants in each other behavioural style is highlighted in light red. The four orange sub-figures follow the same format as the yellow ones, but they represent how users with a particular behavioural style changed their behaviour when the app category was shown as Social.

As a whole, 39.5% of participants (425/1077) changed their reported trust when the app category was Tool, and 37.8% of them (407/1077) when it was Social apps. Less than half (133/281, 47%) of subjective rating believers (P3) changed their reported trust for at least one app (see Fig. 4.7 c). Most of them (30%) had an extreme shift towards balanced believers while a few of them (6%) shifted to trust objective ratings. However, objective ratings believers (P4) seemed to have a stronger view as
they shifted only slightly towards other groups (27% total participants who change their views) (see Fig. 4.7 d). For disbelievers (P1), the behavioural change had a wider distribution, whereby 40% (on average of both categories) of participants migrated to balanced believers more (see Fig. 4.7 a). Finally, P2 had less behavioural distribution (36% on average) as P1 but with high level of trust in subjective ratings (see Fig. 4.7 b).

![2-D distribution of participants' responses according to their behavioural pattern in each app category condition](image)

Figure 4.7: 2-D distribution of participants’ responses according to their behavioural pattern in each app category condition
4.4. RESULTS

Table 4.7: Results of the multinomial logistic regression conducted on the age factor as the predictor of participants’ self-reported trust

<table>
<thead>
<tr>
<th>Predictor</th>
<th>b</th>
<th>SE</th>
<th>p-value</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>26-35 su</td>
<td>-0.147</td>
<td>0.419</td>
<td>45.4 × 10^{-4}</td>
<td>0.863</td>
</tr>
<tr>
<td>26-35 ne</td>
<td>-0.176</td>
<td>0.055</td>
<td>14.5 × 10^{-3}</td>
<td>0.838</td>
</tr>
<tr>
<td>26-35 ud</td>
<td>-0.169</td>
<td>0.049</td>
<td>64 × 10^{-4}</td>
<td>0.845</td>
</tr>
<tr>
<td>≥ 36 su</td>
<td>-0.324</td>
<td>0.045</td>
<td>4.6 × 10^{-13}</td>
<td>0.723</td>
</tr>
<tr>
<td>≥ 36 ne</td>
<td>-0.631</td>
<td>0.063</td>
<td>&lt; ε</td>
<td>0.532</td>
</tr>
<tr>
<td>≥ 36 ud</td>
<td>-0.185</td>
<td>0.052</td>
<td>33 × 10^{-4}</td>
<td>0.831</td>
</tr>
</tbody>
</table>

\( \chi^2 = 130.84 \) (\( p < \varepsilon \)), McFadden \( R^2 \): 0.00258. The baseline of the independent variable is “≤ 25” balanced believers. The reference of app rating is “ob”.

Demographic Factors

More multinomial logistic regressions have been conducted to test the possible impact of demographic factors including gender, age, and skill level on participants’ self-reported trust. The results showed that the effect is not significant when testing gender and skill factors (\( \chi^2 = 165.41 \), \( p < \varepsilon \), McFadden \( R^2 \) = 0.0033, odds ratios are not far from 1). Nevertheless, that was not the case for the age factor. As in Table 4.7, the results prove the overall effect of age is statistically significant which suggests that the age can be a good predictor of the self-reported trust.

To further investigate how age influence users’ self-reported trust, the participants’ collective behaviours have been displayed in 3-D histograms based on the age group that they belong to. In Fig 4.8, the 3-D histogram shows participants responses who are 25 years old or younger. Despite the app category condition, the majority of this age group are in the balanced personality group. Yet, it is clear that not many of them fully trusted the objective rating. Figure 4.9 displays the same as the former figure but for the age group between 26 to 35 years. A majority of participants in this group are in the balanced area as well. However, there is a slight increase toward objective ratings, suggesting that more people in this age group trust objective rating than those who are younger. The very interesting finding can be seen in Fig. 4.10. For the age group older than 35 years, the majority of participant here fully trusted the objective rating. Nevertheless, the four personality clusters were visually obvious in all groups, which further confirms what is mentioned earlier in 4.4.2.

Rating Dependencies

As the presented apps in this experiment were not real, there is no point of comparing the users’ responses between individual apps. However, it might be interesting to look into app pairs to see if the rating combinations have an effect on the user responses. If that is the case, the participant self-reported trust can be predicted based on the provided details about ratings.
Figure 4.8: 3-D histograms of collective behaviors for participants less than 26 years.

- Social Apps
- Plain Apps
- Tool Apps
Figure 4.9: 3-D histograms of collective behaviours for participant's between 26 to 35 years
Figure 4.10: 3-D histograms of collective behaviours for participants over 35 years old.
A Stuart-Maxwell $\chi^2$ test was applied to see if the distribution of participants’ responses significantly shifted when the app changed between different sessions. In Table 4.8, the results do not show significant differences for each single app when the category condition was changed from plain to tool/Social or from Tool to Social.

Table 4.8: Results of the Stuart-Maxwell $\chi^2$ tests on participants’ perception of subjective and objective app ratings for different app sessions

<table>
<thead>
<tr>
<th>Predictor</th>
<th>$\chi^2$</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>App1(P) vs App1(T)</td>
<td>8.999</td>
<td>0.0293</td>
</tr>
<tr>
<td>App2(P) vs App2(T)</td>
<td>5.375</td>
<td>0.1463</td>
</tr>
<tr>
<td>App3(P) vs App3(T)</td>
<td>1.707</td>
<td>0.6354</td>
</tr>
<tr>
<td>App4(P) vs App4(T)</td>
<td>8.812</td>
<td>0.03189</td>
</tr>
<tr>
<td>App5(P) vs App5(T)</td>
<td>2.327</td>
<td>0.5073</td>
</tr>
<tr>
<td>App6(P) vs App6(T)</td>
<td>7.557</td>
<td>0.0561</td>
</tr>
<tr>
<td>App1(P) vs App1(S)</td>
<td>3.087</td>
<td>0.3784</td>
</tr>
<tr>
<td>App2(P) vs App2(S)</td>
<td>1.946</td>
<td>0.5838</td>
</tr>
<tr>
<td>App3(P) vs App3(S)</td>
<td>2.752</td>
<td>0.4315</td>
</tr>
<tr>
<td>App4(P) vs App4(S)</td>
<td>3.801</td>
<td>0.2837</td>
</tr>
<tr>
<td>App5(P) vs App5(S)</td>
<td>5.324</td>
<td>0.1495</td>
</tr>
<tr>
<td>App6(P) vs App6(S)</td>
<td>14.09</td>
<td>0.00279</td>
</tr>
<tr>
<td>App1(T) vs App1(S)</td>
<td>2.810</td>
<td>0.4218</td>
</tr>
<tr>
<td>App2(T) vs App2(S)</td>
<td>1.781</td>
<td>0.6191</td>
</tr>
<tr>
<td>App3(T) vs App3(S)</td>
<td>3.919</td>
<td>0.2704</td>
</tr>
<tr>
<td>App4(T) vs App4(S)</td>
<td>1.255</td>
<td>0.7398</td>
</tr>
<tr>
<td>App5(T) vs App5(S)</td>
<td>3.319</td>
<td>0.3449</td>
</tr>
<tr>
<td>App6(T) vs App6(S)</td>
<td>2.064</td>
<td>0.5593</td>
</tr>
</tbody>
</table>

*a P, T and S refers to plain, tool and social app condition, consequently.

Table 4.9 shows a comparison between different app pairs when apps were displayed as Tool. A number of Stuart-Maxwell $\chi^2$ tests were used for testing homogeneity for the four rating options (su, ob, ne and ud). The results showed significant differences between different app pairs, suggesting that users’ trust and decision-making were dependent on the displayed ratings. Table 4.10 is the same as the former table, but for app pairs when they were displayed as Social. It also indicates significant differences between each pair.

### 4.4.3 Users’ Reported Reasons

Users’ reasons behind their reported trust were collected. Figure 4.11 compares the percentage of reasons given for each app in different contexts. Each reason labelled with a different number (refer to Table 4.2) and colour. In total, 18% of participants reported that objective ratings are more trusted because the objective rating was the lower rating which made them feel it is safer. 25% of participants selected objective or subjective ratings as they matched their expectations while 7%
Table 4.9: Results of the Stuart-Maxwell $\chi^2$ tests on participants’ perception of subjective and objective app ratings for different Tool app pairs

<table>
<thead>
<tr>
<th>Predictor</th>
<th>$\chi^2$ (p-value)</th>
<th>Predictor</th>
<th>$\chi^2$ (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV 1 vs AV 2</td>
<td>3.1881 (0.3635)</td>
<td>AV 1 vs AV 3</td>
<td>30.473 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>AV 1 vs AV 4</td>
<td>20.567 (1.3 x 10^{-4})</td>
<td>AV 1 vs AV 5</td>
<td>140.08 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>AV 1 vs AV 6</td>
<td>213.73 (&lt; $\varepsilon$)</td>
<td>AV 2 vs AV 3</td>
<td>36.327 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>AV 2 vs AV 4</td>
<td>22.415 (5.3 x 10^{-5})</td>
<td>AV 2 vs AV 5</td>
<td>153.13 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>AV 2 vs AV 6</td>
<td>219.75 (&lt; $\varepsilon$)</td>
<td>AV 3 vs AV 4</td>
<td>12.033 (7 x 10^{-3})</td>
</tr>
<tr>
<td>AV 3 vs AV 5</td>
<td>62.464 (&lt; $\varepsilon$)</td>
<td>AV 3 vs AV 6</td>
<td>132.56 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>AV 4 vs AV 5</td>
<td>96.386 (&lt; $\varepsilon$)</td>
<td>AV 4 vs AV 6</td>
<td>154.88 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>AV 5 vs AV 6</td>
<td>25.523 (1.2 x 10^{-5})</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AV refers to ‘Anti-Virus’.

Table 4.10: Results of the Stuart-Maxwell $\chi^2$ tests on participants’ perception of subjective and objective app ratings for different Social app pairs

<table>
<thead>
<tr>
<th>Predictor</th>
<th>$\chi^2$ (p-value)</th>
<th>Predictor</th>
<th>$\chi^2$ (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG 1 vs MSG 2</td>
<td>0.7904 (0.852)</td>
<td>MSG 1 vs MSG 3</td>
<td>27.822 (4 x 10^{-6})</td>
</tr>
<tr>
<td>MSG 1 vs MSG 4</td>
<td>16.94 (7.3 x 10^{-4})</td>
<td>MSG 1 vs MSG 5</td>
<td>155.96 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>MSG 1 vs MSG 6</td>
<td>216.91 (&lt; $\varepsilon$)</td>
<td>MSG 2 vs MSG 3</td>
<td>25.58 (1.2 x 10^{-5})</td>
</tr>
<tr>
<td>MSG 2 vs MSG 4</td>
<td>15.67 (1.3 x 10^{-3})</td>
<td>MSG 2 vs MSG 5</td>
<td>167.15 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>MSG 2 vs MSG 6</td>
<td>236.26 (&lt; $\varepsilon$)</td>
<td>MSG 3 vs MSG 4</td>
<td>8.5674 (0.03563)</td>
</tr>
<tr>
<td>MSG 3 vs MSG 5</td>
<td>85.049 (&lt; $\varepsilon$)</td>
<td>MSG 3 vs MSG 6</td>
<td>157.5 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>MSG 4 vs MSG 5</td>
<td>113.9 (&lt; $\varepsilon$)</td>
<td>MSG 4 vs MSG 6</td>
<td>177.55 (&lt; $\varepsilon$)</td>
</tr>
<tr>
<td>MSG 5 vs MSG 6</td>
<td>19.55 (2.1 x 10^{-4})</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MSG refers to ‘Messenger’.

4.5 Discussion

In this study, users’ perceived trust of app ratings given by two rating sources (automated computer programs and human expert opinions) were compared. Hence, it targets providing a novel contribution to the related literature. Nevertheless, the impact of users’ subjectively created judgement on their behaviour show some overlapping results. This section expands on the interpretation of the results focusing on the hypotheses that are given at the beginning of this chapter.

Before moving on to discuss more, a point to observe is that some participants selected “neither” as none of the ratings matched their expectations. 18% of those who selected subjective ratings were influenced by their desire to be on the safe side, mainly because subjective ratings were lower than objective ratings.
4.5. DISCUSSION

Figure 4.11: Percentages of users’ reasons behind their trust on app ratings.
the observed changes are a reflection of realistic behaviour in real world. We acknowledge the natural limitations of using crowdsourcing workers for conducting user studies especially on quality of data collected, but the nature and some design elements of this study (simple tasks that crowdsourcing workers did not have much motivation to cheat) have given some level of confidence on the results that have been reported in this work. In future, the plan is to conduct an even larger scale study with more apps, more crowdsourcing workers and also a medium-scale lab-based study to further validate the results of this experiment.

As mentioned earlier in Section 4.3.2, another work on users’ perceived trust of password strength ratings has been conducted in parallel with this reported work. The intention is to investigate user’s self-reported trust of subjective and objective ratings in variant contexts. The design of app privacy and password strength sessions in this experiment has been aligned to allow comparing them. The work of the other part which is users perceived trust of password strength ratings is reported in [113].

4.5.1 Contextual effects

First of all, the app category condition has proven to have an influence on users’ self-reported trust. The finding showed a significant shift of participants’ collective behaviour after the app category condition changed, which confirms the hypothesis H1. Likewise in the password strength session, the shift of collective behaviour of users self-reported trust was more obvious when the password condition is changed from hidden to displayed passwords. As a consequent, that would suggest that such a finding can be potentially generalised to wider contexts.

Additionally, participants tended to trust objective and subjective ratings more when the app category was displayed. It also shows that users’ own subjective judgements on app privacy played an active role in their trust in both objective and subjective app ratings. This was more obvious in the password strength case, where passwords were displayed as participants were supplied with more information.

It has been also observed that some participants were risk-averse as they showed willingness to select the rating that matched their expectations while some others had a higher tendency to trust a trustee even when there is not enough information. These behaviours can be attributed to the strong impact of people’ behavioural patterns on their trust perception.

4.5.2 Behavioural patterns and their effects

This experiment revealed the existence of different behavioural patterns that have a significant impact on users’ self-reported trust, providing support for the hypothesis H2. Some participants appeared conservative and cautious when it came to trusting a particular type of app ratings. This is reflected in some participants’ tendency
towards selecting the lower rating to be on the safer side, suggesting that the app rating can also have an influence on perceived trust.

Some other behavioural patterns can be associated with trust bias. Participants who had an extreme trust in either subjective or objective ratings can be considered risk-seeker. In contrast, some participants seemed to be risk-averse as they avoided trusting any solutions in most cases, regardless of the specific situation.

The effect of the behavioural patterns was very obvious in all app categories condition. This suggests that user’s perceived trust can be predicted if his/her trust behavioural pattern is known. That means that it could help if contextualised and personalised app privacy ratings are provided based on the user’s behavioural pattern.

In the password strength part, the effect of the behavioural patterns becomes less obvious with displayed passwords compared with hidden password. This could be attributed to the fact that many participants based their self-reported trust on their own subjective judgements on the password strength, while their behavioural styles just become more or less a default setting.

### 4.5.3 Individual app rating impact on user’s trust

Since displayed apps in this work were not realistic, there was no point of comparing individual app impact. Yet, the associated ratings with apps is an interesting element to look at. The intention is to see if users’ behaviour is affected by the displayed ratings. The participants’ collective behaviour showed that the app category had an obvious influence on users’ self-reported trust. However, when individual apps from one category were compared to those from another category (e.g., App1 vs. Messenger1, Antivirus3 vs. App3), there were no observed significant differences. Yet, the case is not the same when the comparison was done within the six apps in each category, Stuart-Maxwell $x^2$ tests returned very low values. This indicates that user’s decision making was dependent on the displayed ratings.

Whereas the main objective of this study is to know what affects users more the subjective or objective rating, all other app details were excluded to avoid complicating the study. As mentioned earlier, the aim is to eliminate any potential impact of other factors such as app name or familiarity, icon attractiveness, etc.

### 4.5.4 Demographic factors

Unexpectedly, we did not observe any significant influence of gender and skill-level on users’ self-reported trust in subjective and objective ratings. This could be linked to the effect of an unbalanced population sampling for these demographic factors, which is one of the limitations of this study. Yet, this is not the case for the age population of the experimented sample. Age has proven to be a good predictor of users trust, which indicates significant influence on users’ self-reported trust of ratings. Consequently, the hypothesis H4 can be partially accepted.
Age element has revealed an interesting finding in terms of participants’ collective behaviours that requires further investigation. Clearly, the older people age group showed more constant responses in all sessions comparing to other age groups. This can be related to experience and age level that allowed them to make the same decisions despite the app category. Yet, younger people (less than 36) showed more randomness in terms of the collective behaviour in the plain session. Once the app category was displayed, their collective behaviour became closer to how the older age group people behaved.

4.6 Limitations

The choice of apps and their subjective and objective ratings impose some limitations. Users’ trust may be influenced by many other factors [104, 105] that are not easy to control and analyse in a single user study. These factors can include apps’ other features (i.e., icon feel and look, brand, developer, familiarity), context of use, etc.

Although it has been attempted to select representative app categories for the experiment, a larger set of apps are needed in order to study the effect of many other categories on users’ self-reported trust of privacy ratings.

It has been stated earlier that using real apps will complicate the design of the study. Having too many factors with the displayed app will not help knowing what influenced users’ responses. Thus, the choice was made to use dummy apps. Studying other factors is left for future studies, although it might be challenging to be indicative in allocating both subjective and objective ratings for real apps (instead of dummy apps) since there is no perfect objective estimator, and no available “ground-truth” subjective ratings obtained from security experts.

As mentioned before, the behavioural analysis of the app category condition implies that it is more likely that many (if not most) participants were well engaged. One question that needs to be addressed, however, is whether participants who did not change their answers for all app sessions were actually engaged. Although there is a possibility of cheating, there is no clue about the actual number of cheating behaviours or misunderstandings of the questions. This intrinsic problem could be attributed to the use of a crowdsourcing platform, and future research is needed to see if this issue can be studied with more evidence on the level of engagement of each individual participant.

4.7 Conclusion & Future Work

This chapter reports a novel study (to the best of the author’s knowledge) comparing users’ perceived trust on mobile apps’ privacy ratings given by automated algorithms and by human experts. The main findings include: 1) users’ self-reported trust in subjective and objective ratings of app privacy is heavily influenced by users’ own
subjective judgements; 2) users behave differently for different app categories; 3) there are different behavioural patterns that can strongly influence users’ decisions; 4) users have a (slightly) higher tendency to trust objective ratings when their own subjective judgements match the objective ratings; 5) age can play a significant role in users’ self-reported trust in subjective and objective ratings.

The user study produced some relatively more surprising results. Particularly, the lack of observed effects of gender and skill level on self-reported trust is unexpected. While a number of possible explanations can be speculated, the results imply that users’ perceived trust and their knowledge on apps are more complicated than it has been previously thought. The results may also be related to the limitations of the crowdsourcing method itself, where the demographic information provided by participants may contain much more noise than other more controlled experiments. The exact effect of demographic factors requires further investigation. Nonetheless, it has been observed that age has an obvious influence on the user’s self-reported trust.

Although the demonstrated work is about app privacy ratings only, a parallel study on subjective and objective ratings of password strength was also conducted. The results showed that participants’ collective behaviours differed from the app case, which led to the belief that the application context also matters (as expected).

As a whole, it is believed that more future research should be conducted to enrich the evidence on how users perceive mobile apps’ privacy ratings and how they choose what to trust. Particularly, considering the limitations of crowdsourcing based studies, our plan is to conduct more crowdsourcing based studies and also a traditional lab-based study to further validate the reported results. Such studies will help the design and deployment of mobile apps’ privacy ratings, and possibly the presented details of apps in app stores.

In terms of the role of the difference between subjective and objective ratings, some future work is required to study how the rating differences can influence the results. The ratings may also need some careful handling, e.g., it might be better to use real apps and real subjective and objective ratings to confirm findings reported in this chapter. Factors are built based on real data might affect the users’ self-reported trust in subjective and objective ratings. Therefore, it is planned to include experts opinion to represent subjective ratings as well as to use real apps with some real objective ratings. This will allow to study the effect of them on users’ trust which might lead to interesting findings of this research.
Chapter 5

PAltRoid: Improving Mobile User Engagement in Privacy-Related Decisions

5.1 Introduction

As mentioned earlier in Chapters 1 and 4, the more mobile apps are getting life-engaging and sophisticated, the more privacy concerns come on the scene. We have given many examples that show how mobile apps can invade user’s privacy. The user is entitled to choose her/his apps. Furthermore, the way how Android permission framework works, as described in 2.6.1, it leaves granting access permissions to sensitive resources to the user too.

The research has shown that mobile users often fail to understand or ignore communicated privacy details at installation time, despite the risk that users are willing to take in favour of acquiring what they look for. These two reasons contributed to the problem of privacy leakages on mobile devices. The research community has suggested many ways to communicate privacy indicators and nudges in a more understandable and concise approaches. All that takes place to assist users in making better privacy-related choices.

In this scenario, the user is very likely to end up with some privacy invasive apps on his/her device. Due to lack of awareness, users often get a lot of privacy invasive apps installed on their devices. Moreover, there is a lack of privacy awareness enhancing tools on mobile devices. Enhancing privacy awareness can help users make more informed decisions about installing and uninstalling apps. This could be achieved if solutions or alternatives are provided to the user. However, there are only few tools where authors recommended more privacy-friendly apps to users.
Consequently, we decided to develop a tool to fill that gap, and conducted user studies to verify the usefulness of the proposed tool.

The rest of the chapter is organised as follows. Section II discusses related work. The design of PAltRoid is described in Section III. Section IV presents a lab-based and a crowdsourcing-based study we conducted to verify the performance of PAltRoid. Section VI illustrates our analysis of the results of the user studies. Section VII discusses the implications of the findings and some limitations. The last two sections illustrate planned future work and concludes the chapter, respectively.

5.2 Related Work

This section is divided into three subsections, improving users privacy awareness, privacy scoring of Android apps and alternative apps identification. In the first subsection shows how researchers tried to improve privacy awareness of mobile device users. Next subsection presents some work focusing on privacy scoring of Android apps. Lastly, related work on generation of similar Android apps is illustrated.

5.2.1 Improving Privacy Awareness of Users of Mobile Devices

Researchers have proposed many approaches to improving the privacy awareness of users of mobile devices. These methods mainly focus on communicating privacy nudges and information to the user, so that they can make more informed decisions. “A nudge is a soft, paternalistic intervention, usually manifested as a subtle yet persuasive cue that makes people more likely to decide in one direction or the other”, according to Zhang and Xu [115].

Android has been developing its permission granting system to make users more privacy-aware. We discussed earlier how that model can assist the user in making more choices towards safer privacy. However, the research has proven that users still make bad calls. Therefore, researchers suggested many solutions. In this subsection, we review some of the selected recent ones. One of the methods is to equip users with privacy details prior to app installation. For instance, Tian et al. [116] proved that users’ decisions can be improved by giving users some details about Internet of Things (IoT) apps to avoid granting permissions to over-privileged apps. The idea is to close the gap between what IoT apps offer and how they function in run-time. SmartAuth (as Tian et al. call it) analyses collected data from an app’s description and source code and then communicates it to users. User studies conducted on the system provided evidence that the proposed design had improved users’ decisions in the lab. Gerber et al. [117] suggested an alternative design for Google’s permission-granting interface called COPING. Using an online study, COPING authors compared it against several published interfaces including Google’s legacy and recent permission interface. The results show that reducing interface complexity does not necessarily help users make privacy related decisions. Consequently, the design should help maximise both comprehensiveness of communicated details
and user understandability of them. Dini et al. proposed MAETROID [118] that analyses an app’s set of permissions and metadata collected from Google Play and generates a simple icon represents the app’s risk level (Trusted, Medium Risk or High Risk). Kleek et al. [119] investigated how users’ choices are effected when revealing how apps handle data. Their key hypothesis is that exposing users to richer details about previously hidden information flows out of the apps, such as what data has been collected, who is behind it and how the data is being used, will help users make privacy-aware choices. Kleek et al. shown through their lab study that privacy indicators do support people in making more confident and consistent choices towards privacy side. Rajivan and Camp [120] studied the influence of privacy priming and privacy cue framing on users’ choices of Android apps. The outcome of their user study indicates that some icon framing led to more privacy-aware decisions and priming for privacy increased users’ awareness.

Other researchers looked into protecting privacy to rise users’ privacy awareness. Hatamian et al. came up with FAIR [121] and proposed to generate a privacy risk score by applying fuzzy logic to data collected by apps from different sources. That score highlights the privacy invasiveness of installed apps, so users can decide if they should uninstall or replace more privacy-invasive apps. INSPIRED [122] detects suspicious security and privacy behaviours and tries to infer the underline intention behind such behaviours. It matches the user’s intention with apps’ intended behaviours and justifies granting the required permission. To understand the user’s intention, INSPIRED applies a machine learning based permission model that collects user behavioural data. SecuRank [123] analyses permission usage within app groups to detect over-privileged apps, and suggests a functionally similar apps that require less sensitive access.

Zhang et al. [124] proposed a different way to close the gap between app descriptions and permissions. Authors came up with DESCRIBEME tool, which generates security-centric app descriptions that help the user to avoid malware and privacy threatening apps. Eventually, the user will be provided with security-centric details that aid him to make better calls. Kong et al. [125] approached the same problem using a different way. Authors bridged the gap between the security issues and users perception by automatically assessing the review-to-behaviour fidelity of mobile apps. To prove their idea, they introduced AUTOREB applies some state-of-the-art machine learning techniques to infer the relations between users reviews and 4 categories of security-related behaviours. Furthermore, AUTOREB employs a crowdsourcing approach to automatically aggregate the security issues from review-level to app-level. Experiments show that it is possible to infer the mobile app behaviours at user-review level with high accuracy.

5.2.2 Privacy Scoring of Android Apps

Many researchers are interested in methods that can help users to understand the potential impact of mobile apps on users’ privacy. The reason is that users do not get
enough information and help, so they are making decisions that harm their privacy, such as ignoring warnings about requested permissions in installation [93]. Kang et al. [94] proposed a privacy metre, which clarifies the potential risk of mobile apps based on their requested permissions. PrivMetrics [3] has a rating system utilising basic anomaly detection methods along some privacy level indicators. PrivMetrics groups apps based on their functionality and supposes that each group have a similar behavioural pattern. Apps deviating from their groups are treated as anomalous ones. An app’s final rating is the mean of each of the following-calculated probabilities [126]:

1. Number of requested permissions
2. Deviation of requested permissions from average behaviour
3. Number of integrated third party trackers
4. Number of executed dangerous API Calls by the trackers

Another work in this category is PrivacyGrade [95,127]. The aim of this tool is to rise awareness of the behaviours that many mobile apps have that may affect users’ privacy. It uses a privacy model that measures the gap between an app’s actual behaviour and users’ expectations of that app’s behaviour. The app’s behaviour is obtained from statically analysing sensitive data that the app uses and how the data is used. More precisely, the static analysis check if tested apps use sensitive data primarily due to third-party libraries included with the app. For instance, the analysis can help in inferring how sensitive resources are used on the phone and for what. If app A requires access to GPS location, the use of third-party libraries will tell whether this app uses it for map purposes or other advertising purposes. On the other hand, users’ expectations are collected via asking crowdsourcing workers about their expectations of a core set of 837 different apps. Then, PrivacyGrade shows the result of applying its model in a form of grade to make it easier to be understood by users. To be more specific, all apps are assigned scores based on PrivacyGrade model. Apps show no privacy issues are assigned an A+ grade. The rest of the apps are sorted by their rating and then split into quartiles. The first quartile of apps are graded as A, the second quartile are assigned B, the third quartile received C, and the last quartile assigned D as the lowest grade.

5.2.3 Alternative Apps Identification

One of the proposed methods to identifying apps similar to a target app is to compare the textual information associated with candidate apps and that of the target app. In the natural language processing community, similar techniques have been extensively investigated resulting in many approaches to measuring textual semantic similarity, which is intended to tell how related text A is to text B. The comparison can be extended depending on the granularity of the texts. The word-to-word level comparison has been widely studied, and the WordNet Similarity
library [128] was introduced for this purpose and used by many researchers. At the sentence-to-sentence and text-to-text levels, there are also many proposed methods such as Optimal Matching, Greedy Pairing, LSA (Latent Semantic Analysis), LDA (Latent Dirichlet Allocation), and others [129]. For instance, in LSA, the texts are transformed into vectorial representations and a semantic similarity score is computed from the cosine of these vectors [130].

Some researchers considered the problem of identifying alternatives of a given Android app. PrivMetrics 5.2.2 recommends functionally similar apps and better in terms of privacy through using LSA to compare similarity of descriptions of two given mobile apps [3]. AnDarwin detects similar Android apps based on their semantic information. It is based on application code analysis via comparing apps to a subset of their closest neighbours [131]. Chen et al. showed how to find cloning apps through matching centroids generated out of encoded control flow graphs of apps [132]. Gorla et al. [133] implemented LDA on app descriptions and clustered generated topics using the $k$-means algorithm to find apps with similar descriptions. Thung et al. used CLAN (a system for detecting similar software systems) on systems tags from SourceForge to detect similar applications [134]. Moreover, Thung et al. presented CLANdroid to help users find alternative mobile apps by comparing Android-specific semantic anchors such as API calls, intents, permissions and a few others [135].

Another interesting piece of work is the AlternativeTo website [136]. This web service is entirely built upon provided alternative apps from crowdsourcing users. It is a free service that help users to find alternatives of products from variant platforms such as web application, Desktop software, mobile apps, wearable apps, etc. All alternative apps are provided by human users of the website: users can add new apps, suggest new alternative apps for any existing app, and vote on any alternative app recommended by others via likes and dislikes.

5.3 Design

This section illustrates the objectives and hypotheses of the whole study. Then it explains the design of the presented application along with used and collected data and details of alternative apps identification. PAltRoid’s name was chosen to represents the related terms (P)rivacy, (Alt)ernatives and And(Roid).

5.3.1 Objectives

The objectives of this study are as follows:

- To evaluate the presented design user interface for Android apps, in terms of to what extent users will be engaged in privacy-related decisions.

- To collect participants view of the new designed interface and their feedback on how PAltRoid’s design can be further improved.
5.3.2 Hypotheses

Using the app, mobile users will be effectively engaged more in privacy-related decisions on their devices. Therefore, it will help the users to improve their overall privacy level.

5.3.3 PAltRoid Overall Design

The main purpose of designing PAltRoid is to improve mobile users engagement in privacy-related decisions. To achieve that goal, we apply the following ideas into the design of PAltRoid. First is adding privacy ratings and making them more visible to users will help. Moreover, giving users alternative apps with similar functionality but better privacy scores can influence users towards installing more privacy-friendly apps. That may motivate users uninstall apps that are not privacy friendly and not important, which can help reduce overall privacy risks without compromising utility.

Figure 5.1 shows the architecture of PAltRoid. It consists of three main components. Namely they are the mobile app, data server and alternative apps engine. The proposed interface design is represented in the mobile app to communicate privacy-related details. This app is connected to the server to be able to grab mobile apps’ privacy scores and information of alternative apps when needed. Finally, the alternative app engine generates similar apps in terms of functionality to be offered as replacement of apps with low privacy scores.
5.3. DESIGN

5.3.4 PAltRoid Components

This subsection contains a detailed description of the proposed system components.

The mobile app

has three activities, which are mainly used to communicate privacy information to the user. The first activity shows apps’ privacy ratings on a bar chart as in Fig. 5.2a. The length and colour of each bar represents the privacy score of an installed app. Very low privacy rated apps are listed on top of the chart and will be viewed first in order to emphasise on the risk that they pose. Thus, apps are ranked based on privacy-invasiveness from top to bottom. The implemented privacy scoring is a 5-grades scale ranging from A+ (most privacy sensitive) to D (least privacy sensitive). The 5-grade ratings scale was obtained from PrivacyGrade which is mentioned earlier in 5.2.2. There are five privacy ratings A+, A, B, C and D which are respectively colour-coded as dark green, green, yellow, orange and red. PrivacyGrade team made a website where the grades are colour coded this way [137]. In our PAltRoid design, the same colouring scheme as PrivacyGrade has been followed to provide consistency aside from being clear indicators of a low or high privacy level.

In cases of low privacy scoring apps, namely B, C and D, their privacy bars will be incremented with a grey stacked bar to highlight possible improvement of the privacy score if an alternative app is installed. The improvement can be made if the risky app is replaced by one of the recommended alternatives. When the grey stacked bar is clicked, the second activity will be launched. It will display recommended alternatives of the selected app as Fig. 5.2b shows. These alternatives are ranked on two levels, first from better privacy ratings to lower ones then based on how similar they are to the clicked app. Each alternative app is equipped with user ratings from Google Play to help the user in making the right decision. Furthermore, clicking on any recommended alternative app will take the user to the third activity, which will provide the user with app description as it appears on Google Play alongside with its link to the app store. By offering that, the user will be able to select between alternatives and once decided, PAltRoid will take the user to the app page on Google Play to proceed. Figure 5.2c shows an example of how the third activity appears on the mobile screen.

The data server

works as the data supplier for the mobile app as well as a data collector (on user’s behaviour). Collected data from users is described later at subsection 5.3.6. This server has a database holding two tables. One table contains relevant details including privacy scores of more than 1.4 million free third-party apps which are obtained from PrivacyGrade, and the other one represents alternative apps data. The server resides online to allow different instances of the client app to access it from anywhere.
over the Internet. MySQL is used to construct the database on the server, and the server is equipped with PHP scripts that allow mobile app clients to communicate with the database.

**Alternative apps engine**

The development of this component went through two stages. In the first stage, several *textual measuring techniques* on app descriptions were applied to determine similarities between any two given apps. SEMILAR [138], a semantic similarity toolkit, has a Java library that has been utilised in the generator to find similarities among compared peers. Yet, some pre-processing on textual data needs to take place to apply similarity methods. Therefore, textual data has been pre-processed in four steps, as follows:

1. tokenisation (i.e., removing punctuation from words),
2. converting words into their base forms,
3. part-of-speech tagging,
4. syntactic parsing.

A small test on available semantic similarity methods on the SEMILAR library was set to come out with the approach that suits what is needed. It was found that LSA is more sufficient that other tested methods in finding similarities in the prepared testing set. LSA was better in giving more reasonable similarity scores, as well as the incurred overhead is lower than other approaches. Since semantic measuring techniques come with the price of false positives, it has been decided to improve this components to provide better alternatives.
In the second stage, previous results were replaced with another data obtained from a crowdsourcing website AlternativeTo (introduced earlier at subsection 5.2.3). The power of crowdsourcing allows it to offer more precise and relevant result than computer generated, based on our subjective evaluation on a random small subset. The alternative apps engine has been incorporated with a web crawler to fetch relevant data from AlternativeTo website. Owners of this website have generously allowed us to do the web crawling.

5.3.5 Data Used

In PAltRoid, the proposed design interface requires two forms of data. The first set of data provides details about third-party Android apps and their privacy ratings. Since designing a new privacy scoring algorithm goes beyond the scope of our work, we were searching for a mobile app privacy scoring database that can meet our need. Hong et al. [95, 127] have generously supplied us with a snapshot of their dataset (dated from last quarter of 2016). This dataset covers more than 1.4 millions Android apps, including details such as app descriptions, privacy ratings, Google Play links, and many other useful information.

The second type of required data is alternative apps. This source of data proved useful in affecting users’ decision. As mentioned earlier in 5.3.4, in the initial design of PAltRoid, alternative apps were calculated using LSA. However, that approach produced many false positives. Approximately each app was provided with 20 alternative apps, out of them 15 to 17 on average were considered irrelevant apps by participants from a pilot study (which will be described later at Section 5.5). For that reason the option has been made to use data which is built on users’ crowdsourced preferences. After doing a comprehensive search for web services to provide that data, AlternativeTo [136] was found to be a suitable candidate as its data is primarily based on crowdsourcing. One drawback of data from AlternativeTo is that it does not cover as many apps as PrivacyGrade, but the number of apps covered are sufficient for our research and our inspection on a random subset of the covered apps showed a much higher accuracy than our LSA based approach.

5.3.6 Collected Data

Collected data are a list of users’ installed apps, participant ID as a unique identifier (not linked to the participant’s personally identifiable information), and apps privacy scores. Participants’ behaviour was monitored to see if using the app triggered them to make privacy-related decisions on mobile apps. Recorded information related to such behaviour includes the number of newly installed apps, the number of uninstalled apps, the frequency of accessing the PAltRoid app and the amount of time spent in each screen activity.
5.4 Experimental Studies

In this section, details of the conducted studies to evaluate PAItRoid can be found. The studies were carried out in a lab and crowdsourcing-based formats, whereas a quantitative method has been followed to test the hypotheses in 5.3.2. The main reason for conducting these studies is to evaluate our proposed PAItRoid.

5.4.1 Ethical approval

Both lab-based and crowdsourcing-based studies was reviewed and given a favourable ethical opinion by the University of Surrey’s University Ethics Committee (UEC). The UEC reference number for the lab-based study is ‘UEC/2017/074/FEPS’, while for the crowdsourcing-based study is ‘UEC 2017 074 FEPS Amendment 1’.

5.4.2 Study Groups

This study involved three groups to test the proposed design. The following list explains these groups and the reason behind choosing them.

- Experimental Group 1 (Group 1): In this group, PAItRoid full design has been experimented. This version has all design elements explained earlier at subsection 5.3.4.

- Experimental Group 2 (Group 2): Participants in this group experimented a basic design of PAItRoid. This version of the app has only the same main screen activity without offering any alternatives.

- Control Group (Group 3): the interface has nothing displayed to participants. In this group, the aim is to investigate participants’ base-line behaviour which will eventually tell their uninstall rate of apps.

5.4.3 Lab-Based Study

In this subsection, the procedure followed and recruitment details for the lab-based study can be be found.

Procedure

To test the hypotheses, a between-subjects lab-based experiment was conducted. Participants were allocated to one of the user groups. In this lab-based experiment, there were two user groups (the third group introduced later at the crowdsourced-based study), Group 1 and Group 2 (Group 2 was considered as a control group at this stage). Users in Group 1 have experimented the proposed design interface, while in Group 2 participants have only exposed to the first app activity which displays privacy scores of installed apps (no alternatives offered). In the beginning
of the study, participants were given a brief description of the study, including the study objectives and tasks. Moreover, participants were provided with a copy of Participant Information Sheet ‘PIS’ (refer to Appendix A) to inform them about the study. The PIS explains everything to the user from their role, purpose of the study, benefits and risks, who handle the data and many other details.

Each study participation consists of three sessions. First one is where participants are being briefed and helped to install the study app. The second session involves allowing some time for participants to experiment the app in real world. In the last one, participants are being debriefed and offered help to uninstall the experimented app.

In the first session, each participant was helped to get the PAAltRoid app installed on her/his device. The research coordinator (Saeed Alqahtani) encouraged participants to interact actively in the process of using the targeted app design. Then, participants in the second session were asked to use the app on a daily basis for one week, and at least once a day. To keep that going, the app was set up to trigger a reminder in the notification area in case a participant did not access the app for longer than 24 hours. Relevant anonymous data (described in subsection 5.3.6) were uploaded to the server that is connected to all ‘client’ apps.

In the third session, the research coordinator helped participants to uninstall PAAltRoid app from their mobile devices. Participants were then had a short semi-structured interview with the research coordinator, which was used to collect their subjective views, overall experiences on different aspects of the PAAltRoid app and their privacy awareness (refer to Appendix B for interview schedule). Moreover, they were also asked to take a quick survey to collect their demographic information and other feedback of the study. The questionnaires for Group 1 and 2 can be found at Appendix C.

Recruitment Details

Below are the details of the recruitment process of this study:

- **Test environment**: First and third sessions of the user study were conducted in a controlled lab environment using one computer at the University of Surrey and the participants mobile device.

- **Test equipment**: To conduct the planned user study, a desktop was prepared to help us install the PAAltRoid app to participants’ mobile devices. Each participant was asked to also use that computer to fill the electronic survey at the end, if the participant gives his/her consent.

- **Incentives**: Participants were compensated for their time spent for participating in the study. Each participant received a payment in cash based on the standard rate of £10 per hour. As there were two sessions to attend at the lab, users who fully participated were compensated with £20 in total.
• **Task duration:** Each session of the study did not exceed one hour. On average, each study session lasted for approximately half an hour. For the time between sessions, users were asked to keep the PAltRoid app with them for at least one week. We estimated that 7 days should be sufficient for collecting enough data about participants’ behaviours.

• **Inclusion criteria:** The participants were required to be over the age of 18. Also, users must own an Android mobile device and the Android version must be higher than API 16: Android 4.1 (Jelly Bean). That is due to the reason PAltRoid uses features work on API 16 and higher.

• **Sample size:** The aim is to collect on minimum 80 participants in each group. Meaning that, 160 in total (2 experimented designs * 80 = 160), yet the study was driven to collect as much as possible to strengthen the observed result of the analysis.

• **Recruitment:** We aimed at recruiting students, staff of the University of Surrey and people from the local community through internal advertisements such as posting recruiting flyers and advertising posters on campus and online social networks. Appendix D shows the advertising poster used for the lab-based study.

### 5.4.4 Crowdsourcing-based Study

At the planning stage of this experiment, a crowdsourcing-based study was reserved as a back-up plan in case we have any difficulties completing the lab-based study (e.g., recruiting enough participants). We also planned to use a crowdsourcing-based study to cross-validate the results of the lab-based study. The lab-based study proved difficult in recruiting the targeted number of participants. However, based on the experience from the work in Chapter 4, crowdsourcing approach can be considered faster in recruitment process. This subsection explains the crowdsourcing-based study that was carried out to test PAltRoid.

#### Crowdsourcing Platform

In this study, Amazon Mechanical Turk (AMT) was chosen as the crowdsourcing platform. AMT has some extra features makes it more preferred than other platforms. Most importantly is the advanced API and command line interface that help to design complicated tasks with minimal effort. AMT is quite popular among researchers and crowdsourcing users. Furthermore, it gives more control over recruited participants, such as paying satisfactory work, recruit many users in less time, etc. People who create tasks are called ‘requesters’, while participants are called ‘workers’. Each worker has a unique identifying worker ID. Moreover, a requested job or a designed task is referred as Human Intelligence Tasks (HIT) [139].
5.4. EXPERIMENTAL STUDIES

Procedure

For the crowdsourcing-based study, we followed the same overall procedure as the lab-based one. Nonetheless, minor aspects of the design were changed to suit the nature of the crowdsourcing studies, as described in details below.

To reach the targeted number of participants, the crowdsourcing-based experiment was conducted. This crowdsourcing study was split into two HITs. Each was designed to be completed within 30 minutes. At the beginning of the first HIT, a modified version of the PIS (refer to A) was displayed to participants. By taking the that HIT, workers consent to the study terms and conditions (described in the PIS). After a participant had agreed to proceed, he/she was instructed to follow a download link of the PAIrtRoid app preceded with how-to steps shown on the HIT. When the app was launched, it asked the participant to insert her/his worker ID on AMT which helped to identify who completed what. Also that ID was used to compensate and qualify workers for the second HIT. Workers who completed first HIT were motivated to comeback after one week time of experimenting the app.

![Figure 5.3: First time instruction fragment](image)

Seven days after the first HIT was closed, the second HIT was posted on AMT. Only workers of the first HIT who followed instructions were allowed to proceed. In the second HIT, the participants were instructed to uninstall the testing app. Then, their views and feedback were collected through an online survey embedded in the task. The estimation of this session is to last for about have an hour, yet most of the workers managed to complete it in less than ten minutes.

In total, the needed time for the whole experiment is approximately two hours. Each HIT requires 30-45 minutes and the daily usage is estimated to be between 3-4 minutes per day.
Study Survey Design

In the second HIT of this study, participants' subjective views were collected using an online survey. Since the crowdsourcing workers were recruited virtually, it was not possible to interview the participants due to their large number. Therefore, the design of the survey has been extended to collect more related feedback. The survey consists of four parts. The first part contains the task instructions. The second part gathers demographic details about the participants such as gender, age, nationality, language/s spoken and the level of education. Then some background questions were asked in the third part to collect information about the privacy awareness of the participants. The last part is an assessment of the tested PAltRoid app. Different types of questions have been used ranging from multiple choices, Likert-type scales, open-ended questions and free-formatted comments. The type of question was selected based on the nature of the information sought. The attached survey in Appendix E shows how this survey was designed.

Compensation and Quality Assessment

The total amount of money given to each participant is $5 where $1 for the first HIT and the rest rewarded once the second part was completed. The second HIT offered more money to encourage workers to comeback after the first HIT. After investigating the average given hourly wage on AMT, it was found that requesters mostly pay around $2 per hour. For this study, we decided to give $2.5 for each spent hour.

AMT allows requesters to have sometime before compensating workers in order to check the credibility of the submitted work. On the other hand, the collected data on the server helped in identifying outliers who did not follow the study instructions, and therefore excluded from the second HIT. Moreover, the compiled survey results also helped in determining whether a human or robots took the HITs. That can be told from the provided answers where this can be judged based on the percentage of sensible answers.

5.5 Results & Analysis

This section reports the finding from both the lab-based and crowdsourcing-based studies.

5.5.1 Participants

Based on a statistical power analysis, the power of our experiment design is 0.865, where given an expected medium effect size of 0.5 and $p < 0.05$. Knowing that the sample size of the presented crowdsourcing-based study is 76 at each group.
5.5. RESULTS & ANALYSIS

In the lab-based study, the minimum number of participants could not be achieved. There were 12 participants in the experimental group and 10 in the control group (Group 2). The lab-based study proved challenging and time-consuming, although variant means of advertising the study was used. Therefore, the decision was made to switch to the crowdsourcing-based study and the lab-based one was used as a pilot.

Through the crowdsourcing platform used, namely AMT, we managed to recruit 300 participants in total. 228 of them have succeeded to follow HITs’ instructions. The remaining 72 workers were excluded as some were considered outliers and some did not comeback to do the second part. These were allocated randomly to one of the three groups (76 in each group). Moreover, the recruitment was held at different times and via multiple batches, in order to ensure the demographic diversity of the recruited participants. Actually, not all workers were accepted as some didn’t follow the instructions. However, AMT allows us to call for another worker whenever a previous one’s work is rejected. The experimental group was completed first with 76 valid participants. As a consequent, it was aimed to get the same exact number in other groups to facilitate the statistical analysis.

The participants in this crowdsourcing-based study were from 19 different nationalities. In Group 1, there were 59 male and 17 female participants, while 52 male to 18 female participants were in Group 2. The male-female ratio was roughly 76% to 24%. The numbers of participants in different age groups are: 8 and 7 in age group “25 or below”, 50 and 43 in age group “26-35”, 18 and 20 in age group “36 or older”, in Groups 1 and 2, respectively. We did not invite Group 3 workers to the second HIT. Therefore, no demographic information are available for them.

5.5.2 App Ratings

The distribution of apps is shown based on their privacy scores. The focus here is on Groups 1 and 2 as they are designed to display privacy ratings to the participants. Group 3 does not have such details as it was used only to know the “base-line” behaviour of participants: the “natural” rate of app uninstallations.

Figure 5.4 shows the distributions of apps’ privacy scores in both groups. In Group 1, participants had 3,394 apps in total, about 52% (1,755 apps) of them have a privacy rating in our dataset that was displayed to the participants. Out of these apps, there were 30 apps rated D, 71 rated C, 226 rated B, 1384 rated A and 44 rated A+. On the other hand, in Group 2, there were 4,603 apps in total where 52.4% (2,414 apps) have a privacy score in our dataset that was shown to the participants, which include 29 apps rated D, 73 rated C, 289 rated B, 1955 rated A and 68 rated A+. Figure 5.5 shows the participant-specific number of apps with a privacy score in our database, for both Groups 1 and 2. This figure shows that there are two participants in each group who can be considered as extreme participants as they had more than 100 rated apps on their mobile devices. On the contrary, there are four participants had an unusually low number of (less than 5) rated apps.
5.5.3 Lab-based Results

Although the number of participants in the lab-based study was not large enough, we still report the result of this study as a pilot. The study recruited only 12 and 10 participants for Groups 1 and 2, respectively. 5 participants in Group 1 made at least one uninstall attempt and only one participant substituted one B rated app. Those five participants removed one C-rated app, two B-rated apps and one app without a privacy score. Nonetheless, in the semi-structured interview and the survey participants often expressed that they had considered removing some apps due to their low privacy scores. Particularly, 7 users in Group 2 considered removing 4 D-rated apps, 1 C-rated apps, and 2 B-rated apps. In Group 1, only two participants reported that they had considered uninstalling a D-rated and a C-rated apps. Due to the fact that the number of participants in this study is not sufficient, the statistical tests was not carried out.
5.5.4 Privacy-Related Decisions

There are three variables that have been analysed to capture the participants’ behaviours in the crowdsourcing-based study. The first variable is the number of substituted/removed apps due to their low privacy scores, which will be referred to as $SR$. The value of this variable was obtained via the collected data on the server side and confirmed from the reported answers of the survey. The second variable is the number of apps that participants had considered removing but they did not for some reasons, referred to as $CR$. The value of this variable could be obtained only from the participants’ responses to the survey. The last variable is the number of all app uninstallations observed during the experiment, which will be referred to as $PE$. The value of this variable is automatically obtained as the PAItroid app monitored and recorded all app uninstallations during the experiment. It was noticed that participants had removed some low-score apps without reporting them in the survey. This was considered as a potential positive effect caused by the use of PAItroid app.

Before going deep into analysis in this subsection, let us have a look at the average percentage of uninstalled apps with each specific privacy score in Groups 1 and 2, where the average percentage is calculated as the average of each user’s app uninstallation rate. The line-graphs in Fig. 5.6 show the average frequency of uninstalled apps with different privacy scores and for different groups. The graphs suggest that the behavioural nudging effect in Group 1 is higher than in Group 2. Furthermore, it seems that the impact of the PAItroid app on the uninstallation rate increases as the privacy level decreases (goes from A to D). A number of statistical tests were conducted to confirm the observations and to produce more insights. It is worth mentioning that there was not a single case that an A+ app was removed or replaced during the experiment.

As said before about Group 3, the reason of having it is to monitor users’ uninstall rate. Hence, the app was just a blank UI where workers only needed
to access the app once a day. Rather surprisingly, we did not see any case of app uninstallation among the 76 participants who enrolled into this group. It is expected that some users might install and uninstall an app in less than 24 hours, but that is not of an interest to this study. Data from Group 3 can tell us the “natural” rate of app uninstallation as a reference for us to check how many app uninstalls observed in Groups 1 and 2 may not be the consequence of using the PAltRoid app. Accepting the result that normally no participant would uninstall any apps within a week (as what we observed for Group 3), all uninstall decisions in Groups 1 and 2 could be considered as a result of using the PAltRoid app (the full version and the simplified version without alternative apps). In Table 5.1, the numbers of uninstalled apps in Groups 1 and 2 across all participants are shown, based on the assumption that all PE decisions were influenced by the PAltRoid app. The data in Table 5.1 shows that the effect on Group 1 is higher than Group 2. Taking into account that Group 3 did not record any uninstall calls, we believe that the PAltRoid app did help motivate participants in Groups 1 and 2 to make safer privacy-related decisions.

Table 5.1: App uninstallation statistics in Groups 1 and 2, assuming PE decisions are all valid

<table>
<thead>
<tr>
<th>Compare element</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of users who uninstalled app(s)</td>
<td>32(42.11%)</td>
<td>21(27.6%)</td>
<td>0</td>
</tr>
<tr>
<td>No. of uninstalled D-rated apps</td>
<td>6(20%)</td>
<td>4(13.8%)</td>
<td>0</td>
</tr>
<tr>
<td>No. of uninstalled C-rated apps</td>
<td>11(15.5%)</td>
<td>4(5.5%)</td>
<td>0</td>
</tr>
<tr>
<td>No. of uninstalled B-rated apps</td>
<td>18(8%)</td>
<td>5(1.7%)</td>
<td>0</td>
</tr>
<tr>
<td>No. of uninstalled apps without a score</td>
<td>34(2.1%)</td>
<td>20(0.9%)</td>
<td>0</td>
</tr>
</tbody>
</table>

The number and percentage are calculated based on the total number of apps/participants in that group.

From a statistical point of view, the impact of the PAltRoid app on users’ app uninstallation behaviours is studied as follows. To investigate the impact of the PAltRoid app on each individual participant’s behaviour, further analysis was conducted. The app uninstallation rate for each privacy score per participant was calculated. Then, Welch’s Two Sample t-test was applied using the app uninstallation rate to determine whether there is a significant difference between the means of two groups. The t-test was chosen as the dependent variable was normal (the uninstall rate for each participant). Table 5.2 clarifies that there is a significant difference in uninstalled B-rated apps between Groups 1 and 2. However, that is not the case for C- and D-rated apps which may be explained by the much smaller number of such apps. Moreover, the not small variance of the number of apps each user means the app uninstallation rate may not be a robust indicator allowing us to observe the expected effects.

As a consequent, participants’ decisions were dichotomised to avoid using the app uninstallation rate. While the dependent variable, the participant’s behaviour is defined as a binary variable: the value 0 means that the participant did not take
Table 5.2: Two sample t-test results on uninstalled apps with different privacy scores, for Groups 1 and 2

<table>
<thead>
<tr>
<th>Privacy Score</th>
<th>Group 1 mean</th>
<th>Group 2 mean</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>0.11842</td>
<td>0.02961</td>
<td>0.02126</td>
</tr>
<tr>
<td>C</td>
<td>0.075</td>
<td>0.04386</td>
<td>0.3865</td>
</tr>
<tr>
<td>D</td>
<td>0.05263</td>
<td>0.03289</td>
<td>0.5425</td>
</tr>
</tbody>
</table>

any app uninstallation or replacement action, and the value 1 means she/he took at least one such action against an app. The independent variable here is the privacy score of the app that the user uninstalled. This variable is categorical and has five values: A+, A, B, C, and D. The values that of an interest in this study are B, C and D, as they indicate problematic privacy levels. In this study design, the comparison is between independent groups. Since the dependent variable can be considered as categorical here, a number of $\chi^2$ tests were applied to test the first hypothesis at 5.3.2. In Table 5.3, a comparison of participants’ responses between groups on the three outcome variables $SR$, $CR$ and $PE$. The $\chi^2$ test results on $SR$ tell that the participant’s app uninstallation behaviour in Group 1 is significantly different from that in Group 2. That leads to the conclusion that participants who saw alternative apps in Group 1 were influenced more towards safer privacy-related decisions. However, the results on $CR$ and $PE$ did not show a significant difference between Groups 1 and 2. The $SR$ results may indicate that participants were affected by the proposed deign when they reported that they considered removing some low privacy scoring apps.

Table 5.3: $\chi^2$ test results for analysing participants’ app uninstallation behaviours

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>$\chi^2$</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SR$: Group 1 vs. Group 2</td>
<td>4.7896</td>
<td>0.02863</td>
</tr>
<tr>
<td>$CR$: Group 1 vs. Group 2</td>
<td>0.30536</td>
<td>0.5805</td>
</tr>
<tr>
<td>$PE$: Group 1 vs. Group 2</td>
<td>1.9891</td>
<td>0.1584</td>
</tr>
</tbody>
</table>

As earlier, a binary variable was defined for each participant and each problematic privacy score (D, C, B, and no rating). Then, a number of Wilcoxon rank sum test (due to the fact that dependent variable here is categorical) were conducted to find out if participants in Group 1 responded differently from those in Group 2. Table 5.4 depicts the result of running these tests. The results seem close to those from the t-test shown in Table 5.2, showing a significant difference only for B-rated apps.
Table 5.4: Wilcoxon rank sum test results on uninstalled apps with different privacy scores, for Groups 1 and 2

<table>
<thead>
<tr>
<th>Privacy Score</th>
<th>W</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>3268</td>
<td>0.01692</td>
</tr>
<tr>
<td>C</td>
<td>3040</td>
<td>0.232</td>
</tr>
<tr>
<td>D</td>
<td>2888</td>
<td>1</td>
</tr>
<tr>
<td>Null</td>
<td>3078</td>
<td>0.2915</td>
</tr>
</tbody>
</table>

5.5.5 Analysis of Participants’ Subjective Feedback

In this subsection, participants’ subjective feedback on the PAltRoid app is analysed. The feedback was collected from the online survey filled by participants at the end of the study. The analysis is presented here on two parts, the first showing how participants responded to assessment questions on the survey and the second showing a sentiment analysis of their reviews. We focus on Groups 1 and 2 only because participants in Group 3 did not see any design elements so had nothing to comment.

General Assessment of the App Design

To know participants’ assessment on different aspects of the app design, several questions were asked as part of the survey. Some questions are direct such as ‘Do you think the tested app is useful?’ followed by another saying ‘Please tell us why you think the app is/isn’t useful’. In Group 1, 60 picked ‘Yes’ (79%), 14 chose ‘Maybe’ (18.4%), and only 2 said ‘No’ (2.6%). In Group 2, the responses were as follows: 47 chose ‘Yes’ (61.8%), 22 chose ‘Maybe’ (29%), and 7 chose ‘No’ (9.2%). The results showed that most participants found the privacy ratings are useful and showing alternative apps made more feel the app useful. Moreover, the question ‘How would you assess the design of the tested app?’ was asked where participants had to choose one of the following three options: ‘simple and clear’, ‘needs some time to be understood’, ‘confusing and can be further improved’. Table 5.5 shows that participants in Group 1 were slightly more positive than those in Group 2. Yet, the majority in both groups believed that the design of the app was simple and clear. Another related question is ‘Do you think you gained knowledge while using the tested app?’ In Group 1 three workers answered that with ‘No’ to that, 10 were ‘Not sure’ and the rest said ‘Yes’ (82.9%), whereas in Group 2 eight said ‘No’, 14 ‘Not sure’ and the rest ‘Yes’ (63.2%). As a whole, participants in Group 1 seemed more positive about the knowledge acquisition question (i.e., the value on privacy awareness enhancement).
Table 5.5: Participants’ response to the question ‘How would you assess the design of the tested app?’

<table>
<thead>
<tr>
<th>Group</th>
<th>Simple and clear</th>
<th>Needs some time to be understood</th>
<th>Confusing</th>
<th>Can be further improved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57</td>
<td>9</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>15</td>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

Sentiment Analysis of Participant’s Reviews

In this part, the sentiment analysis on participants’ reviews is presented. This kind of analysis allows to understand whether users’ feedback about PAltRoid is positive or negative. Using a very basic sentiment analysis package in ‘R’ which is called (sentimentr), every participant’s review was given a positive/negative score, calculated by counting the number of positive and negative words in the text. Although it is very simple, it was applied as a quick test to have a sense of the data dealt with. For Group 1, 78.95% of the comments are positive and 21.05% are negative, while 72.86% are positive and 27.14 are negative at Group 2. Nevertheless, these figures can be misleading and it has been required to do a further investigation using more advanced sentiment analysis tools.

To approach the sentiment analysis more in depth, we decided to extract emotion-related information from the reviews. Using a ‘WordEmotion Association Lexicon’ algorithm [140], the scores of eight emotions were generated for reviews from both groups. Figure 5.7 highlights these emotion scores along with negative and positive scores. Looking at negative emotions which are anger, disgust, fear, sadness and the overall negative score, it can be seen that Group 2 scores higher than Group 1, implying participants in Group 1 found the full version of the app less frustrated than Group 2. The positive emotion scores are almost the same for both groups, implying participants in both groups find the full and simplified versions of the app equally useful. Nonetheless, we noticed that some emotion scores are misleading as the meaning can be different in this context. For example, participants used a lot of security related terms such as, ‘secure’, ‘protect’, ‘privacy’, etc., which eventually affected the score and considered to be negative while they were meant to be praising the tested app. Considering the issues we encountered using automated tools, we decided to conduct a manual sentiment analysis.

To conduct the manual sentiment analysis, two independent researchers, Dr. Abdulaziz Alanazi from Department of Physics and Mr. Saleh Jalbi from Department of Civil and Environmental Engineering, were recruited from the University of Surrey. We recruited them using a broadcast message to several groups on ‘WhatsApp’ application. The two researchers were asked to identify positive and negative phrases from participants’ feedback. Any phrases that are not either praising or criticising the design of the app are considered neutral. This procedure was blindly
Figure 5.7: Emotion scores for Groups 1 and 2
done to avoid any unconscious bias, meaning that all comments were given to the researchers without disclosing to which group they belong. After manually analysing all participants’ feedback, it was confirmed that a majority of participants in both groups have a positive feeling toward the PAI\textsc{t}Roid app. Yet, participants in Group 1 (72.37\% of them) is significantly more positive than those in Group 2 (62.68\%). 15 participants in Group 2 gave negative comments and mostly they were wondering why there was no solution to low-scoring apps or how the scores are generated. For Group 1, there were only 6 participants who negatively commented against the tested app. Some of their comments were criticising the apps without a privacy score, which is not directly related to the PAI\textsc{t}Roid app. Some other participants believed that the PAI\textsc{t}Roid app needs to be more interactive such as working in the background and dynamically monitor new apps installed and their activities. For the rest of the reviews, there were 10 neutral comments in each group where the opinions do not support a particular group.

5.6 Discussions

Experimental studies done in this study have proven the positive impact of PAI\textsc{t}Roid on participants’ privacy-related decisions. The feedback from participants shows that most of them did pay attention to privacy warnings shown by the PAI\textsc{t}Roid app and took necessary actions. From participants’ responses to survey questions about their background knowledge of privacy, it can be seen that several of them were aware of privacy issues yet they still had privacy invasive apps on their devices. The well-known privacy paradox between privacy and utility was also observed in our data. For instance, we noticed that several participants had mentioned they did not like the the app ‘Uber’ due to its very low privacy score (D), but they could not give it up as its utility is very essential to them. In another case, some participants explicitly asked for better alternative apps or good practices on how to use apps with low privacy ratings. A few participants even asked for the user interface to be further improved because they really liked the app and would like it to be even more interactive, e.g., its working in the background to pop up privacy-related messages whenever necessary. Pleasantly, two participants said they had been looking for a similar app that works like an anti-virus app but for privacy protection. Another useful thing to mention here is about negative comments from Group 2. The majority of the participants were not happy with the design for one reason – they felt they had been left without a solution (i.e., alternative apps), which reflects the usefulness of the full version of the PAI\textsc{t}Roid app and their willingness to take actions when proper support is provided.

As far as we know, in the literature, there is no similar work that PAI\textsc{t}Roid can be directly compared with. However, the closest work to look at is PrivMetrics [3]. Although its main focus is not about improving the user interface like PAI\textsc{t}Roid, it has a way of displaying the result that may seem relevant. It investigates the implications of third-party apps collecting users’ personal information and provide
the user with relevant details to help users decide about their apps. PrivMetrics utilises a different way of assessing apps’ privacy from PrivacyGrade, which is an intrinsic difference. The focus of the presented work is to display related privacy details in a new user interface to help users to make more informed privacy-related decisions, so our work is independent of the privacy scoring algorithms used.

The remedy to apps with low privacy scores that PAltRoid offers is alternative apps with better privacy scores and similar functionalities. The main goal here is to nudge users towards safer privacy states and foster more risk-aware behaviours. In order to make this remedy work in real world, these alternative apps must be similar enough while having a better privacy score. This task proved not trivial. The machine produced alternatives incur computational overheads and come with the price of many false positives. That was the reason for us to go for a crowdsourcing-based approach based on AlternativeTo, but it currently does not offer a large number of alternative apps and not for all privacy invasive apps.

People might wonder why the crowdsourcing workers were paid less than lab-based ones, or even than the minimum hourly wage. A short answer to that question is because that what was given is the market price and even slightly higher. In such platforms, users sign up to do tasks for a low pay. There are some reasons to explain that. One of them is that this kind of jobs is not a full time one to compare it to the minimum wage, and mostly workers do these tasks in their free times or as a second source of income, despite the fact that most workers usually do multiple tasks in the same hour. Doing a crowd sourcing task does not require attendance somewhere or at sometime. This kind of situation affected how HIT requesters set the pay of their advertised tasks. Moreover, that could be a reason why quality of this platform is less than lab-based way. In a study that analysed AMT payroll [141], authors said: “workers earned a median hourly wage of only \$2, and only 4% earned more than \$7.25/h” where they surveyed 2676 workers performing 3.8 million tasks. In this presented experiment, workers from AMT were compensated with \$5 for a total work that does not exceed two hours on average. This decision has been made after considering all above thoughts and for a very important reason which is avoid over-motivating workers so their behaviour will not be tampered.

Lab-based and crowdsourcing based studies have their own pros and cons. For that reason, we cannot replace each other. One of the challenging tasks to accomplish this work was the participants recruitment. The aim of acquiring a sufficiently large number ended up with spending longer time than planned. Furthermore, the nature of this study requires a period of time to allow users experiment the app in order to extract their realistic behaviour, that also made it harder, despite that some users were not keen because they were worried about their privacy on their personal phones. This presented work was done mainly with the crowdsourcing based study, with a very limited validation in a lab-based pilot. Nonetheless, it will be helpful to conduct a proper lab-based study and more crowdsourcing based studies in future (maybe using different platforms and other designs) to further validate the results we obtained.
5.7. **FUTURE WORK**

There is one behaviour we observed in our study that arouse curiosity and therefore future studies. As expressed earlier, there were many apps that do not have a privacy rating in the PrivacyGrade dataset. It was planned that those apps are not in the scope of this study, and so such apps were intentionally displayed at the bottom of the list of all apps. Interestingly, some participants in both Group 1 and 2 decided to uninstall some of such apps. Since this phenomenon was discovered only during the data analysis phase, we could not ask those participants about why they took such action. We hypothesize that this may be some psychological effect where people lean towards reducing unnecessary risks if they can be done easily (e.g. those low-rated apps are not actually important or used much).

There are some aspects of the design of the PAltRoid app which limit what we could have achieved in our study. Surely, one of them is not all apps are privacy rated. The presented piece of work is dependent on what the PrivacyGrade dataset offers but this dataset covers free apps only and it is quite outdated (last quarter of 2016). However, it was the only available dataset we could obtain access during our study. Some participants clearly expressed their annoyance of seeing apps without a privacy score. Another limitation is the lack of sufficient user-friendly details about how the privacy scores are calculated. The PAltRoid app’s user interface has a section on ‘details’ tab that can show how PrivacyGrade calculates the ratings. Yet, some participants were not satisfied with the information given and wanted more in-depths explanation especially for apps with low privacy scores. They believed that knowing more about the reasons behind the low privacy scores will help them to decide how to use the app in a more risk-aware manner such as blocking individual permissions for access to some resources (which is now allowed under the new dynamic permission model in newer version of Android OS).

The biggest challenge we encountered in our study is about identification of right alternative apps satisfying both utility and privacy requirements. Machine generated apps in this context can be similar but not always a better alternative. You could provide ‘A’ taxi hire app as an alternative to ‘B’ but it might not work in all countries or cities. Some apps are difficult to be replaced as they provide a unique and excellent service (while being privacy invasive). Therefore, this part of the design should be maintained properly. Otherwise, users will gradually lose trust of the proposed design. In relevance to this limitation, providing a download page or apk file for alternatives is not easy. Again, not all apps are available at all Google Play markets. Offering a cure name without providing it can be very frustrating.

5.7 **Future Work**

While working on PAltRoid there are several identified future ideas. Some of them are to improve the current design and others are a different way of displaying privacy related information. This section reports all planned future work.
5.7.1 Improvement on Current Design

In regard to improving the current design, there are some ideas built based on results and participants’ feedback. First of all, in case more details can be obtained about these rating, they will be added to the design where users can see them. On the other hand, alternative apps require similar attention where they need to be updated directly from AlternativeTo. Its API is currently unsupported. Nonetheless, one idea can be more straightforward. The aim here is to allow users add their own suggestions to the alternative section which will be connected to the server. Moreover, users will have an up-vote and down-vote buttons to improve current displayed similar apps. When PAltRoid becomes available at application stores, the hope is that users will help in enriching the data. Applying human-in-the-loop (HITL) approach will naturally make this issue disappear.

5.7.2 New Ideas for Different Design Interfaces

On light of result from this Chapter, it has been planned to design different interfaces to communicate both types of subjective and objective ratings. There are many different ways of visualising privacy scores, and here we explain few of them as examples. One of the planned goals is to use a different way of displaying objective privacy scores than PAltRoid. Apps privacy ratings will be displayed in a 2-D colourful bar chart. This design is meant to present a comprehensive sense of the privacy level on user’s mobile. This will be achieved by showing the chart on one screen view. Each bar will represent a privacy level of one app and can be clicked. That will allow to show details about the rating and alternatives. With more user studies on current and future proposed designs, it will be possible to confirm which design is more efficient and in what context.

Another idea is to make use of subjective user ratings and comments on apps publicly available on different app stores. The subjective information about user apps will be visually displayed in a friendly interface. For the reason that comments and feedback are going to be lengthy and need to be validated, the plan is to do an analytic processing on the data. Analysis will involve contextualising privacy related details from the comments. Also, there are some thoughts to make the app more intelligent to make its data personalised by learning from user usage of it. Ultimately, user studies will explain how users perceive different types of ratings and variant design interfaces, which is believed to be useful in improving user experience with mobile apps in terms of privacy.

5.8 Conclusion

This chapter presents PAltRoid, a system for helping users to make more informed privacy decisions about mobile apps on on their mobile devices. It was implemented as a mobile app that communicates privacy scores of mobile apps in a
user-friendly interface. It also comes with better alternative apps to installed apps with low privacy ratings, as potential solutions for replacing the problematic apps. A crowdsourcing-based and a pilot lab-based user-studies have been conducted to assess the efficiency of this presented software. The experiments provided evidence that participants’ decisions were improved with the help of the PAItRoid app.
Chapter 6

Conclusion and Future Work

This chapter gives an overview of the entire works presented in this PhD thesis. Moreover, it draws some planned future work and possible research directions.

6.1 Overview

In this thesis, the mobile privacy leakage problem has been investigated to enhance user experience in this context. In the beginning, a review on the background of this problem has been presented to comprehend current related issues in order to determine possible solutions (Chapter 2). It is obvious that the research has contributed a variety of technical remedies to this problem. While reproducing some of the related work from research community, namely TaintDroid and ScrubDroid, we realised that there is a need for a benchmarking system that can evaluate privacy protection tools in run-time.

To fill that gap, PPAndroid-Benchmarker has been developed for analysing mobile privacy detection and protection apps (Chapter 3). That will help developers and users to know where they stand in the middle of variant provided privacy protection tools. We evaluated our system by testing against 165 tools gathered from research and industry. We also learned that real-time dynamic monitoring tools perform better than others, as they operate with the underlying operating system. This approach can be very challenging and often impossible for the end-user to apply as it needs an advanced technical level. Therefore, we decided to take a step back and see how users choose their apps, which are the source of threat on users’ privacy. That will help to know what affect their decision making process.

No matter solutions are provided on the technical side, users are always prone to make mistakes. Therefore, the best procedure, as author believes, is to help educate the user in making better calls. As an action towards understanding the users, a user perceived trust of subjective and objective ratings of mobile apps privacy has
been studied (Chapter 4). The author came to know that the subjective judgement and the behavioural pattern of a user affect the decision of trusting subjective or objective rating of apps. Thus, it has been decided to investigate the possibility of supporting user decisions to be more privacy-aware.

In Chapter 5, the author has shown how PAltRoid helps to rise the privacy awareness of the user. Communicating privacy details and providing better alternative apps can help users in making more privacy-related decisions. The interface design can be an efficient approach to deliver privacy warnings about installed apps. PAltRoid uses a colourful horizontal stacked bar chart to show the privacy level of each app on the phone and the apps are ordered based on privacy score from weak (top of screen) to strong (bottom of screen). This way of displaying information is purposely made to increase users attention of privacy threats. Additionally, each threatening app will be offered a possible improvement when alternatives are taken, when possible. The experimental studies done on this app have shown the participants positive interaction towards privacy-aware decisions.

6.2 Selected Future Work

This section outlines some future research directions that are planned to follow up in the privacy research of mobile computing. Some of the future directions are about extra ideas for further improving what have been done in this PhD work, while others are identified gaps in the wider area of research.

6.2.1 Improving PPAndroid-Benchmarker

In Chapter 2, we explained the future plans to improve PPAndroid-Benchmarker which can be summarised as follows. One component to add is the Test Apps Generator. This component will help to benchmark static analysis tools. The goal is to allow generation of apps with different privacy leakage capabilities and different leaking methods. By using the source code of the Benchmarker App and the user’s descriptions of the test apps wanted, this component will generate a number of apps for feeding into the tools to be tested.

Another component to add is the Reconfigurability Engine. This will allow 1) adding and removing sources and sinks; 2) adding and reconfiguring more information processing operations and tricks against static and dynamic analysis techniques.

Furthermore, we are also planning to add the Off-line Analyser. This aims at producing more visualised results to ease understanding of the results. In addition, this component can be designed to generate a score of the tested app to give a sense of the privacy protection level.
6.2. SELECTED FUTURE WORK

6.2.2 Users Perceived Trust of Mobile Apps’ Privacy Ratings: More User Studies

We believe that a lab-based study needed to confirm results from the crowdsourcing based studies reported in Chapter 4. In addition, further investigation is required to fully comprehend users’ perception of app privacy ratings, and how they form their decisions. Future studies in this area should consider to use realistic apps and possibly real subjective and objective ratings to enrich evidence of the findings.

Another possible direction is to extend current user studies in this field. We aim to study different factors that could impact users perceived trust. Some of the factors that we plan to study is the use of strong wording in apps description, as well as brand names. We would like to see how such factors influence users’ choices of mobile apps.

6.2.3 Extending PAltRoid

As mentioned in Chapter 5, several plans have been set to improve the current app. Firstly, we aim to make privacy scores and offered alternative apps are up-to-date by directly fetching data from the source websites. Also, we hope to provide more useful information behind the privacy scores to help users understand how to use their apps and why some apps are more privacy invasive. Furthermore, we plan to add in human-in-the-loop (HITL) feature into PAltRoid to allow users to add their own suggested alternative apps and vote on current provided ones (i.e., incorporating a service similar into AlternativeTo to PAltRoid).

More user interface design and features are also a possible future work to help achieve the same goal of PAltRoid. There are many other ways to display privacy scores and the alternative apps. For instance, the 1-D bar charts used can be extended to 2-D or even 3-D to show a more comprehensive view of the privacy levels of all installed apps (e.g., showing the privacy scores of all installed app as a 2-D image with different colour patches, where each colour patch represents an installed app). Besides that, another feature is planned to utilise subjective user ratings and comments of apps already existing on app stores. Such data can be displayed in a user-friendly way to inform users about other how other users see an app with a bad privacy score and all the alternatives, e.g., the comments may be shown as a word cloud so that the most important keywords can be communicated to the user and more detailed information can be revealed if the user clicks on a particular word. More user studies will be needed to find out which user interface can engage users better to make more informed privacy-related decisions.

6.2.4 Wider Directions

It has been remarked that many privacy protection tools use a limited and static list of information sources, leaving other sources unmonitored. Among those missed sources, some are user specific such as files storing the user’s personal information...
and other sensitive data. We will study how to apply machine learning tools to identify such personalised information sources that deserve protecting. This likely requires a human-in-the-loop approach as the learning process has to be helped by the user himself/herself.

In addition, almost all mobile privacy leakage detection tools simply produce a notification for every detected leakage and leave further actions to end users who often feel annoyed or confused on how to respond. So there is a need to improve the user’s overall experience from detection to further actions against privacy invasive apps. There is a clear gap here and much less work has been done on helping users make final decisions after privacy leakage attempts are detected. We have developed some ideas in this direction, and two of such ideas are to develop “smarter” filters to show users fewer notifications and to rank such notifications in a more user-relevant (i.e., personalised) manner. We plan to employ machine learning techniques to learn from users’ feedback of notifications to optimise the “smarter” filters and to provide more useful tips to users for further actions against potential privacy leakages.

Lastly, little work has been done on understanding how leaked personal information is actually used by receivers of such information in the real world. We expect such information can be collected through several useful techniques such as honeytokens and digital watermarking, which can allow tracking leaked data along the information usage chain. This direction has a flavour of active forensics so can potentially help both mobile device users and law enforcement agencies.
Appendices
Appendix A

Participant Information Sheet

In this appendix, the PIS of the lab-based study followed by PIS of the crowdsourcing-based study can be found, respectively.
INFORMATION SHEET FOR PARTICIPANTS

Title of Study: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs

University of Surrey Ref:

Ethics Reference Number: [INSERT ONCE PROVIDED BY REVIEW BODY]

YOU WILL BE GIVEN A COPY OF THIS INFORMATION SHEET

Introduction

My name is Saeed Alqahtani, a PhD student at the University of Surrey/Department of Computer Science. I would like to invite you to take part in this research project which forms part of my PhD research. You should only participate if you want to; choosing not to take part will not disadvantage you in any way but you must be a user of an Android device. Before you decide whether you want to take part, it is important for you to understand why the research is being done and what your participation will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask me if there is anything that is not clear or if you would like more information.

What is the purpose of the study?

The purpose of the research is to evaluate the effectiveness of a mobile app user interface, in terms of notification and user engagement in privacy-related decisions.

Why have I been invited to take part?

You have been invited to take part in this study to collect your view and opinion on one of the mobile app user interfaces in this experiment. At least 20 participants to be recruited for each mobile app interface.

Do I have to take part?

No, you do not have to participate. There will be no adverse consequences in terms of your legal rights and your care / treatment / employment status / education, if you decide not to participate or decide to withdraw at a later stage. You can withdraw your participation at any time. You can request for your data to be withdrawn up to one month after your participation without giving a reason and without prejudice.

Version 2 - 11/09/2017

Study Title: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs
If you withdraw from the study, data which is not identifiable will be retained because we cannot trace this information back to you. Identifiable data will be deleted. No further data would be collected nor any other research procedures carried out on or in relation to you.

What will happen to me if I take part?

If you agree to take part, you will be asked to attend 2 sessions at the University of Surrey, Guildford, of about 1 hour duration each. At the first visit, you will be asked to sign a consent form. If you do decide to take part you will be given this information sheet to keep and a copy of your signed consent form. Also, you will be asked to install and use a mobile app that will be provided to you for two weeks, which you need to access once a day at minimum.

On the second visit, you will be invited to come back to uninstall the mobile app and provide us with your feedback and opinion on the mobile app used and your personal experiences in using similar apps in a short semi-structured interview. If you give consent for audio recording, the interview will be recorded; otherwise the research coordinator will make notes on the content of the interview. The second visit will take place after two weeks of the first visit.

The mobile app will only collect a list of your installed apps only (and not their usage) and send it to a secure server that I manage for data analysis. No personal or other information is collected at any time during the experiment. You are only required to access the app once a day during two weeks.

What are the possible benefits and risks of taking part?

You will gain monetary compensation which is £10 for attendance at each session. That means, if you attend both sessions you will be compensated by £20, knowing that travel costs are not refunded. However, you need to keep the app in your mobile for two weeks, and access it at least once a day. Furthermore, you will hopefully increase your privacy awareness on mobile devices.

Will my taking part in the study be kept confidential?

What is said in the interview and/or data collected is regarded strictly confidential and will be held securely until the research is finished. All data for analysis will be anonymised. In reporting on the research findings, I will not reveal the names of any participants or the organisation where you work.

All project data (e.g. consent forms) will be held for at least 6 years and all research data for at least 10 years in accordance with University policy and that personal data is held and processed in the strictest confidence, and in accordance with the Data Protection Act.

Version 2 - 11/09/2017
Study Title: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs
All information gathered will be held for long-term storage on University secure servers. Hard files will be kept in locked cabinets within the Department of Computer Science. No identifiable data will be accessed by anyone other than me (Saeed Alqahtani), members of the research team (Dr Nouf Aljaffan and Dr Shujun Li) and authorised personal from the University and regulatory authorities for monitoring purposes. Anonymity of the material will be protected by members of the research team.

No data will be able to be linked back to you. You will not be identified in any reports/publications resulting from this research and those reading them will not know who has contributed to it / with your permission we would like to use verbatim quotation/audio recordings in reports.

How is the project being funded?

The project is being funded by King Saud, Saudi Arabia and the research is organized by the University of Surrey. This study has been given a favourable ethical opinion by the University Research Ethics Committee.

What will happen to the results of the study?

We expect that the outcomes of the research will be published as a research paper.

Who should I contact for further information?

If you have any questions or require more information about this study, please contact me using the following contact details:

Saeed Alqahtani (Researcher)  
Email: s.alqahtani@surrey.ac.uk  
Phone: 01483 686085

Dr Nouf Aljaffan (Researcher)  
Email: n.aljaffan@surrey.ac.uk  
Phone: 01483 686085

Dr Shujun Li (Supervisor)  
Email: shujun.li@surrey.ac.uk  
Phone: 01483 68 6057

Dr Helen Treharne (Head of Department)  
Email: h.treharne@surrey.ac.uk  
Phone: 01483 68 3161
What if something goes wrong?

Any complaint or concern about any aspect of the way you have been dealt with during the course of the study will be addressed, please contact Saeed Alqahtani, Principal Investigator via s.alqahtani@surrey.ac.uk in the first instance, his supervisor Dr Shujun Li via shujun.li@surrey.ac.uk, or Head of Department Prof Mark Plumbley via m.plumbley@surrey.ac.uk.

The University has in force the relevant insurance policies which apply to this study. If you wish to complain, or have any concerns about any aspect of the way you have been treated during the course of this study then you should follow the instructions given above.

Thank you for reading this information sheet and for considering taking part in this research.
INFORMATION SHEET FOR PARTICIPANTS

Title of Study: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs

University of Surrey Ref:

Ethics Reference Number: UEC/2017/074/FEPS

YOU WILL BE GIVEN A COPY OF THIS INFORMATION SHEET

Introduction

My name is Saeed Alqahtani, a PhD student at the University of Surrey/Department of Computer Science. I would like to invite you to take part in this research project which forms part of my PhD research. You should only participate if you want to; choosing not to take part will not disadvantage you in any way but you must be a user of an Android device. Before you decide whether you want to take part, it is important for you to understand why the research is being done and what your participation will involve. Please take time to read the following information carefully and discuss it with others if you wish. Ask me if there is anything that is not clear or if you would like more information.

What is the purpose of the study?

The purpose of the research is to evaluate the effectiveness of a mobile app user interface, in terms of notification and user engagement in privacy-related decisions.

Why have I been invited to take part?

You have been invited to take part in this study to collect your view and opinion on one of the mobile app user interfaces in this experiment. At least 20 participants to be recruited for each mobile app interface.

Do I have to take part?

No, you do not have to participate. There will be no adverse consequences in terms of your legal rights and your care / treatment / employment status / education, if you decide not to participate or decide to withdraw at a later stage. You can withdraw your participation at any time. You can request for your data to be withdrawn up to one month after your participation without giving a reason and without prejudice.

Version 4 - 13/06/2018

Study Title: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs
If you withdraw from the study, data which is not identifiable will be retained because we cannot trace this information back to you. Identifiable data will be deleted. No further data would be collected nor any other research procedures carried out on or in relation to you.

**What will happen to me if I take part?**

If you agree to take part, you will be asked to attend 2 sessions on the crowd-sourcing platform (Amazon MTurk/ Figure-Eight) of no more than 1 hour duration each. At the first task, you will be given this information sheet to keep. Also, you will be asked to install and use a mobile app that will be provided to you for one week, which you need to access once a day at minimum.

On the second task, you will be invited to come back to uninstall the mobile app and provide us with your feedback and opinion on the mobile app used and your personal experiences in using similar apps by filling an electronic survey. The second visit will take place after one week of the first task.

The mobile app will only collect a list of your installed apps only (and not their usage) and send it to a secure server that I manage for data analysis. No personal or other information is collected at any time during the experiment. You are only required to access the app once a day during one week.

**What are the possible benefits and risks of taking part?**

You will gain monetary compensation which is £10 for attendance at each session. That means, if you attend both sessions you will be compensated by £20 However, you need to keep the app in your mobile for one week, and access it at least once a day. Furthermore, you will hopefully increase your privacy awareness on mobile devices.

**Who is Handling My Data?**

The University of Surrey, as the sponsor, will act as the ‘Data Controller’ for this study. We will process your personal data on behalf of the controller and are responsible for looking after your information and using it properly. This information will include your (installed apps’ names, gender, age, nationality and mother language), which is regarded as ‘personal data’. We will use this information as explained in the ‘What is the purpose of the study’ section above.

**Will my taking part in the study be kept confidential?**

What is said in the interview and/or data collected is regarded strictly confidential and will be held securely until the research is finished. All data for...
analysis will be anonymised. In reporting on the research findings, I will not reveal the names of any participants or the organisation where you work.

All information gathered will be held for long-term storage on University secure servers. Hard files will be kept in locked cabinets within the Department of Computer Science. No identifiable data will be accessed by anyone other than me (Saeed Alqahtani), members of the research team (Dr Nouf Aljaffan and Dr Shujun Li) and authorised personal from the University and regulatory authorities for monitoring purposes. Anonymity of the material will be protected by members of the research team.

No data will be able to be linked back to you. You will not be identified in any reports/publications resulting from this research and those reading them will not know who has contributed to it / with your permission we would like to use verbatim quotation/audio recordings in reports.

What will happen to my data?

As a publicly-funded organisation, we have to ensure when we use identifiable personal information from people who have agreed to take part in research, this data is processed fairly and lawfully and is done so on the basis of public interest. This means that when you agree to take part in this research study, we will use your data in the ways needed to conduct and analyse the research study.

All project data related to the administration of the project, (e.g. consent form) will be held for at least 6 years and all research data for at least 10 years in accordance with University policy. Your personal data will be held and processed in the strictest confidence, and in accordance with current data protection regulations.

Your rights to access, change or move your information are limited, as we need to manage your information in specific ways in order for the research to be reliable and accurate. If you decide to withdraw your data from the study, we may not be able to do so. We will keep the information about you that we have already obtained. To safeguard your rights, we will use the minimum personally-identifiable information possible.

You can find out more about how we use your information: https://www.surrey.ac.uk/information-management/data-protection and/or by contacting dataprotection@surrey.ac.uk

What if I want to complain about the way data is handled?

Version 4 - 13/06/2018
Study Title: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs
If you wish to raise a complaint on how we have handled your personal data, you can contact our Data Protection Officer Mr James Newby who will investigate the matter. If you are not satisfied with our response or believe we are processing your personal data in a way that is not lawful you can complain to the Information Commissioner’s Office (ICO) (https://ico.org.uk/).

For contact details of the University of Surrey’s Data Protection Officer please visit: https://www.surrey.ac.uk/information-management/data-protection

**How is the project being funded?**

The project is being funded by King Saud, Saudi Arabia and the research is organized by the University of Surrey. This study has been given a favourable ethical opinion by the University Research Ethics Committee.

**What will happen to the results of the study?**

We expect that the outcomes of the research will be published as a research paper.

**Who should I contact for further information?**

If you have any questions or require more information about this study, please contact me using the following contact details:

**Saeed Alqahtani (Researcher)**  
Email: s.alqahtani@surrey.ac.uk  
Phone: 01483 686085

**Prof Liqun Chen (Supervisor)**  
Email: liqun.chen@surrey.ac.uk  
Phone: 01483 684615

**Prof Shujun Li (Supervisor)**  
Email: shujun.li@surrey.ac.uk  
Phone: 01483 68 6057

**Dr Helen Treharne (Head of Department)**  
Email: h.treharne@surrey.ac.uk  
Phone: 01483 68 3161

**What if something goes wrong?**

Version 4 - 13/06/2018
Study Title: Improving notification and user engagement in privacy-related decisions via new mobile user interface designs
Any complaint or concern about any aspect of the way you have been dealt with during the course of the study will be addressed; please contact Saeed Alqahtani, Principal Investigator via s.alqahtani@surrey.ac.uk in the first instance, his supervisor Dr Shujun Li via shujun.Li@surrey.ac.uk, or Head of Department Dr Helen Treharne viah.treharne@surrey.ac.uk.

The University has in force the relevant insurance policies which apply to this study. If you wish to complain, or have any concerns about any aspect of the way you have been treated during the course of this study then you should follow the instructions given above.

Thank you for reading this information sheet and for considering taking part in this research.
Appendix B

Interview Schedule
Interview Schedule

During the interview, we would like to discuss the following topics:

Demographic Information
- Age, gender, education, first language and culture background, knowledge on ICT especially mobile devices and privacy, profession

Background Questions
- Privacy of mobile apps (what and why?)
- Mobile app threats on privacy (what?)
- Use of mobile privacy protection apps (what, when and how?)
- Improving privacy on mobiles (how?)

Assessment of the mobile app design
These will be asked in the second session:
- Usability and intuitiveness (how much?)
- Usefulness (what, how?)
- Current feedback (Enough, sufficient or adequate?)
- Information density (advantages and disadvantages)
- Gained knowledge (what?)
- Discussion on recorded behaviours (why?)
- Things to be changed or added
- Differences and similarity with other mobile privacy apps
Appendix C

Lab-Based Study Questionnaires

In this appendix, the lab-based study questionnaire of Group 1 followed by the questionnaire of Group 2 can be found, respectively.
Your Feedback on VAndroidRater

In this survey, we would like to collect your subjective view on different aspects of the tested mobile app. Furthermore, to help understand and analyse results of this lab study, we need to obtain your experience and background around privacy in mobile devices.

Please be aware that we will not collect any personal information that can be linked back to you, and all private details will be kept anonymous.

We would like to express our gratitude for taking part of this study and filling this questionnaire.

If you have any questions, please contact the research coordinator:

Saeed Alqahtani
Lecturer, Taibah University, Saudi Arabia
PhD Student, Department of Computer Science, University of Surrey, UK
Email: s.alqahtani@surrey.ac.uk

*Required

Demographic Questions

1. Age: *
   Mark only one oval.
   - 18-24
   - 25-34
   - 35-44
   - 45-54
   - 55-64
   - >=65

2. Gender *
   Mark only one oval.
   - Female
   - Male
   - Others

3. Your nationality/nationalities *
   you can enter more than one (separate them with commas)

   ______________________________________________
   ______________________________________________
   ______________________________________________

4. Your mother language(s) *
   you can enter more than one (separate them with commas)

   ______________________________________________
5. Highest degree or level of education *
   Mark only one oval.
   - [ ] PhD degree or equivalent
   - [ ] Master's degree or equivalent
   - [ ] Bachelor's degree or equivalent
   - [ ] Others

Background Questions

6. Are you aware of privacy issues that mobile apps can cause? *
   Mark only one oval.
   - [ ] Yes
   - [ ] No
   - [ ] Not Sure

7. Please name some mobile app threats on privacy:
   you can enter more than one (separate them with commas)
   - ____________________________
   - ____________________________
   - ____________________________

8. Do you use any mobile privacy protection apps? *
   Mark only one oval.
   - [ ] Yes
   - [ ] No

9. If previous question is 'Yes', name the protection app/apps you use:
   you can enter more than one (separate them with commas)
   - ____________________________
   - ____________________________
   - ____________________________

10. Please write optional free-formatted comments on how to improve privacy on mobile devices.
    - ____________________________
    - ____________________________
    - ____________________________
    - ____________________________
    - ____________________________
    - ____________________________
    - ____________________________
    - ____________________________
Assessment of The Mobile App Design

11. Do you think the tested app is useful? *
   Mark only one oval.
   □ Yes
   □ No
   □ Maybe

12. Please write why you think VAndroidRater is/isn’t useful. *

13. How would you assess the design of the tested app: *
   Mark only one oval.
   □ Easy and clear
   □ Needs some time to be understood
   □ Confusing
   □ Can be further improved

14. Do you think you gained knowledge while using the tested app? *
   Mark only one oval.
   □ Yes
   □ No
   □ Not sure

15. Did you feel that you need to remove/substitute apps with low privacy scores? *
   Mark only one oval.
   □ Yes
   □ No
   □ Not sure

16. Name apps that you substituted/uninstalled due to their problematic privacy scores:
   you can enter more than one (separate them with commas) leave empty if none
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
17. Name apps that you considered substituting/uninstalling them due to their low privacy rating:
you can enter more than one (separate them with commas) leave empty if none


18. How relevant are the provided alternative apps? *
Mark only one oval.

1 2 3 4 5
Irrelevant Relevant

19. What is your general feedback on the provided alternative apps? *


20. Please tell us how VAndroidRater design interface can be further improved.


21. Please write a free-formatted comments if you want to share any ideas/opinion about VAndroidRater app.


Powered by Google Forms
Your Feedback on VAndroidRater

In this survey, we would like to collect your subjective view on different aspects of the tested mobile app. Furthermore, to help understand and analyse results of this lab study, we need to obtain your experience and background around privacy in mobile devices.

Please be aware that we will not collect any personal information that can be linked back to you, and all private details will be kept anonymous.

We would like to express our gratitude for taking part of this study and filling this questionnaire.

If you have any questions, please contact the research coordinator:
Saeed Alqahtani
Lecturer, Taibah University, Saudi Arabia
PhD Student, Department of Computer Science, University of Surrey, UK
Email: s.alqahtani@surrey.ac.uk

*Required

Demographic Questions

1. Age: *  
   Mark only one oval.
   - 18-24
   - 25-34
   - 35-44
   - 45-54
   - 55-64
   - >=65

2. Gender *  
   Mark only one oval.
   - Female
   - Male
   - Others

3. Your nationality/nationalities *  
   you can enter more than one (separate them with commas)

   __________________________________________________________________________
   __________________________________________________________________________
   __________________________________________________________________________

4. Your mother language(s) *  
   you can enter more than one (separate them with commas)

   __________________________________________________________________________
5. Highest degree or level of education *
   Mark only one oval.
   ☐ PhD degree or equivalent
   ☐ Master's degree or equivalent
   ☐ Bachelor's degree or equivalent
   ☐ Others

Background Questions

6. Are you aware of privacy issues that mobile apps can cause? *
   Mark only one oval.
   ☐ Yes
   ☐ No
   ☐ Not Sure

7. Please name some mobile app threats on privacy:
   you can enter more than one (separate them with commas)

   __________________________________________________________
   __________________________________________________________
   __________________________________________________________

8. Do you use any mobile privacy protection apps? *
   Mark only one oval.
   ☐ Yes
   ☐ No

9. If previous question is 'Yes', name the protection app/apps you use:
   you can enter more than one (separate them with commas)

   __________________________________________________________
   __________________________________________________________
   __________________________________________________________

10. Please write optional free-formatted comments on how to improve privacy on mobile devices.

    __________________________________________________________
    __________________________________________________________
    __________________________________________________________
    __________________________________________________________
Assessment of The Mobile App Design

11. Do you think the tested app is useful? *
   Mark only one oval.
   - Yes
   - No
   - Maybe

12. Please write why you think VAndroidRater is/isn't useful. *

13. How would you assess the design of the tested app: *
   Mark only one oval.
   - Easy and clear
   - Needs some time to be understood
   - Confusing
   - Can be further improved

14. Do you think you gained knowledge while using the tested app? *
   Mark only one oval.
   - Yes
   - No
   - Not sure

15. Did you feel that you need to remove/substitute apps with low privacy scores? *
   Mark only one oval.
   - Yes
   - No
   - Not sure

16. Name apps that you substituted/uninstalled due to their problematic privacy scores:
   you can enter more than one (separate them with commas) leave empty if none

   ______________________________________________________
   ______________________________________________________
   ______________________________________________________
   ______________________________________________________
17. Name apps that you considered substituting/uninstalling them due to their low privacy rating:
you can enter more than one (separate them with commas) leave empty if none


18. Please tell us how VAAndroidRater design interface can be further improved.


19. Please write a free-formatted comments if you want to share any ideas/opinion about VAAndroidRater app.


Powered by

Google Forms
Appendix D

Advertising Poster
Research User Study on:

Mobile Apps' Privacy Scoring

We are looking for participants to take part in a study of the effectiveness of a new designed interface for educating users on mobile privacy.

As a participant in this study, you will have a mobile app installed on your Android device for two weeks, and at the end you will be interviewed for feedback of using the app. Each of the two visits will last a maximum of 60 minutes.

This study has been reviewed and given a favourable ethical opinion by the University of Surrey Ethics Committee.

Contact Saeed Alqahtani via
email: s.alqahtani@surrey.ac.uk / phone: 01483 686085

Get paid
£10 CASH
Payment rate per session

Enrolling NOW
Appendix E

Electronic Survey Design
Survey Instructions (Click to collapse)

- Thank you for taking part of this study. You are here because you have successfully completed the first HIT of this experiment.
- The purpose of this survey is to collect your subjective view on different aspects of the tested mobile app, PAitRoid. Furthermore, to help understand and analyse results of this lab study, we need to obtain your experience and background around privacy in mobile devices.
- Please be aware that we will not collect any personal information that can be linked back to you, and all private details will be kept anonymous.
- Please take your time to answer the questions below. Your response is highly appreciated as it will contribute to the improvement of PAitRoid design.

Demographic Questions

1. What is your age?
   - 18-24
   - 25-34
   - 35-44
   - 45-54
   - 55-64
   - >=65

2. What is your gender?
   - Female
   - Male
   - Prefer not to say

3. Your nationality/nationalities:

4. What is your mother language(s)?

5. What is your highest degree of level of education?
   - PhD degree or equivalent
   - Master's degree or equivalent
   - Bachelor's degree or equivalent
   - Others

Background Questions

1. Are you aware of privacy issues that mobile apps can cause?
   - Yes
   - No
   - Not sure
2. Please name some mobile app threats on privacy: 

3. Do you use any mobile privacy protection apps? 
   - Yes
   - No
   - Not sure

4. If previous question is ‘Yes’, name the protection app/apps you use: 

5. Please write free-formatted comments on how to improve privacy on mobile devices. 

**Assessment of the Tested Mobile App**

1. Do you think the tested app is useful? 
   - Yes
   - No
   - Maybe

2. Please write why you think the app is/isn’t useful. 

3. How would you assess the design of the tested app? 
   - Easy and clear
   - Needs some time to be understood
   - Confusing
   - Can be further improved

4. Do you think you gained knowledge while using the tested app? 
   - Yes
   - No
   - Not sure

5. Did you feel that you need to remove/substitute apps with low privacy scores? 
   - Yes
   - No
   - Not sure

6. Name apps that you substituted/uninstalled due to their problematic privacy scores: 

7. Name apps that you considered substituting/uninstalling them due to their low privacy rating: 

8. How relevant are the provided alternative apps? 
   - Very Irrelevant
   - Irrelevant
   - Not Sure
   - Similar
   - Very Similar

9. Give a general feedback on the provided alternative apps, please:
10. Please tell us how the app interface design can be further improved:

11. You can use this space to write any free-formatted comments if you want to share any ideas/opinion about the tested app, this survey or/and the HIT in general:

12. Are you satisfied with the total amount paid for both HITs in this experiment?

<table>
<thead>
<tr>
<th>Very Unhappy</th>
<th>Somehow Unhappy</th>
<th>Neither or Not Sure</th>
<th>Somehow Satisfied</th>
<th>Very Satisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submit
Bibliography


