



**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

# Android Watchdog

A Privacy Preserving Android Application

**Sigurd Hagen Falk**

**Fredrik Stenbro**

Master of Science in Computer Science

Submission date: June 2015

Supervisor: Anders Kofod-Petersen, IDI

Norwegian University of Science and Technology  
Department of Computer and Information Science



# Problem Description

Many of Telenor's customers are using smartphones with any number of applications installed. Most applications require the user to allow some access to different hardware (e.g. camera), personal information (e.g. contacts) or even to communicate on behalf of the user (e.g. sending SMSs). Some access is purely for local purposes (e.g. a QR-code reader obviously requires access to the camera), while some applications will transmit data to central servers. The Telenor Watchdog application is part of an ongoing project at Telenor, where we aim to help our customers manage their privacy. We are looking for an Android application that can monitor other third-party applications and tell the user which of these are a threat to his/her privacy.



# Sammendrag

Denne oppgaven utforsker problemstillinger knyttet til personvern, både generelt og spesielt med tanke på Android smarttelefoner. Tidligere forskning indikerer at mennesker ofte er irrasjonelle når det kommer til personvern. De sier de har kontroll over personlig informasjon lagret digitalt, men motbeviser seg selv gjennom sine handlinger. Android Permissions har den hensikt å gi brukere av en Android-enhet informasjon om den kritisk funksjonaliteten en tredjeparts applikasjon kan implementere ved å kreve de når applikasjonen blir installert. Det er bevist at dette er en lite effektiv måte å gi brukeren forståelse: de forstår ikke hva det innebærer å akseptere Permissions. Det er på grunn av dette vi har utviklet en modell for å rangere tredjeparts applikasjoner. Modellen er basert på vår ekspertise, en applikasjon sine påkrevde Permissions og brukerens preferanser om disse Permissions. Vi foreslår også nye skriftlige forklaringer til alle Android Permissions som informerer brukere om hva det innebærer å akseptere de. Vi har implementert disse forklaringene sammen med vår modell for rangering i en Android applikasjon. Denne applikasjonen har som hensikt å øke brukernes evne til å opprettholde sitt personvern. Vi evaluerte applikasjonen med en gruppe på 20 studenter der de fleste ikke hadde teknisk bakgrunn. Resultatene fra evalueringen indikerer at vår analyse av Permissions og modell for rangering av applikasjoner kan endre brukere sin holdning til personvern i en positiv retning. Evalueringen viser også at brukerakseptansen for en slik applikasjon er høy.



# Abstract

This study explores issues related to privacy, both in general, and especially on Android smartphones. Previous research indicates that people often are irrational when it comes to privacy. They state that they are in control of their digitally stored personal information, but their actions show the opposite. On Android devices, permissions are intended to provide users with information about the critical functionality an application can implement by requesting it on install-time. This vision have proven be ineffective: users do not understand what it entails to accept them. Motivated by these issues, we developed a model for ranking of third-party Android applications by threat to user privacy based on our expertise, required permissions, and the users' preferences on permissions. Also, we propose new descriptions for Android permissions that educate the user about its abilities. These new descriptions and our ranking model are implemented in an Android application with the purpose of increasing Android users ability to maintain their privacy. We evaluated the application with a group of 20 students, most with a non-technical background, using it over a short period. Our evaluations show that the analysis of permissions and implementation of our privacy risk score model in an Android application can change users' attitude towards privacy in a positive direction. We also find the user acceptance of such an application to be high.





# Preface


This thesis is submitted to the Norwegian University of Science and Technology (NTNU) as our final requirement for the degree of Master of Science in Computer Science. The project was carried out during the spring of 2015, at the Artificial Intelligence Group of the Department of Computer and Information Science, Faculty of Information Technology, Mathematics and Electrical Engineering, NTNU. Our supervisor has been Adjunct Professor Anders Kofod-Petersen. The project has, together with NTNU, been supported by Telenor Research.

## Acknowledgements

We wish to thank Anders for motivation, guidance, and creative ideas throughout the period of this thesis. Gratitude is also given to Wilhelm Walberg Schive which has been a close collaborator. His participation in discussions have been helpful. We would also like to thank our fellow students in Room 363 - Korp at "IT-Vest" for a creative and inspirational atmosphere and working environment. Finally, acknowledgements are given to the faculty for providing us with equipment and an office space.



Sigurd Hagen Falk



Fredrik Stenbro

Trondheim, June 8, 2015



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background and Motivation . . . . .	1
1.2	Goal and Research Questions . . . . .	2
1.3	Research Method . . . . .	2
1.4	Contributions . . . . .	3
1.5	Thesis Structure . . . . .	4
<b>2</b>	<b>Background Theory and Motivation</b>	<b>5</b>
2.1	Background Theory . . . . .	5
2.1.1	Defining Privacy . . . . .	5
2.1.2	The Principle of Minimal Asymmetry . . . . .	6
2.1.3	Approximate Information Flow . . . . .	6
2.1.4	Human-Computer Trust . . . . .	8
2.1.5	Evaluating User Acceptance of Privacy-Aware Applications	10
2.1.6	Introducing Android . . . . .	12
2.1.7	Explanation-Aware Computing . . . . .	14
2.2	Literature Review . . . . .	15
2.2.1	Structured Literature Review Protocol . . . . .	16
2.2.2	Ad-Hoc Literature Search . . . . .	19
2.3	Motivation . . . . .	20
2.3.1	Introduction . . . . .	20
2.3.2	User Relationship to Privacy . . . . .	20
2.3.3	Users do not Understand Permissions . . . . .	24
2.3.4	Users Value Utility over Privacy . . . . .	26
2.3.5	There is no such thing as a free application . . . . .	27
2.3.6	Explanations are needed for users to trust applications . . .	28
2.3.7	There is a lack of good Android privacy applications . . . .	29
2.3.8	Related Work . . . . .	30

<b>3</b>	<b>Experiments and Results</b>	<b>39</b>
3.1	Detection of Third-Party Application Behaviour . . . . .	39
3.2	Possible Malicious Use of Android Sensors . . . . .	43
3.3	Comprehensive Descriptions of Android Permissions . . . . .	46
3.4	Android Permission Patterns . . . . .	48
3.5	Privacy Risk Score . . . . .	53
3.6	User Behaviour Analysis . . . . .	58
<b>4</b>	<b>Android Watchdog Application</b>	<b>63</b>
4.1	Our Vision . . . . .	63
4.2	Graphical User Interface . . . . .	64
4.3	Explaining Application Behaviour . . . . .	65
4.4	Implementing our Research in UtilityComponents . . . . .	65
4.4.1	BehaviourAnalysis . . . . .	65
4.4.2	SystemMonitor . . . . .	67
4.4.3	PrivacyScore . . . . .	67
4.4.4	UserPreferences . . . . .	67
4.4.5	NotificationManager . . . . .	68
4.5	Application Activities . . . . .	68
4.5.1	Main . . . . .	68
4.5.2	ApplicationList . . . . .	71
4.5.3	ApplicationDetail . . . . .	71
<b>5</b>	<b>Application Evaluation</b>	<b>77</b>
5.1	Evaluation Plan . . . . .	77
5.2	Evaluation Hypothesis . . . . .	78
5.2.1	Privacy Survey Hypothesis . . . . .	78
5.2.2	Application Usage Hypothesis . . . . .	78
5.2.3	Mobile Services Acceptance Model Hypothesis . . . . .	78
5.3	Evaluation Setup . . . . .	79
5.4	Privacy Survey Results . . . . .	80
5.5	Application Usage Results . . . . .	84
5.6	Mobile Services Acceptance Model Results . . . . .	85
5.6.1	Descriptive Results . . . . .	85
5.6.2	Data Analysis . . . . .	86
5.7	Evaluation Limitations . . . . .	89
<b>6</b>	<b>Conclusion and Future Work</b>	<b>91</b>
6.1	Discussion . . . . .	91
6.1.1	Research Approach . . . . .	91
6.1.2	Android Watchdog Application . . . . .	93
6.2	Answers to Research Questions . . . . .	95

---

6.3	Thesis Contributions . . . . .	96
6.4	Final Remarks . . . . .	97
6.5	Future Work . . . . .	97
6.5.1	Centralised Backend for Calculation of User Preferences . . . . .	98
6.5.2	Detection of "Proxy Applications" . . . . .	98
6.5.3	System Signed Application to Access Root Privileges . . . . .	98
6.5.4	Permission Threat Level Based on Application Category . . . . .	99
6.5.5	Privacy Risk Score Evaluation . . . . .	99
6.5.6	Watchdog Recommender . . . . .	99
6.5.7	Multiparty Differential Privacy for Permission Weights . . . . .	99
<b>Bibliography</b>		<b>101</b>
<b>A Permission Descriptions and Threat Levels</b>		<b>107</b>
<b>B Mobile Services Acceptance Model Questionnaire</b>		<b>119</b>
<b>C Privacy Survey Questionnaire</b>		<b>123</b>
<b>D Privacy Survey Questionnaire Results</b>		<b>127</b>



# List of Figures

2.1	An Illustration of Asymmetric Information Flow . . . . .	7
2.2	A Design Space for Privacy Solutions in Ubiquitous Computing . .	9
2.3	Model of Human-Computer Trust Components . . . . .	10
2.4	The Technology Acceptance Model . . . . .	11
2.5	The Mobile Services Acceptance Model . . . . .	12
2.6	User Interface for Presentation of Privacy Leaks . . . . .	22
2.7	Android Permissions Presented to the User at Install-time . . . . .	25
2.8	Improved Android Privacy Summary Interface . . . . .	34
3.1	An illustration of an Android View drawn on top of the Android Home Screen . . . . .	45
3.2	The Procedure for Security Requirements Identification . . . . .	50
3.3	The Sigmoid Function Curve . . . . .	56
3.4	An Illustration of how Applications in the Test Set fit our Privacy Risk Score . . . . .	58
4.1	Overview of The Android Watchdog Architecture . . . . .	64
4.2	Illustration of explanations in the ApplicationList-activity and the ApplicationDetail-activity. . . . .	66
4.3	Illustration of the applications Main-activity. . . . .	69
4.4	Illustration of how disharmonious applications are presented in the application. . . . .	70
4.5	Illustration of the ApplicationList-activity. . . . .	72
4.6	Illustration of the ApplicationDetail-activity. . . . .	73
4.7	Illustration of the activity presenting our Permission Descriptions .	74
4.8	Illustration of how application updates are summarised and how status bare notifications are shown. . . . .	75
4.9	Illustration of the activity presenting our Permission Patterns . . .	76
5.1	Grade of how Concerned Subjects are with their Privacy . . . . .	81

5.2	Grade of to what extent Subjects understand what Information Third-Party Applications on their Android device collect . . . . .	82
5.3	Grade of Trust Subjects has to Social Media Applications. . . . .	83
5.4	PLS Analysis of our Mobile Services Acceptance Model Question- naire . . . . .	88
6.1	Possible Framework for Explanation of Android Permissions . . . . .	93



# List of Tables

- 2.1 Literature Review Search Terms . . . . . 17
- 2.2 Most Used Permissions Among Over-privileged Applications . . . . . 33
- 2.3 Top 20 Used Permissions by Malicious Applications . . . . . 37
  
- 5.1 Gender, Age, and Department Distributions over Evaluation Subjects . . . . . 80
- 5.2 Number of Views per Activity collected from Google Analytics . . . . . 84
- 5.3 Average Response Value for Questions in our Mobile Services Acceptance Model Questionnaire . . . . . 86
- 5.4 Cronbachs Alpha Value for each Construct in our Mobile Services Acceptance Model Analysis . . . . . 87
- 5.5 Key Values for the PLS Analysis of our Mobile Services Acceptance Model Questionnaire . . . . . 89



# Chapter 1

## Introduction

*This chapter will introduce our motivation, goal and research questions. Further, it will give insight on how research is conducted, sum up contributions, and explain the structure of this thesis.*

### 1.1 Background and Motivation

Privacy is an increasingly frequent topic in society and media. It is also partly a twofold issue because when asked, people tend to seem surprisingly concerned about privacy. However, research indicates the opposite - that most people do not care about privacy, or lack the knowledge and information to preserve it [4]. As technology advances, most of the privacy-related issues today regards the Internet and devices connected to it.

Going back five to ten years, being connected meant being able to call or send a text message. In 2015, it corresponds to having hundreds of communication and entertainment services available at all time. We can not argue about the value these services provide in the form of connecting people and making it simpler to share information. However, these services come with a cost. Since many of them are free, they survive by collecting and selling information about users [57, 29]. Combined with incomprehensible terms and conditions, this makes it challenging for users to understand how the use of new services affect their privacy [20].

Today, most people own a smartphone, and the majority of these run with the Android operating system [30]. Many of the current privacy-related issues is directly caused by the increasing popularity of smartphones, which has introduced simple ways to collect large amounts of personal information about users. In Android applications, this wide range of information is only restricted by a permission system that has become less effective than originally intended

[33, 15]. This easy access to personal information has also created opportunities for developers with malicious purposes.

Lately, privacy related to smartphones has been the topic of many articles published in different medias [29, 40, 9, 56, 43, 54]. In the wake of this increased attention, several applications have been proved to disclose and misuse personal information. There is definitely a need to educate users on how they can participate in preserving their privacy on Android devices. This motivates the idea of creating an application that helps users manage their privacy, educate them, and inform them about potential threats.

## 1.2 Goal and Research Questions

The following goal and research questions forms the basis of this thesis and guides further research and experiments.

### 1.2.1 Goal

- Create an application that increase users' ability to maintain their privacy on an unrooted Android device by informing about actual and possible threats for disclosure of personal information.

### 1.2.2 Research Questions

1. Which techniques can be used to detect possible malicious behaviour of third-party applications based on real-time system monitoring and application analysis on an unrooted Android device?
2. What is the best way to inform users about threats in installed third-party applications on an Android device and provide them with incentives to uninstall these applications?
3. Which user interaction patterns can be employed to make users aware of their privacy-related behaviour?

## 1.3 Research Method

Our initial approach will be theoretic to form a foundation of knowledge about the problem domain and to address goals and underlying research questions stated in the previous section. We will then apply this in a practical/experimental phase for research and development of the application. Finally, we will conduct an

evaluation of our solution to confirm and/or identify limitations and weaknesses in our approach.

In our research, our motivation was to find the necessary literature to understand the principles of privacy in general, users' relationship to privacy, and privacy in smartphones with the Android operating system. This information was gathered from different curriculum among courses at NTNU and from a literature search targeting our goals and research questions. It provided us with background information as well as a better foundation for further research and motivation for our project. It made us better fit to perform the experimental phase described next.

Our experimental phase is where we find answers to our research questions listed in the previous section. We have performed multiple experiments linked to one or more of these questions. The specific methodology for each experiment is thoroughly explained in their respected section in Chapter 3.

The evaluation process was conducted to measure the usefulness of our solution. This phase seeks to reveal strengths and weaknesses in our approach, and how it impacts the user. We applied two evaluations to reveal both changes in users' attitude towards privacy and users' acceptance of our application. The results and analysis of the evaluation further forms the basis for our arguments towards the achievement of the thesis' main goal.

## 1.4 Contributions

Several contributions are made in this report. Our literature study combines research on how users relate to privacy and what privacy issues are connected to the Android Operating System. We have found weaknesses in the Android API, making it possible to use Android device sensors in a malicious manner. Further, we found methods to detect such misuse of sensors, however, these methods are probable best fit for research purposes because they can interrupt other third-party applications. Next, we have developed a privacy ranking system for third-party Android applications. This system is based on required permissions, identifying patterns among these permissions to detect possible privacy violating behaviour, and the user's opinion about them. We have also used different kinds of explanations in our application to strengthen its trustworthiness. Finally, we have evaluated the user acceptance of our application and also found what impact such an application has on users' attitude towards privacy.

## 1.5 Thesis Structure

The thesis is structured as follows. The next chapter will present background theory, motivation and related work. This defines several important elements for our problem domain. It will also include our literature review, motivation, and related work. After this, Chapter 3 outlines our experimental phase, which includes plans, setups and results. This is followed by Chapter 4 which presents our application with an elaboration of its architecture, components, design and functionality. Then, Chapter 5 show the evaluation of our application with results and analysis of results. Finally, Chapter 6 sum up our work, discuss our approach and relate this to our main goal. A brief summary of our contributions and future work is also presented.

## Chapter 2

# Background Theory and Motivation

*This chapter will present background information related to our problem domain. It also outlines our literature search, including a Structured Literature Review Protocol. Finally, we motivate our goal based on findings in the literature search and also identify relevant research.*

### 2.1 Background Theory

This section will introduce important definitions and background theory that has played key roles in the further development of this thesis.

#### 2.1.1 Defining Privacy

There is no widely accepted definition of privacy. Oxford Dictionaries defines the noun privacy as "A state in which one is not observed or disturbed by other people". This describes an absolute state, but with the rapid evolution of technology and techniques of surveillance there is a need to define privacy in a matter that can be quantified. Wikipedia states "Privacy is the ability of an individual or group to seclude themselves, or information about themselves, and thereby express themselves selectively". This is probably closer to what people think of privacy today. The rise of social media and the market of selling personal information presents a need for users to select what they prefer to share. It also presents a need for users to understand the consequences of sharing information. If a third party are buying personal information with the purpose of earning money from it, the person the information belongs to should know about

this. A more precise definition was given by Noam [44]: "Privacy is an interaction, in which the information rights of different parties collide. The issue is of control over information flow by parties that have different preferences over 'information permeability'". When considering privacy today, the ability to store massive amounts of information digitally has to be taken into account. There must be some interaction between the owner of the information and at the other end, gatherers and users of the information. Defining privacy this way will help highlight the need to minimise the asymmetry in information flow.

### 2.1.2 The Principle of Minimal Asymmetry

Environments with asymmetric information describe situations in which some actors hold private information that is relevant to everyone [31]. To further relate this to mobile privacy, we will use an example where an application is tracking a user's position using the GPS-sensor in his/her smartphone (Figure 2.1 visualises the example). When downloading this application users are told it needs permission to access position information to show the current position of the smartphone on a map. However, every time it is launched, the application calculates the current position and then sends this data to a centralised server owned by the company who developed the application. This information is then sold to a third party that specialises in targeted marketing. In this scenario, the user (data owner) has no knowledge of the fact that his/her position data acquired by the application (data collector) is being sold to a third party (data user). The asymmetry appears when the data collector and the data user knows more than the data owner about how the data collected is going to be used. Jiang et al. [31] developed The Principle of Minimal Asymmetry, which aims to minimise the asymmetry between data owners on one side and data collectors and data users on the other. A privacy-aware system should reduce the asymmetry of information between data owners, and data collectors and data users by:

- **Decreasing** the flow of information from data owners to data collectors and users
- **Increasing** the flow of information from data collectors and users back to data owners

### 2.1.3 Approximate Information Flow

Approximate Information Flow (AIF) is a novel model developed by Jiang et al. [31] for privacy-aware ubiquitous computing architectures that embody the Principle of Minimum Asymmetry. AIF is not meant to enforce privacy, but rather



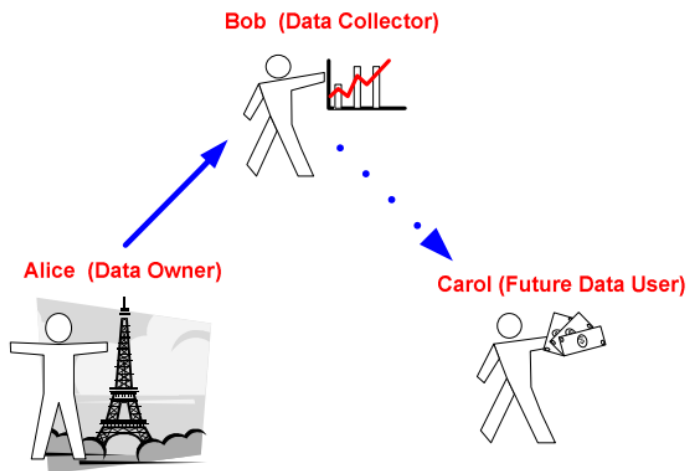


Figure 2.1: An illustration of asymmetric information flow: Alice sends her data to Bob, but are not informed that Bob resell this data to Carol for second-hand use.

provide a way to describe information flow within a system of people and computers. It is based on three abstractions - information flow, data lifecycle and themes for minimising asymmetry. The last two abstractions are used to build a design space for categorising privacy protection mechanisms (described later in this section).

Information spaces are where data is being stored, either by the information owner, collector or user. These repositories have three important properties regarding privacy:

- **Persistence of data:** Data stored in an information space can have different lifetime and the quality of the data during the lifetime could be degraded. For example, information about an online purchase may only exist until the merchandise is delivered.
- **Accuracy of data:** Data can have different levels of accuracy. For example when tracking location, how often position data are gathered define how accurate the position trace is.
- **Confidence of data:** Measures the uncertainty of the data contained in an information space. Continuing the example of location tracking, there would be more confidence in positions retrieved from a GPS-sensor compared to triangulating.

Further, information spaces do not need to be bound to a specific location, device or person. They can be delimited by three different boundaries. **Physical boundaries** separate information spaces by their location. **Social boundaries** separate them through social groups, for example, a project team. Last, **activity-based boundaries** serve to distinguish information spaces by what the user are doing.

Data contained in an information space can be altered by a set of operations. Addition/Deletion/Update provide standard database operations. Ownership and release policies are handled by authorisation and revocation. Persistence, accuracy and confidence of data can be taken care of by promotion and demotion. Composition and decomposition combine data from different sources or split data into different pieces. Finally, fusion and inference aim to gain higher-level information based on raw data.

The second abstraction is data lifecycle. It defines the different stages data are undergoing from when it is gathered to the initial access of collected information to the possible use of it from third parties. To be precise, collection refers to the point at which data is gathered. Here, properties like accuracy, persistence and confidence should be emphasised. Access is when gathered data is initially being accessed. What data should be available, by whom it should be available for and for what purpose are questions that need to be answered. Second-hand use are sharing of collected data after the initial access. It should define which parties data can be shared with and what they can do with it.

Themes for minimising asymmetry is the last abstraction. It provides a way for categorising privacy protection mechanisms into three themes. Prevention may reduce the accuracy, confidence or persistence of data to prevent it from being used in an undesirable manner. It can also eliminate privacy risky operations to achieve this. Avoidance seeks to inform users of information systems about risks and benefits of sharing information. Detection detects illegal use of data and supplies ways of holding actors doing so accountable.

The last two abstractions, data lifecycle and themes for minimising asymmetry, are combined to provide a design space for categorisation of privacy protection mechanisms for information flow between information spaces (see Figure 2.2). Here techniques for enhancing privacy can be placed in the relevant stage of the data lifecycle and to what extent it ensures privacy in detection, avoidance or prevention.

#### 2.1.4 Human-Computer Trust

Trust has shown to be a crucial element in keeping users of new technology motivated and cooperative. If the user is not able to understand actions performed by a system, the trustworthiness of that system will decrease [42]. A consequence

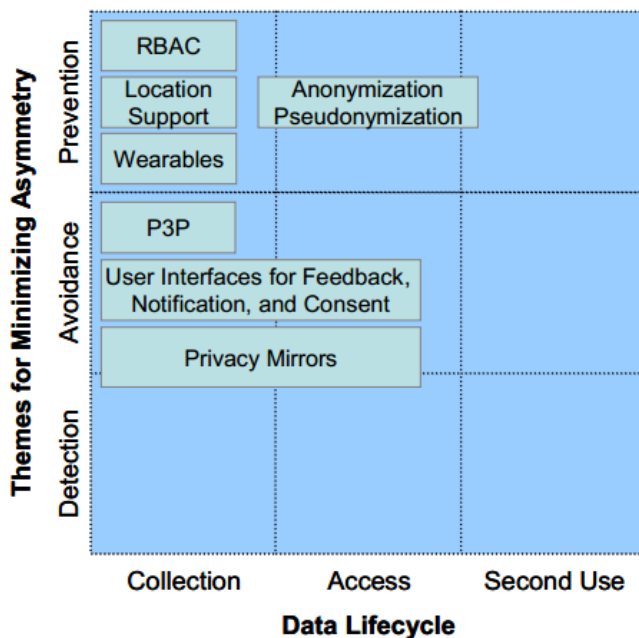


Figure 2.2: A design space for privacy solutions in ubiquitous computing proposed by Jiang et al. [31]. The figure illustrates how different privacy-preserving approaches cover each step in the data lifecycle.

of this is reduced willingness among users to interact with the system, and in the worst case to an abort in use [45].

Mayer et al. [38] defines three levels that build the base of trust in human relationships - ability, integrity and benevolence. Each level can be seen as one, but are related to the others. To gain a high level of trust, all of these levels must also be perceived high. For a human to trust a computer system, similar bases should be taken into consideration. Figure 2.3 presents a human-computer trust model developed by Madsen and Gregor [37]. In their model, personal attachment and faith build the bases for affect-based trust and perceived understandability, perceived technical competence, and perceived reliability for cognition-based trust. This model is based on a definition adapted from McAllister [39] and incorporates both the users' confidence in the system and their willingness to act on the systems decisions and advice:

"The extent to which a user is confident in, and willing to act on the basis of, the recommendations, actions, and decisions of an artificially

intelligent decision aid.”

The distinction between the users’ confidence in a system and the willingness to use it to perform the a decision task. Confidence may be seen as the primary outcome from cognition-based trust (the users’ intellectual perceptions of the system’s characteristics) and willingness may be regarded as an outcome of both cognition-based and affect-based trust (the users emotional responses to the system).

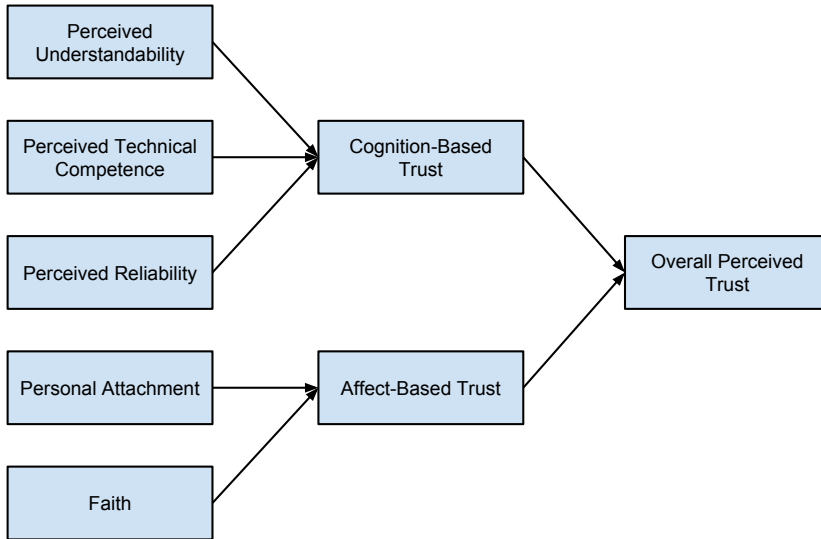


Figure 2.3: Model of Human-Computer Trust Components proposed by Madsen and Gregor [37]. To gain high overall perceived trust, both cognition-based trust and affect-based trust must be perceived as high.

To validate their model, they conducted a field study that resulted in 78 complete surveys. Their results prove the model to be both a reliable and valid measure of human-computer trust.

### 2.1.5 Evaluating User Acceptance of Privacy-Aware Applications

A measure for speeding up development and adoption of new technologies is for researchers to evaluate their work based on standard evaluation frameworks. User satisfaction needs to be an essential part of such frameworks. Standard frameworks for evaluation represents a way to easily reuse research and ideas. Several

models have been developed to test the users' attitude towards and intention to adopt new technologies or information systems. Among these different models that have been proposed, the Technology Acceptance Model (TAM) [12], appears to be the one most widely adopted [17].

TAM is an information systems theory that models how users come to accept and use new technology. The model (see Figure 2.4) suggests that when users are presented with a new technology, some factors influence their decision about how and when they will use it, notably:

- **Perceived usefulness (PU):** The degree to which a person believes that using a particular system would enhance his or her job performance.
- **Perceived ease-of-use (PEOU):** The degree to which a person believes that using a particular system would be free from effort.

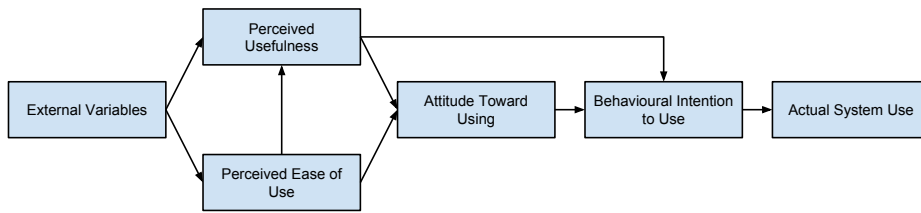


Figure 2.4: The Technology Acceptance Model (TAM) developed by Davis [12]. TAM considers both perceived usefulness and perceived ease of use to be important for the intention to use new technology.

Although studies of TAM provides empirical evidence on the relationships that exist between usefulness, ease of use and system use [2, 28], critics of the model has pointed out TAMs limitations relative to extensibility [5] and explanation power and that TAM needs to be extended with additional variables to provide a stronger model [35]. Gao et al. [17] recognises this and proposes an extension of TAM suitable for evaluating modern mobile services (see Figure 2.5). They propose an extension of TAM including context, personal initiatives and characteristics, and trust.

The research performed by Gao et al. [17] was designed to study mobile information services' adoption from university students perspective. However, their extension of TAM can also be suitable for evaluating privacy-aware applications for the general public. Their findings from a survey of 46 students indicate that personal initiatives and characteristics, trust, perceived usefulness and perceived ease of use are key determinants on adoption of mobile information systems.

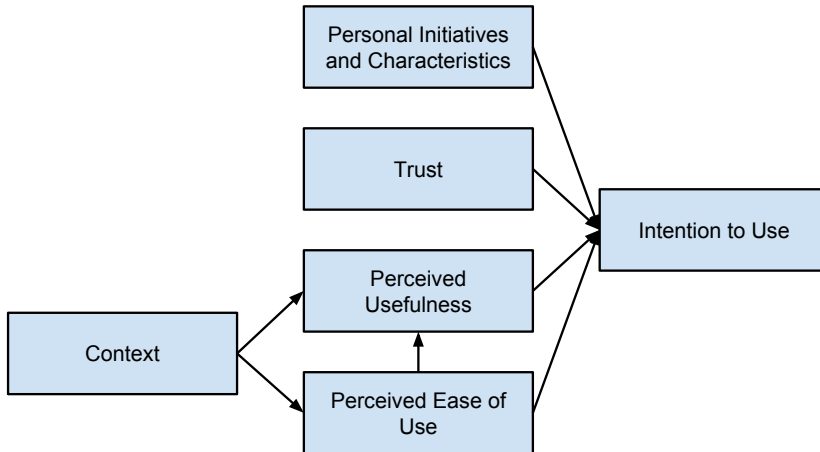


Figure 2.5: An illustration of the Mobile Services Acceptance Model proposed by Gao et al. [17]. Their model is an extension of TAM with two new evaluation criteria: personal initiatives and characteristics, and trust. They have evaluated this extension of TAM to be a better fit for acceptance of mobile services.

### 2.1.6 Introducing Android

This section will provide a short introduction of the Android operating system, and some of its key elements.

#### Android Operating System

The Android operating system was primarily developed for mobile devices with touchscreens by Android Inc. It was sold to Google in 2005 and in 2008, the first phone with Android was sold. Since then, its popularity has increased, and in the first quarter of 2014, Android had a market share of around 80 % globally [30]. Android is an open source operating system that allows everyone to develop applications. Due to the massive user base and the increasing developer community Google created Google Play Store. Google Play Store is a system for distributing and selling applications. This enables developers to earn money by selling their applications, and many companies survive solely on this business platform. In July 2013, Google announced that there were over one million applications published on Google Play Store and over 50 billion downloads. The Android system is based on an ARM-architecture and runs with a Linux kernel. The system library is written in C and framework applications in Java. Applications for Android are developed using the Android Software Development Kit

(SDK).

### **Android Permission System**

The Android permission framework is intended to serve two purposes in protecting users: (1) to limit mobile applications' access to sensitive resources, and (2) to assist users in making decisions about installing applications. All applications on Android run in an application sandbox. By default, an Android application can only access a limited range of system resources. Protected APIs to sensitive information are protected through permissions. These protected APIs include:

- Camera functions
- Location data (GPS)
- Telephony functions
- SMS/MMS functions
- Network/data connections

To access these protected APIs, developers has to declare which of the permissions are intended to be used in their application. Failure to declare a permission will lead to denied access from the protected APIs. When installing an application users are presented with all permissions required by the application. They must agree to all of them to proceed the installation, hence single permissions can not be denied. Once they are granted, permissions are valid for the application as long as it is installed. There are currently around 100 different permissions defined in the Android API [22].

The second intention for the permission framework is about how users understand third-party applications. Permissions are also for users to evaluate an application's functionality together with general information about the application and the developer to decide if it meets their needs and expectations. The intention behind having users to accept all permissions before installing is to enhance user experience and having users switching seamlessly between applications at will. The main argument behind not having users accepting permissions when they are used is that over-prompting causes the users to reply "OK" to any dialogue that is shown [23].

### **Intents, Services, and Broadcast Receivers**

A Intent is a messages that can be passed to another application component to request access to its actions. Intents can start three types of components; an activity, a service or deliver a broadcast. Further, there are two types of intents;

explicit intents and implicit intents. The explicit intents are mostly used to start components in the same application, often in response to a user action or to start a service. The implicit intent is used to access components in other applications, such as the system camera or location service [25].

A service is a component within an application that can perform background operations, not just when the application is running in the foreground, but also at a specified time or interval when the associated application is not running. There are two types of services: started and bounded. The started service can provide a background process to run indefinitely, independently of applications, and it usually performs a single operation, often network related, without the return of an explicit result to the owning application. The bound service, on the other hand, is more coupled with the application, as it only lives as long as an application component is bound to it. It runs while the application is in the foreground and lets components interact with it, sending requests and extracting results [27].

Broadcast receivers receive intents and performs customised actions accordingly, and are often used to initiate services. Receivers can be registered to answer to both external and internal intents, and is also used to receive systems events, such as confirmation on system being booted up completely up or has low battery [24].

### 2.1.7 Explanation-Aware Computing

Explanations are defined as a combination of description and comprehension, with the purpose of exposing something in a way that makes it understandable and satisfactory [34]. In other words, it answers questions that starts with "how", "why" and "what". The famous cognitive physiologist and computer scientist Roger Schank presented the following statement about explanations:

*"Explanations are considered the most common method used by humans to support decisions."* [51]

Creating computer systems that have the ability to explain their reasoning and inference process contains highly valuable treats for the human-computer relationship. In relation to Schanks statement, explanations can help in making privacy-aware computing easier, both in the form of building trust with the user, but also helping the user in decision-making. Explaining can help establishing a trustworthy human-computer relationship by providing a transparent reflection over both system knowledge, prerequisites and all relevant factors leading up to a result. Another favourable aspect of explaining is achieved through learning. Interacting with a system that presents terms and concepts in an application



domain where a potential user might have little or no experience creates challenges. Explanations can provide valuable education, and help users interact with unfamiliar systems.

Cassens and Kofod-Petersen [7] describes different types of explanations listed below:

- **transparency explanations:** A transparency explanation serves the purpose of providing a full understanding of a systems intention. This includes a complete overview of every element that influences system processes. The user should be able to examine both data and the assumption made by the system.
- **justification explanations:** This form of explanation provides confidence in the answer offered by the system. Increasing confidence usually comes in the form of supporting the conclusion with additional and simplified knowledge and information about the inference process.
- **conceptual explanations:** This type of explanation seeks to provide understanding of concepts and terms used by the system or in the related discipline. A conceptual explanation is critical to ensure full comprehension of the system for novice users in complex domains.
- **learning explanations:** A learning explanation can provide valuable education in application domains. This kind of explanation is often equally interested in explaining the reasoning and not just the answer. Learning explanations also tend to make use of dialogues with the user along the learning process.

## 2.2 Literature Review

This section addresses the literature review and describes stepwise the approach taken to find research relevant to our research questions. The scope of the literature search is in both the domain of privacy and smart systems that run the with Android operating system. The search is therefore conducted in a manner where both are investigated separately, but with focus on intersecting areas. We created a Structured Literature Review Protocol (SLR) to make our literature review reproducible and to guarantee a thorough method for unveiling existing research and solutions relevant to the application we aim to develop. To find more general research on our other research questions, we performed an ad-hoc search. This resulted in relevant literature from the chosen search engines, our courses at NTNU, and recommendations from people with knowledge in our research field.

## 2.2.1 Structured Literature Review Protocol

### Defining the Problem Domain

**Problem (P):** How to perform automated privacy analysis and measurement of installed third-party applications on an Android device, with a focus on comprehension, transparency, and trust.

**Method (C):** Perform analysis of installed third-party applications on a unrooted Android device.

**System (S):** An Android application.

### Research Questions (RQ)

1. What are the existing solutions to P?
2. How do the different solutions found by addressing RQ1 compare to each other with respect to C?
3. What is the strength of the evidence in support of the different solutions?
4. What implications will these findings have when creating S?

### Identification of Research

The literature search will be performed on the following sources:

- ACM Digital Library
- IEEE Xplore Digital Library
- SpringerLink
- CiteSeerX
- ScienceDirect

These are archives for computer science. Because our problem domain do not extend much beyond this field, we found them sufficient for our literature review.

Based on our research questions, we have defined several search terms. These terms reflect what we are searching for in the above libraries. To eliminate the possibility of not retrieving relevant literature because authors use different words in equal context, we have also defined synonyms to these terms (see Table 2.1) so that we can combine them when forming search strings.

Group 1	Group 2	Group 3	Group 4
Android	Smart Phone		
Application	Program	System	Tool
Permission	Privilege		
Third party			
Privacy	Security	Sensitive information	Personal information
Automated	Realtime	Runtime	
Evaluate	Analyze	Grade	Classify
Explanation			

Table 2.1: The table presents terms used in our literature search. All terms in every group were combined in our search. Using these terms increases the possibility for us to find all relevant research.

### Conducting the Review

This section describes the three steps that will be performed to find relevant studies.

#### Step 1: Selection of primary studies

The number of studies uncovered by the main search will probably be way to high. Because different search engines are being used, there are probably also a high degree of redundancy in this result. To reduce the number, the following elimination criteria for papers in the main search will be applied:

1. Duplicates
2. The same study published in different sources
3. Title indicating no relevance to our problem

#### Step 2: Study quality assessment

After removing studies by the method described in Step 1, the resulting papers will be suspect to the first quality assessment. In this step, we first define primary and secondary inclusion criteria. Then, we define quality screening criteria. Finally, we describe how papers must fulfil these criteria through three steps performing different analysis to not be filtered out from our review.

##### Primary inclusion criteria:

1. The main concerns of the study is on one of the defined research questions.
2. The study is a primary study presenting empirical results.

**Secondary inclusion criteria:**

1. The study focuses on methods or/and approaches.
2. The study describes a solution.

**Quality screening criteria:**

1. There is a clear statement of the aim of the research.
2. The study is put into the context of other studies and research.

**Stages:**

1. **Abstract inclusion criteria screening:** In this stage abstracts are evaluated based on the primary inclusion criteria. If it does not meet these criteria, it will be excluded.
2. **Full text inclusion criteria screening:** If the abstract answered to the primary inclusion criteria, we would further investigate its content. This stage is to analyse the study as a whole, applying both primary and secondary inclusion criteria. At this stage, studies can be evaluated as relevant even if it do not describe a solution if the method/approach presented prove to be well documented.
3. **Full text quality screening:** If the study was not excluded in the previous stage, it would be suspect for a rough quality screening. Here, the previous define quality screening criteria are used to exclude papers with low quality.

**Step 3: Detailed study quality assessment**

The studies that have gone through Step 2 are in this step subject to a more detailed quality assessment. Here a larger set of quality criteria are defined to ensure the worthiness of studies to be classified as relevant. These criteria are the following:

1. Is there is a clear statement of the aim of the research?
2. Is the study is put into the context of other studies and research?
3. Is the approach thoroughly justified?
4. Is the approach reproducible?
5. Is the approach thoroughly explained?
6. Is it clearly stated in the study that other approaches the study's approach has been compared with?
7. Are the results provided by the approach analysed and evaluated?
8. Is there any evidence presented?

## Literature Review Evaluation

### Step 1: Selection of primary studies

In our main search, we applied searches with the defined search terms in the respective literature libraries. This resulted in a combined set of around 200 papers. After applying the elimination criteria, we reduced this number to around 150 papers.

### Step 2: Study quality assessment

We applied the three stages in quality assessment on our primary studies. Many papers could be eliminated by reading the abstract. When analysing papers in greater detail, we often found relevant studies not revealed in our primary study. We also introduced papers acquired from some of our courses at NTNU. These papers were included in our literature, and we performed an equal quality assessment on them. The result of our first quality study resulted in a set of around 30 relevant papers.

### Step 3: Detailed study quality assessment

The literature still found relevant in this step is further described and discussed in the following motivation. The complete list of relevant literature is presented through the bibliography.

## 2.2.2 Ad-Hoc Literature Search

We also found useful papers and research by examining references and citations from papers uncovered in the literature review. In addition to the SLR, we did searches to find information about topics that we did not manage to cover otherwise. This was mainly regarding background information, outlying topics, or specific methods of interest. The curriculum of certain courses at NTNU also provided us with useful information. Some of our papers were also discovered by recommendation of individuals outside the project team. The following elements are research questions developed with the purpose of providing information these areas:

1. What is privacy and how do users relate to it?
2. How does the Android operating system maintain user privacy?
3. Which techniques and methods can be used to ensure privacy in Android applications?

The ad-hoc search was conducted on the same search engines as the SLR. The search results were examined from the top, based on their relevance provided by the search engine. Further, we examined around 40 results on each

research question. The complete list of relevant literature is presented through the bibliography.

## 2.3 Motivation

This section presents important findings in our literature search. First, we will motivate our goal by highlighting privacy-issues connected to use of smartphones. After this, a outline of research relevant to our research questions are presented.

### 2.3.1 Introduction

The concept of privacy is based on the ability an individual has to control the amount of personal information that is shared with others. The large amount of Android applications available has caused users to loose oversight of what information that is being collected and distributed. They do not know what information they share, when they share it, and to whom they share it with. There is a need for an application that can provide this overview. An application that can examine, monitor, and analyse third-party applications installed on an Android device. This would help the user regain control of their privacy. Closely related to the goal of reinforcing privacy is the desire to develop an explanation-aware application that provides understanding of the privacy domain, transparency and trust, and strengthen the users' ability to make privacy-related decisions.

There is much work done within the field of reinforcing privacy on Android devices. Most of these are based on external tools and analysis techniques not fit for our problem. Few studies have focused on creating an application that will, solely based on client-side functionality, perform the envisioned goals of this project. However, the topics of concern are highly similar. Disregarding malicious software, the three main privacy issues identified by our literature search are:

- Over-privilege of permissions requested by Android applications.
- Lack of transparency leads to asymmetric relationships.
- Misuse and non-consensual disclosure of information.

### 2.3.2 User Relationship to Privacy

To simplify how we define users awareness of privacy, it will be helpful to have a general characteristic to describe and quantify the level of awareness and understanding. Cranor et al. [10] did a research on users' attitudes towards online privacy where they analysed over 500 surveys. In their survey, several privacy issues were raised. Among them, how people would respond to situations where

personal information is collected. When analysing responses they defined three different groups users would belong to:

- **Privacy Fundamentalists (17%)**: Extremely concerned about data sharing even when protection mechanisms designed to protect privacy was present.
- **Pragmatic Majority (56%)**: Also concerned about data sharing, but were willing to share if they found privacy protection mechanisms satisfying.
- **Marginally Concerned (27%)**: Would share their data in almost any circumstances. They express a mild general concern about privacy.

This general classification of users can be used to design better applications in the future. Developers have to produce applications with the ability to convince users of their intentions. People are privacy fundamentalists for a reason. They do not trust the companies and developers to sufficiently maintain their shared information. On the other side, marginally concerned users need to be educated about how they can maintain their privacy. The general goal should be to move users from these two groups into the pragmatic majority by better implementation and explanation of privacy mechanisms.

Balebako et al. [4] performed a lab study on awareness of data leaks on smartphones. Data leaks are defined as the transmission of personal data without the user knowing about it. They exposed 19 participants to a role-playing scenario, letting them play two popular Android games two times. First, the game was played on a regular Android phone, and secondly on a modified version of Android able to detect data leaks and present them to the users through a simple interface (see Figure 2.6). Finally, they interviewed participants about their experience and general concern about data sharing. They categorised participants in three groups based on their results.

1. **Five participants** had never before thought about information leaving the phone.
2. **Eight participants** believed that data was shared only with application developers for the purpose of improving the application.
3. **Six participants** understood that data was used for marketing but were surprised by the scope of data sharing, including the frequency of data sharing and the destination of data.

These findings indicate that people do not understand how and to what extent their personal data are being used by smartphone applications. When users do not understand the behaviour of an application, it will affect the cognition-based

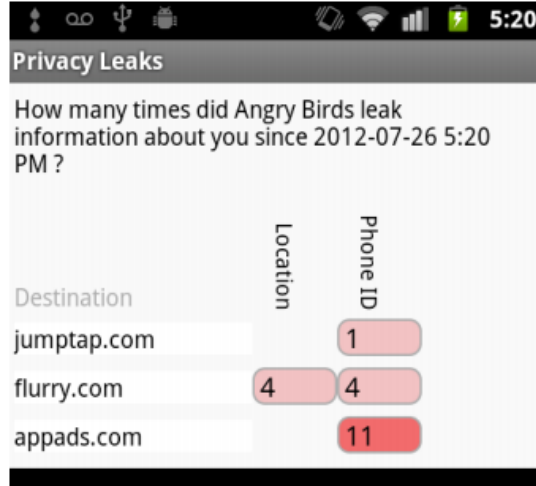


Figure 2.6: Detail screen of privacy leaks used by Balebako et al. [4] in their research. The interface shows how many times location and phone ID were leaked by third-party applications installed on an Android device.

trust. The fact that some users were surprised about the amount of data that was shared, also has a negative effect on affect-based trust. This shows that an application must be able to explain not just how and what data is being shared, but also make the user aware of the frequency of information sharing to achieve high overall trust. The frequency of data sharing impacts how accurate information profiles of users can be constructed. More accurate profiles are good for the people utilising these profiles, but it should also be seen as a greater violation of privacy if the user is not aware of the collection and secondhand use of their information. The low amount of participants in their study do not present statistical results for the general mass of users, but as it corresponds well with Ur et al. [55] findings, we find the research relevant. Many of the participants also asked if they could have the interface alerting them of data leaks installed on their phone. The fact that this interface was based on TaintDroid (described in Section 2.3.2) makes this hard for novice users to use because it requires them to go through a rather complex installation process that also will void their warranty.

Good et al. [20] conducted an ecological study to research to what extent users are able to evaluate the potential consequences of installing an application based on being presented with notices such as software agreements, terms of service (TOS), end user licensing agreements (EULA), and security warnings. They asked 31 participants to install five applications. During the installation



process either a full EULA or a short notice summarising the most important aspects of user privacy combined with the possibility to see the full EULA was presented. Participants in their study applied one of four different strategies when installing the applications:

- **Install first, ask questions later:** Installed all applications without regard to the privacy notices with the intention of examining them in greater detail later.
- **Once Bitten, Twice Shy:** Their decision to install applications were influenced by previous negative experiences like being victim to a "phishing attack". They would only install an application if they found it needed and skimmed EULAs and program information.
- **Curious, feature-based:** Primarily interested in new and interesting functionality. They would only install an application if it was popular or offered something that they would want or need.
- **Computer-Phobic:** Generally wary of anything that had to do with installing programs or configuring a computer. Very concerned with any warning that popped up, and were reluctant to install anything.

Further, they found that over the groups, 80% of the participants who expressed concerns about privacy were primarily interested in the functionality of the application. What this implies is that developers may value privacy less in their application as long as they supply functionality able to convince the user their application provides high utility. This indicates that it is easy for developers of malicious software to hide the actual intention the software behind some attractive functionality. This is a situation where users' willingness to use the system over-weights their perception of confidence in using it.

After the installation process, participants were interviewed to find out to what extent the notice presented to them had impacted their decision to install new software. They found that the participants were ambivalent towards the EULAs in the software they installed. Most of them were aware that they agreed to a set of terms by installing the software. They were unable to recall the content of the agreement, and it rarely influenced their decision to install a program. The short notice was more effective, and 64% stated that it influenced their decision. They were also more able to remember the content of this notice after installation.

Understanding users' general relationship to both privacy in general and for smartphones is important when developing new applications. We have seen that EULAs do not have a high impact on users decisions when installing software. EULAs are intended to provide a foundation for users to comprehend how an application functions and what consequences use of it leads to. It seems that

many users trust applications based on wrong perceptions. This should motivate new applications to properly explain their behaviour. If the trade-off for installing a free application is sharing of personal information, users should be made aware of this.

### 2.3.3 Users do not Understand Permissions

Required permissions are shown in the Android installation process to help users evaluate the functionality of the application and to "give the user the option to not install the application if they feel uncomfortable" [23] (See Figure 2.7). In essence, the application permissions users has to agree to in order to install the application are a tool for them to evaluate their decision whether to proceed with the installation or not. This vision is in high contrast to research performed by Kelley et al. [33]. They conducted an online study with 77 participants seeking to get an understanding of their interactions with their Android devices. They also map issues surrounding the display of permissions, the safety of the Google Play Store, and possible harms of information sharing. Later, 20 of these participants attended a more thorough lab interview. A part of this interview consisted of showing them ten different permissions with accompanying descriptions of the permissions and asked if they could explain how they understood them. The result shows that none of the participants correctly understood all of the permissions. Neither were they able to connect the relationship between an application's functionality and its requested permissions.

Fang et al. [15] conducted research on issues regarding the Android permission system. They identified several issues with the permission system, two of them being coarse granularity of permissions and insufficient documentation. Many permissions are too coarse-grained, which means that the scope of the potential disclosure is too broad to be limited to a single permission. By combining these two issues, one can deduce that to explain a permission, one need to give an overview of every type of disclosure that a particular permission could allow. It also needs to be explained in a manner that a novice user are able to comprehend.

Rebecca Balebako and Sadeh. [48] introduces the term "Soft Paternalism". This is a concept adapted from behavioural economics that takes into account the cognitive and behavioural differences in individuals. They angle the concept at privacy, and how it can be used to reduce asymmetry and assist users in privacy-related decision-making. Based on the fact that users often lack the ability to make decisions about their privacy, the Soft Paternalistic approach suggests the following solution: "making an individual aware of the biases, lack of information, or cognitive overload that may affect their decision". They point out that there is a fine balance between keeping privacy settings easy and oversimplifying, nudging user towards devious configurations. This implies that by

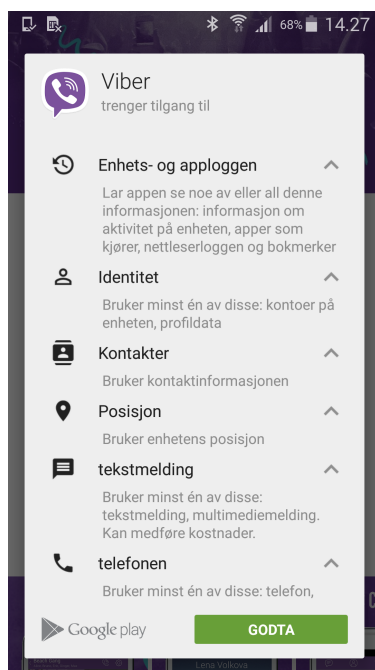


Figure 2.7: The figure illustrates how Android presents requested permissions when installing Viber (a popular communication application).

putting too much scope on few configuration choices may have a damaging effect.

In their research on privacy decision-making, Acquisti and Grossklags [1] discusses the balance between incomplete information and information overload, and how these factors affects a user's decision. In their conclusion, they state that even with the appropriate amount of information, users do not follow a rational pattern in their decision-making. This is due to individual factors such as attitudes, knowledge of risks, different abilities to achieve trust, faith in the data collectors ability to protect their information, and so on. This indicates that there is more to it than just statically providing information. Individualisation and personalisation of information should also be taken into consideration and be based on factors such as trust-building and users attitude towards the system.

Seeing the permission problem being brought up in several research papers only reinforces the potential for increased privacy by providing users with more extensive information about permissions. This builds a stronger foundation for them to make informed decisions in privacy related decision-making.

### 2.3.4 Users Value Utility over Privacy

Research done by both Cranor et al. [10] and Good et al. [20] indicates that the majority of users are interested and concerned about their privacy. However, their study also reveals a negative aspect about the majority of users: they are willing to compromise or freely give away sensitive information if the value or benefit of doing so is satisfying. This also includes users that categorise themselves as highly concerned about privacy. This behaviour have also been uncovered in fully transparent systems. The balance between benefits and costs are often closely linked the context of the privacy. Acquisti and Grossklags [1] points to some of the common factors involved in the decision-making process:

- **Incomplete information:** Users does not know the full extent of what and how information about them are being collected and used. The potential risks of sharing information are not clear. This includes changes in policies, loss of data or the possibilities for disclosure.
- **Bounded rationality:** Users does not have the ability to comprehend all relevant information. This includes memorising and processing the information provided, and instead relying on simplified mental models.
- **Psychological deviation from rationality:** Users have all relevant information available, but still deviates from rationality. This includes elements such as personal- and motivational limitations.

They conducted an online survey where they asked a variety of people to answer questions about age, demographics, employment, salaries, and questions regarding behaviour and attitude on the topic of privacy. 119 responses were collected. The aim of the survey was to identify the gap between users attitude and behaviour towards privacy. On the questions regarding attitude, 82.0% answered that they were over moderately concerned about privacy. 73.1% answered that they did not think that there was enough privacy in society today. 37.2% meant privacy policies were quite important. It is reasonable to conclude that the subjects in this survey are both interested and concerned about privacy. Further, to investigate the actual behaviour of users, 21.8% admitted to having revealed their social security number to services providing benefits, such as discounts, better services or recommendations. 28.6% had given away their phone number in different settings. The survey also showed that users were not especially concerned if the collected data were connected to their person. Their research also show the impact time-inconsistent discounting has on users' decision process. If discount offers are on a time-limit, even poor discounts of insignificant savings, users were more likely to accept, and possible compromise long-term privacy, for short term benefits. Again, they point to the lack of information, simplified mental models,

and overconfidence in systems as the primary reason to why people has the tendency to compromise privacy for benefits. Even the most privacy concerned users from the survey revealed that they had disclosed information for benefits. They also state that they believe that the information collected about them, almost in no circumstances, is being used for something malicious.

Spiekermann et al. [52] conducted an experiment to examine the relation between users self-reported privacy preferences and their actual behaviour. Through a dialogue with an anthropomorphic 3-D shopping bot, users were asked to answer a series of questions, some of highly personal nature. The readiness to disclose information, and the inconsistency in behaviour in this scenario were higher than expected. They formulated questions in cooperation with retail sales personnel. Further, in addition to questions that would likely be asked in such a settings, the researchers occasionally also asked questions from a category with non-legitimate questions. These included questions about how photogenic the users felt, or what the users do with their photographs. In their investigation of behaviour, they found that around 28% of privacy fundamentalists revealed their home address, and as much as 40% of the subjects that place themselves in the category of identity-concerned, gave up this information. During the dialogue with the bot, subjects continued to answer non-legitimate questions, and most also accepted to sell their data to an anonymous entity. The conclusion states that the willingness to disclose information and details in the answered questions is high. A relatively revealing profile of the user could be established on the basis of one dialogue with a simple shopping bot.

Research presented in this section indicates that even the most concerned privacy fundamentalist do not always maintain their privacy properly, and can easily be tricked by some intricate methods. We have with these studies acquired a better understanding of reasons why and how users might compromise their privacy for better utility.

### **2.3.5 There is no such thing as a free application**

In Google Play Store, publishers can choose to either sell their application or let users download it for free. A common practice, especially for games, is to publish two versions: one for a small amount of money, ranging from less than a dollar, up several dollars, and a second version for free. The free version often includes advertising. Advertising elements are placed on the screen, often in menus or when a game is paused. Developers earns money on free applications with advertising whenever a user downloads it and when included advertisements are clicked. Many of these advertisements are personalised, which means that they need to collect personal information about the user.

Vigneri et al. [57] addresses the fact that there are no mechanisms for users to understand which third-parties applications are talking to and how often they converse. They wanted to investigate which applications from Google Play Store made connections to advertisement-servers. Further, they downloaded 2000 free applications from all categories and monitored every URL requested by each application by rerouting the Internet traffic through a proxy. They used lists of known ad-URLs from EasyList and user tracking site from EasyPrivacy to match gathered URLs against known advertisement servers. Their result show that from the 2000 applications analysed, 250 000 different ad-URLs were requested. 10% of the applications analysed requested more than 500 ad-URLs. In one extreme event, a application showed connections to more than 2000 distinct ad-URLs. 30% of the applications connected to user tracking sites, some alone to over 800 servers. This is in no way detectable by the user.

IEEE Spectrum [29] writes about PrivacyGrade, which analyses popular Android applications. Their analysis showed that Google's advertising library, Ad-mob, were found in over 407,181 different applications. Analysis of devices containing applications with ads has shown an increase in the resources used. Results show up to 22% increase in memory use, CPU usage increased with 56%, and battery life reduced from 2.5 to 2.1 hours on average [56].

Lately, several media articles have written about popular applications that misuse their privileges and performs hidden operations without the users knowing about it. Computerworld [9] states that the new trend is applications recording audio through the microphone and transmits it to their servers, even when the device is idle. The publishers claim it to be a feature that gathers contextual information to provide benefits for the users. It further states that users of the applications that included this feature were unaware of this happening.

The more providers of advertisements know about users and their life, the more profitable and efficient their service or marketing will become. Personalisation and user benefits are in many ways useful and can provide value to users. We believe that users should be aware of it, and choose for themselves if they want to share information, not just automatically agreeing upon it when installing new applications. This further motivates us to detect sensor usage and monitor the behaviour of third-party applications.

### **2.3.6 Explanations are needed for users to trust applications**

As introduced in Section 2.1.7, explanations are important for many reasons. Two of these are to provide transparency and increase the confidence and trust in human-computer relationships. As a part of a study on how to establish trust in adaptive agents, Glass et al. [19] asked their participants what would

help them build confidence in systems. 71% answered transparency, and 100% answered transparency to be an important factor affecting overall usability. Further, they found that the components that received the highest level of trust were the ones that provided feedback about what they were doing and how they reached a concrete result. This implies to also include justifying explanations. Systems that performed actions without explaining the underlying reasoning lost a significant amount of trust. The subjects further identified explanations of system behaviour, providing transparency of its reasoning and execution, as the key factors in system comprehensibility. In search for the value of different types of explanations, they revealed a critical problem: most users do not know that computer systems can provide explanations. Hence, it is a need to provide explanations in a way so that it is obvious that the system can provide them. In their investigation of granularity of explanations, they identified the need for adaptation of explanations customised to the needs of different users. Hence, the need to also include conceptual explanations. All subjects were unanimously agreeing upon the fact that there was a need for a certain amount of details, and not feedback containing just "Okey", "Not Okey", or similar vague responses. If the user is using the system for the first time, or has no other basis for trusting the system, a transparency explanation was identified as a key element in building a baseline of trust. These findings provide us with important knowledge and can serve as a starting point for our development of explanations, and help reach our goal of transparency and trust in our application.

### **2.3.7 There is a lack of good Android privacy applications**

In our search for any similar applications matching our vision, we found several applications that to some extent provided similar functionality to our envisioned application. Some of these are mentioned in this section.

Permission Explorer [11] show how currently installed applications can access protected system APIs through permissions. It provides its users with a list of installed third-party applications and what permissions each application requires. Clueful Privacy Advisor is a Android application developed by Bitdefender [6]. It shows installed applications on the user's device and give each of them a rating based on their concerns about privacy. We have not been able to find detailed documentation of how this is done, however, in the application description it stated that the application communicates with Bitdefenders database over verified Android applications. Another application, John McAfee's D-Vasive [32] provides a vast amount of features in securing and detecting sensor usage from third-party applications. We downloaded and tested the free version of this application. The fact that not only did this application feel and behave poorly developed, it were also difficult to use, lacked transparency and to some extent

disrupted the natural use of the phone surprised us.

The idea of detecting sensor usage from the D-Vasive application corresponds to our desired functionality and is experimented on in Section 3.1. Apart from this, little inspiration were found among the existing applications in the Google Play Store.

### 2.3.8 Related Work

Our literature review indicates that many efforts has been made to provide analysis mechanisms for identifying misuse of personal data on smartphones. These efforts generally use static analysis (either the source code or the binary of an application are analysed to identify possible sources and sinks of data leakages), dynamic monitoring (the behaviour of an application is examined at runtime) to identify possible misbehaviour of applications extracted from the public application market, or analysis of requested and used permissions. In this sections, we will present some methods we found interesting for motivating our further work and their results.

TaintDroid [13] is an extension to the Android operating system. It tracks the flow of privacy sensitive data through third-party applications utilising dynamic monitoring, precisely dynamic taint analysis. Their application monitors how third-party applications access and manipulates users' personal data in real-time. If it finds that another application is accessing private data, it posts a notification telling the user what data was accessed and which application accessed it. TaintDroid has to run on an unlocked bootloader (e.g. a rooted device) to be able to monitor data flow in other applications. This limits it from being used by users having a standard distribution of the Android operating system and makes it unfit for our problem domain. However, it provides a useful tool for performing analysing of third-party applications. The developers applied it to a set of 30 randomly chosen popular Android applications available through Google Play Store. They found that 15 applications sent sensitive data to advertisement servers without notifying the user or indicating it in their EULA. Further, two applications transmitted phone information to content servers and seven leaked the device ID - both without specifying it in the EULA. These results indicate a lack of transparency in applications. When denying users the ability to know what their sensitive information is being collected for, and in some cases sent to advertisement servers, these applications violate privacy and produces high asymmetry in the information flow. As the source for TaintDroid is free and available, we can use it as a tool for performance evaluation as it has proven to be effective in detecting information leaks in applications.

Felt et al. [16] performed research aiming to reveal over-privilege by Android developers in the use of permissions. Their method consists of the development



of a tool, Stowaway, using static analysis to detect unneeded use of permissions. To perform the analysis, Stowaway uses disassembled application executables as input to perform code analysis and identify all calls to standard Android API methods. Then they collect all strings that could be used as content provider URIs, links those strings to the content providers permission requirements, and detect the sending and receiving of intents that require permissions. They use this method combined with a set of Android permission descriptions to identify what permissions are needed for each API call. Disassembling application executables are not possible on third-party Android applications, hence the static analysis part of this method do not apply to our problem domain. Android documentation was unable to provide the researchers with a sufficient set of permission descriptions to be used with Stowaway. To build such a set, they modified Android 2.2's permission verification mechanism to log permission checks as they occur. They were able to observe the permissions required to interact with system APIs through test cases for API calls, content providers, and intents. Performing this method, they covered close to 100% of methods belonging to protected APIs and mapped those methods to their respective permissions. For our problem, it would be interesting to have a more thorough description of permissions. When explaining to users what each permission means and to what extent it accesses their personal data, knowing exactly how they interact with the system will provide valuable information. Their method is well documented and seems reproducible. However, it is used on an old version of Android (2.2). For our purpose, we would need to apply the method to a more recent version, probably 4.0+, to get satisfying results due to the changes in Android operating systems since version 2.2. If this is feasible, it presents the need for us to research more recent versions of Android and what changes are made in the permission system. Stowaway was used to analyse permission over-privilege in a set of 940 Android applications. Felt et al. [16] also did a manual analysis of 40 randomly selected in this set to identify tool errors. Their manual analysis uncovered a 7% false positive rate. They conclude the reason for this to be incompleteness in the permission map. In addition to the false positives, over-approximation of content provider operations are considered a weakness by Stowaway because it might overlook some over-privilege. Their method does not perform dead code elimination. In their automated analysis of the full set of applications, Stowaway reported 32.7% of the applications to be over-privileged. Of these applications, 56% have one extra permission, and 94% have 4 or fewer extra permissions. They argue the low amount of extra permissions is due to developers trying to reduce the use of unnecessary permissions. Better documentation of this part of Android would probably help even further. Hence, useful explanations of permissions are not only needed to provide novice users with insight to how applications are accessing sensitive data, they are also necessary for developers to enhance applications, reducing the use of permissions

to facilitate better privacy for end users.

The results produced by Stowaway are questioned by Geneiatakis et al. [18]. They argue that statistical analysis lack the ability to account for the runtime context and that this make them prone to false positives that indeed was the case in the approach suggested by Felt et al. [16]. Geneiatakis et al. [18] propose a way to detect over-privilege by combining static analysis and dynamic monitoring. Their approach reverse engineer applications to identify all methods used and supply each of them with monitoring code before each call to protected APIs in the Android operating system. Then, they assemble a runnable application based on the reversed engineered code able to monitor each method call. This application is then analysed at runtime that enables them to record exactly which calls to permission protected APIs the application executes. Finally, they compare the API calls to the permission mapping published by Felt et al. [16] to connect the API calls to their respective permissions. This approach reduces the amount of false positives by also analysing which part of the application code is not being used. False positives are only present when permissions declared in the application manifest were not reached. They applied their analysis tool on 256 application all randomly selected from the top lists on Google Play Store. Excluding the cases with present false positives, the found that 87% of the applications had unused permissions, a number way higher that what Felt et al. [16] research indicated. When comparing the two approaches, using both static analysis and dynamic monitoring seems like a more robust solution. It provides a way to eliminate false positives in the majority of analysed applications, and it also detects dead code obtained in the static phase. Geneiatakis et al. [18] performed their method on a lower amount of applications than Felt et al. [16] which may reflect that they lack a representative dataset, but the fact that the applications that were chosen belonged to the top lists on the Google Play Store makes their number highly relevant. Their result further highlights the need for better explanations of permissions. It also reveals the most used permissions among over-privileged applications (see Table 2.2). Such results are valuable for our future work in explaining permissions to novice users. If permissions are frequently requested, but not being used by the application requesting them, it will probably lead to confusion among users because there are no possible way of justifying their presence.

Privacygrade.org is a website which provides Android applications with grades reflecting their concerns to privacy. Grades are assigned using a privacy model built using the research conducted by Lin et al. [36] as a foundation. The model is based on a combination of static analysis and crowdsourcing. Their static analysis is based on the use of TaintDroid to obtain information about applications behaviour. Specifically, they use it to find which third-party libraries are being used by applications. For example, if a map application is using GPS to acquire

Permission type	Usage by application
WRITE_EXTERNAL_STORAGE	11%
RECEIVE_BOOT_COMPLETED	9%
READ_PHONE_STATE	9%
ACCESS_NETWORK_STATE	18%
ACCESS_COARSE_LOCATION	11%

Table 2.2: Most used permissions among over-privileged applications presented by Geneiatakis et al. [18]. Over-privileged application request permissions that are left unused. This makes it harder for users to evaluate application functionality based on requested permissions.

user position, they seek to know if this position is being sent to third-party libraries not performing geo-operations. Their goal of performing crowdsourcing is to capture people’s expectations about what an application does and does not do. To precisely measure people’s expectations, they use the notion of their mental model, a simplified model that describes what people think an object does and how it works using an application as the object. To achieve this, they conducted an online survey resulting in 5360 valid responses. The survey presented screenshots of an application with a following description of the applications functionality. Participants were then asked if they expected the application to require its defined permissions. Then, they had to answer why that application would require such permissions and if they felt comfortable letting it do so. Finally, they presented them with their result from the analysis of the application performed with TaintDroid showing how the permission was being used and asked if they based on that information would feel comfortable installing it. Result from their survey highly indicates that users feel more comfortable in installing applications if they are properly explained what it does and what sensitive information it requires access to beforehand. This underlines the work done by Jiang et al. [31] by reducing asymmetry in information flow by the fact that applications fail to inform users about how they collect and use their personal data. As shown by Enck et al. [13], EULAs often fail to inform users how this is done, leaving them to use the requested permissions to evaluate impacts on their privacy by installing an application. The fact that users do not feel comfortable by the low transparency in applications directly links to the Model of Human-Computer Trust Components [37]. It shows that the installation process of an application fails to achieve high levels of both affect-based and cognition-based trust, leading to mistrust in the application when it fails to explain its behaviour.

Based on their findings, Lin et al. [36] suggest a new privacy summary interface (see Figure 2.8). It describes what an application are doing by utilising key

findings of users' expectations to applications. The interface thoroughly highlights the purpose of access to sensitive resources and also incorporates other users' perceptions about the applications behaviour based on their crowdsourcing. They performed user evaluation on three perspectives:

- **Privacy awareness:** Whether users are more aware of the privacy implications.
- **Comprehensibility:** How well users understood the privacy summary.
- **Efficiency:** How long it took participants to understand the privacy summary, measured by the number of seconds they spent on reading the privacy summary screens.

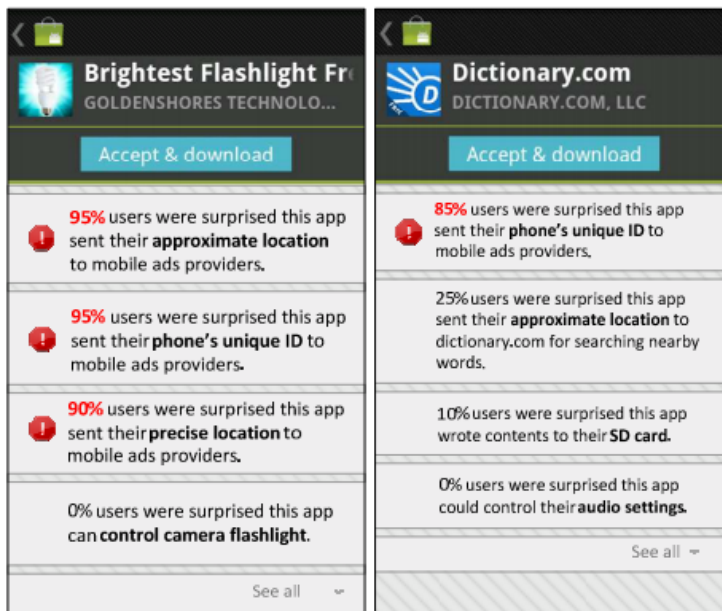


Figure 2.8: Improved Android privacy summary interface developed by Lin et al. [36]. Their interface provides a better foundation for users to comprehend if an application should require certain permissions by visualising how other users evaluate the issue.

Their finding suggests the new interface made users more aware of privacy when installing applications. It also made users more able to describe what sensitive information applications required by requesting certain permissions, and

finally they report participants in the evaluation spent less time reading the privacy summaries. This further motivates the need for explanations to increase users' trust in applications, and it is certainly interesting to our problem domain. A thinkable approach is to identify what permissions are least comprehensible to novice users and then perform a study on how they relate to these permissions and how well they understand consequences of accepting them. We could use this to further refine explanations about permissions to users. We can not change the Android installation screen, however, by detecting when users install new applications on their device we could show them an improved permission summary before they take it in use. This would make them more able to evaluate the application. It could also be used merely to present different permissions to the users, learning them how they work so that it can function as a basis for future decisions.

Kirin is a lightweight Android security service developed by Enck et al. [14]. It analyzes applications at install-time mainly by checking if they can pass a set of security rules defined to detect potential dangerous behaviour. When creating these rules, they seek to the field of security requirements engineering. Security requirements engineering is based upon three basic concepts:

1. **Functional requirements:** Defines how a system is supposed to operate in a normal environment.
2. **Assets:** Entities that someone places value upon.
3. **Security requirements:** Constraints on functional requirements to protect the assets from threats.

They found none of the existing techniques in the field applicable for the Android platform. To fill this gap, a new procedure for identifying security requirements for Android was developed. Some examples of rules defined for Kirin was found by iterating through these five steps listed below:

- An application must not have `PROCESS_OUTGOING_CALL`, `RECORD_AUDIO`, and `INTERNET` permission labels.
- An application must not have the `SET_PREFERRED_APPLICATION` permission label and receive Intents for the `CALL` action string.
- An application must not have `RECEIVE_SMS` and `WRITE_SMS` permission labels.

These rules are incorporated into the Kirin service running on an Android device to detect potential dangerous applications. Enck et al. [14] show empirical results by testing Kirin on 311 popular applications on Google Play Store. Of

these 311 applications, only 12 failed to pass their security rules. This number was reduced to 10 after adjusting some of the rules and by performing manual analysis, they found that 5 of these incorporated potential dangerous functionalities. The remaining 5 had a dangerous configuration of permissions but based on their description their functionality was concluded to be reasonable. As previously shown, many applications are over-privileged with permissions. This may affect Kirin because permissions do not necessarily reflect the functionality of an application if they are unused. However, by considering the most often unused permissions, adjustments can be made to the security rules. Moonsamy et al. [41] research on mining permission patterns for contrasting clean and malicious Android applications can also be used to further refine these rules. They identify most required and also the most used permissions by a set of applications acquired from Google Play Store (see Table 2.3). Since Kirin are developed for an early version of Android, several adjustments must also be made due to the changes in the Android API. Either way, their method for identifying security requirements for Android is interesting for our problem domain. Defining such rules presents a method for analyzing application without both static analysis and dynamic monitoring. Since some of the applications violating security rules do not pose a risk, user interaction should be present so the violations can be seen in comparison with the application description. By providing good explanations of these rules to the end user, they might help them decide whether or not to keep an application.

Inspired by Enck et al. [14], Sarma et al. [50] proposes a model for detection of malicious Android applications. They hypothesise that it should be possible to know the intended functionality or benefit provided by an application by taking into account which Google Play Store category it is uploaded to. They believe applications in different categories often request different kinds of permissions. They study the percentages of applications requesting certain permissions across all Google Play Store categories. Requesting permissions with low request percentage in the respected application category is then seen as a sign of possible malicious functionality. They find their model to have a higher detection rate than Kirin, but it comes with a much higher complexity. This makes it hard to provide good explanations of why applications may be dangerous to the end user.

Peng et al. [47] aims at creating a probabilistic risk-scoring algorithm for Android applications based on permissions. Their goal is to effectively communicate the risk of an application to users, by calculating the likelihood of it being similar to a malicious application. They experiment with different approaches to the Naive Bayes Model (NBM) to analyse how well they satisfy the following three criteria:

- **Monotonic algorithm:** Removing of a permission from the application should reduce the risk score.

Used permission	Frequency
INTERNET	94.6%
ACCESS_COARSE_LOCATION	91.7%
VIBRATE	77.7%
WAKE_LOCK	67.3%
ACCESS_WIFI_STATE	47.6%
ACCESS_NETWORK_STATE	42.3%
READ_SMS	38.5%
WRITE_CONTACTS	34.7%
READ_PHONE_STATE	28.8%
RECORD_AUDIO	26.0%
SET_WALLPAPER	24.2%
ACCESS_FINE_LOCATION	16.2%
GET_ACCOUNTS	14.5%
GET_TASKS	10.1%
RECEIVE_BOOT_COMPLETED	9.0%
ACCESS_CACHE_FILESYSTEM	8.2%
WRITE_OWNER_DATA	4.8%
CHANGE_CONFIGURATION	4.2%
READ_HISTORY_BOOKMARKS	4.0%
EXPAND_STATUS_BAR	3.3%

Table 2.3: The figure illustrates the top 20 used permissions by malicious applications found by Moonsamy et al. [41]. This information can be used rank Android permissions based on risk of being used in a dangerous application.

- **High score for malicious applications:** Applications that are known to malicious should, in general, receive a high score.
- **Easy to understand:** The algorithm should be easy to understand.

Their dataset consists of several hundred thousand applications collected randomly from Google Play Store, which they further use to train their model. They also include a training set of applications which is known to be malicious. Their results show that the model Naive Bayes Model with Informative Priors provides the best results, with over 94% change of identifying the malicious applications in the dataset. This model takes into consideration both the amount of permissions an application demands and enables for different risk factors to be awarded each permission. It is also monotonic and provides understandable feedback to why applications are given high risks.

Our literature review reveals approaches for analysing Android applications

for privacy and security issues usually rely on static analysis and/or dynamic monitoring. These approaches use code analysis and monitoring of applications running on modified versions of Android. We did not find relevant research papers focusing on application monitoring and privacy analysis performed through third-party applications available on the public application market in our literature review. This is probably mainly because of the limited monitoring capabilities of other applications through the standard Android system API (as shown in Section 3.1). These techniques are generally not applicable in our problem domain, but their results provide a base of indicators our further research should focus on when developing our application. Further, studies show that focusing on finding patterns in permissions to detect unwanted behaviour can be effective. This approach may be affected by the large extent of over-privileged applications. However, we have not found any comparison of over-privilege between malicious and clean applications to confirm this.



## Chapter 3

# Experiments and Results

*This chapter will present experiments conducted to find answers to our research questions. For each experiment, we first describe our plan and methodology. Second, we show the experimental setup, and finally, we sum up our results and review to which degree we completed our experimental plan.*

### 3.1 Detection of Third-Party Application Behaviour

The Android API provides developers access to large amounts of information on a user's Android device. This information is intended to be protected by the Android permission system, but as our literature study indicates, this security mechanism often fails. Many applications are over-privileged, and studies show that users tend to install applications regardless of what information they might disclose. This problem creates a need to detect if any applications are misusing their privileges, and what information they are accessing.

#### 3.1.1 Experimental Plan

This experiment is conducted to examine to what extent a third-party application can monitor and extract information about the behaviour and actions of other third-party applications installed on an Android device. This is in relation to our first research question.

This experiment seeks to answer the following questions:

1. Is it possible to detect use of **location sensors** and identify the application that is requesting the functionality?

2. Is it possible to detect use of **camera** and identify the application that is requesting the functionality?
3. Is it possible to detect use of **microphone** and identify the application that is requesting the functionality?
4. Is it possible to detect access to **personal data** and identify the application that is accessing it?
5. Is it possible to measure application **Internet data usage** and identify to whom this data is being communicated.

### 3.1.2 Experimental Setup

The following tools were used in this experiment:

- Android Studio
- Samsung Galaxy S5 with Android 5.0 Lollipop

We created a test application for purposes of finding answers to our research questions by trying to implement custom methods for retrieving data. This application was set to run on Android versions from 16 and higher - equal to the application described in Chapter 4.

### 3.1.3 Experimental Results

The following sections sum up our results on the experimental questions.

#### Detecting Use of the Camera

Programmatically, there are two different methods for accessing the camera on an Android device. The first option is to use the native camera application which is pre-installed on all Android devices with a camera. Detecting when this application captures a picture is trivial because it broadcasts a system intent available for all third-party applications requesting the CAMERA permission to receive. However, it is impossible to know exactly which application requested the picture to be taken. This is the "safe" alternative of using the camera, guaranteeing that the user can see what is being captured.

The second option is to request a camera object from the Android API and implement this for taking pictures in an application. This method can be misused by malicious applications to capture pictures without the user knowing about it (as described in Section 3.2.3). Capturing pictures with the camera object do not send out system wide intents available for third-party applications to receive.

It is possible to detect when the object is in use because it can only be opened by one application at a time. Hence, detection can be done by repeatedly trying to request the camera object from a service and check if it is available or not. We found that implementing this solution for detection often would affect other applications as they were trying to request the camera at the same time our application was checking it for availability. In many cases, this made the applications crash due to lack of error handling when they could not successfully retrieve the camera object. Another point is that this method would only detect if the camera is being used, not if any pictures were taken. We ended our experiments after we experienced the bad influence on other applications. Theoretically, it should be possible to listen to changes on image folders to find out if a picture were saved when usage of the camera object was detected.

Using this detection technique requires the CAMERA permission. It will probably be difficult to explain why this permission is requested as it is not used to capture pictures.

### **Detecting Use of the Microphone**

Detecting the use of the microphone can be done in pretty much the same way as the second method for detecting camera usage (see Section 3.1.3) only with the use of the media recorder object. This would include implementing a service that is running repeatedly trying to access the microphone. This method also has the same consequences for other third-party applications attempting to request the microphone as for camera detection, and it is not intuitively possible to detect if the recorded audio is stored. However, the proposed method for checking if an image was captured proposed in Section 3.1.3 might be applied to this problem.

Using this detection technique requires the RECORD\_AUDIO permission.

### **Detecting Retrieval of User Position Events**

It is not possible for a third-party application to detect the exact application requesting the smartphone's location. For the user, there are no way knowing when his/her location are being used other than overlooking the status bar icon that indicates use of locations sensors. However, we discovered that the most recent coordinates retrieved by any of the smartphones location sensors are stored and made available in the Android Location API. It is accessible through the method `getLastKnownLocation()` in `LocationManager`. By periodically calling this method and checking if the stored coordinates has changed since the last time it was called, a third-party application can detect if another application have used location sensors. In our experimental application we utilise this by running a service every two seconds that calls `getLastKnownLocation()` and detects if the output has changed.

Using this detection technique requires the permission for `ACCESS_FINE_LOCATION` and/or `ACCESS_COARSE_LOCATION`.

### **Detecting Access to Personal Information**

Many events happening on the Android device causes detectable broadcasts or events to be sent. We wanted to examine if there were any events triggered when an application accessed personal information like text messages, the image gallery, the contact list, or the account manager containing email accounts and other personal information.

Most of the information on Android devices are stored in SQLite databases which are accessible through ContentResolvers. An application can store data in their own database or connect to another application's database. We found that there exists a listener that can detect events in a database. This listener is called ContentObserver, and the method `registerContentObserver()` will detect events. Further, research shows that the listener only fires on changes and does not trigger on read-only. Therefore, there are no means to detect if a third-party application accesses a database which stores text messages, images, or the contact list. This information is thus only protected by the permission system.

We also wanted to investigate if it was possible to detect if an application accessed information through intents. Intents can be detected by registering a broadcast receiver with the corresponding intent filter. The image gallery can be accessed by a user-driven event. For an application to gain access to an image file stored on the device outside the application's own database, a user must manually pick the images from a preview of the image gallery. This can not be done programmatically. This means that the user's images are safe from access unless the user chooses to access the image gallery through an intent fired by the application. In our search for detecting access to the Account Manager, which holds user accounts and credentials, we did not find any way to detect access. Some of the functions when using user credentials causes a prompt to appear, otherwise this information is only protected by the permissions `GET_ACCOUNTS` and `USE_CREDENTIALS`.

These experiments show that there are very limited possibilities for detecting third-party applications' access and use of personal information.

### **Data Usage**

The amount of data usage, or Internet traffic, an application is using can be valuable when seeking to investigate and analyse its behaviour. We wanted to create a service that analyses each application's data usage, and notifies the user about high amounts.

In our examination of the Android API, we found that the total amount of downloaded and uploaded data usage of each application can be extracted from the TrafficStats API. A drawback is that the number of bytes provided is only since the device booted up. This implies that usage before this can not be retrieved from the system. To gain a better understanding of an application's behaviour, we wanted to separate it into foreground and background data usage. Foreground data usage measures the data used when a user is actively using the application, and background usage is measured when an application is using data from background services. We found that the developer API does not provide any methods for this separation, so we needed to create a service that divided the data usage into these two groups. We created a service that ran at a given interval, checking which application was running in the foreground. If an application was registered to be in the foreground between two intervals, the data usage between those two intervals can with great certainty be said to be foreground data usage, and added to the respective amount. If not, it will increase the amount of the background usage.

Through our experiments with the interval length, we found that if the service ran often, it would use a high amount of device-resources and drain the battery considerably. If it were ran too rarely, it might miss a significantly amount of data used in the foreground. This approach proved to work, but with a noticeably shortening of battery lifetime when ran at an interval giving the best results. The compromise is to only provide the user with information about total uploaded and downloaded, but this feature already exists in the Android operating systems' settings menu. Our approach may be experienced as malicious since the resources needed affects battery lifetime, and is therefore not prudent to use.

## 3.2 Possible Malicious Use of Android Sensors

Android provides easy access to different sensors (e.g. camera, microphone, and GPS) for third-party applications. This motivates our experiment to find out how and to what extent these sensors may be used for malicious purposes.

### 3.2.1 Experimental Plan

This experiment is conducted to examine to what extent an application may use sensors for malicious purposes. It is related to both our first and second research question. We seek to answer the following questions:

1. Can the **camera** be used to capture pictures or video without the user's knowledge or consent?

2. Can the **microphone** be used to capture audio without the user's knowledge or consent?
3. Can the **GPS** be used to track the location of the device without the user's knowledge or consent?

### 3.2.2 Experimental Setup

The following tools were utilised in this experiment:

- Android Studio
- Samsung Galaxy S5 with Android 5.0 Lollipop

We developed a test application for purposes of finding answers to our research questions by trying to implement custom methods for retrieving data. This application was set to run on Android versions from 16 and higher - equal to the application described in Chapter 4.

### 3.2.3 Experimental Results

The following sections sums up our results on the experimental questions.

#### Malicious Use of Camera

An application may use the camera in two different ways. One option is to request the standard camera application installed on the smartphone to start and then wait for it to send back a picture taken by the user. The second option is to implement a camera function by requesting a camera object from the Android API. Option one is considered safe because an intent is sent to open a standard camera application approved by the owner of the smartphone. This guarantees control over the content obtained by the camera. It is the second option that can be misused by applications with malicious intentions, both when the application is running in the foreground and in the background.

Android requires the camera to be placed within a `SurfaceView` when third-party applications are using it, as described in option two. `SurfaceView` is a layout-object intended to show a preview of the image/video through the camera and thus prohibit applications to use it without the users knowledge. However, as `SurfaceView` is a layout-object and as any Android layout-object, it is customisable by developers, it can be created with width and height set to one pixel and also placed in, for example, the bottom corner of the application view. We constructed such a layout in our experimental application and even when we were aware of its location in the view it was impossible to detect with the clear eye.

This implementation of SurfaceView connected to the camera allows applications to take pictures while they are being used in the foreground.

Our experiments also reveal the possibility to utilise the camera object if the application is running in the background. Our initial thought was that this would be impossible because a SurfaceView needs to be present to enable usage of the camera. However, the Android SDK implements the permission `SYSTEM_ALERT_WINDOW` that protects API methods for drawing views on top of other running applications. For example, Facebook uses this to create their “chat bubbles” as shown in Figure 3.1. We discovered that this could be used to create a view containing a SurfaceView for camera preview on top of the foreground application running on the device from a background service. Again, this view can be drawn as 1x1 pixel and hence be impossible for the user to detect. Android senses this as a preview is showing, and the background service can snap pictures with the camera running in the background.

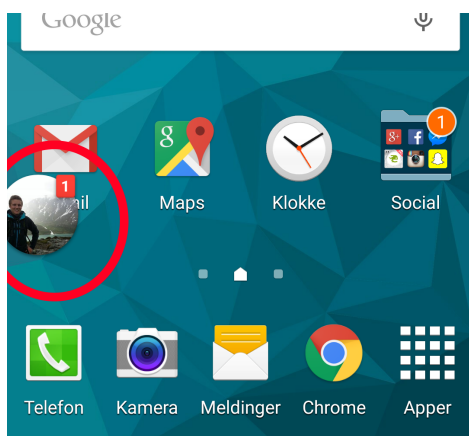


Figure 3.1: The figure show a ”chat bubble” indicating that a person has sent you a message on Facebook Messenger. The bubble is drawn on top of the Android Home Screen by requesting functionality protected by the permission `SYSTEM_ALERT_WINDOW`.

### Malicious Use of Microphone

Malicious use of the microphone is pretty straight forward for third-party applications. The microphone API is accessible with the permissions `RECORD_AUDIO`. There are no mechanisms for user control (other than the permission), such as SurfaceView is for the camera. As long as there are no other applications using

it, it is possible to eavesdrop on users when running either in the foreground or the background as a service.

### **Malicious Use of Location Sensors**

Applications can access the Android location API by requesting permission `ACCESS_FINE_LOCATION` and/or `ACCESS_COARSE_LOCATION`.

`ACCESS_FINE_LOCATION` provides access to the smartphone GPS-sensor with an accuracy down to one meter. `ACCESS_COARSE_LOCATION` uses base station triangulation or Wi-Fi access point information to determine position. Accepting these permissions present a privacy risk because developers may at any time determine the smartphone's location. However, in a micro-perspective, the last mentioned permission is less dangerous than the first since the precision of the acquired position is poor. It is possible to acquire location information from a background service, however, when a location sensor is being used, a icon shows blinking in the status bar making it possible for the user to detect that his/her position is acquired. This status bar icon is not possible for third-party applications without root access to remove.

## **3.3 Comprehensive Descriptions of Android Permissions**

Our research indicates that Android permissions do not fulfil their purpose of providing users with enough information about an application's features before the decision to install it is made (see Section 2.3.3). Without this information, users have to rely solely on the description given by the developer when considering the risks and rewards of acquiring the application. We want to provide users with permission descriptions they can understand. We also wish to highlight the privacy risks of accepting certain dangerous permissions.

### **3.3.1 Experimental Plan**

This experiment is conducted to learn how Android permissions work and what privacy risks are involved when applications are demanding them. This knowledge will be used to develop better, and easy to understand descriptions of all permissions available to third-party applications. This experiment is related to both our first and second research question.

We aim to answer the following experimental questions:

1. What functionality is made available by the requirement of a certain permission?



2. What is the best way to explain permissions and the risks involved in accepting it?
3. What is the best way to classify permissions based on their influence on users' privacy?

We aim to answer these questions by inspecting the Android API-methods protected by specific permissions, reuse results from Section 3.1 and 3.2, and the use of the Android API documentation.

### 3.3.2 Experimental Setup

The following tools were utilised in this experiment:

- PScout[3] - a tool to map protected Android API-methods to specific permissions.
- Android API documentation [21].

### 3.3.3 Experimental Results

The sections below presents our results in creating comprehensive descriptions of permissions and permission threat categories.

#### Permissions Categorised in Threat Levels

Based on the different privacy risks involved in accepting permissions, there is need to categorise them. This will help users in better understanding threats if an explanation of the threat level is presented along with the explanation of the permission. We have placed all permissions into three categories:

##### High risk permissions (Threat Level: 3)

High risk for your privacy. A permission within this category grants access to personal information about you and your contacts stored on the phone. It may also perform surveillance actions or send data from your device under your name without your consent.

##### Medium risk permissions (Threat Level: 2)

Medium risk for your privacy. A permission within this category will not have access to sensitive information, but may learn behavioural patterns about your use the device. (Example: Your connected networks or your browsing history).

**Low risk permissions (Threat Level: 1)**

Low risk for your privacy. A permission within this category poses no direct threat, and will not have access to data that alone can be used in a harmful way.

**New Android Permission Descriptions**

In our new descriptions, we have emphasised an understandable language form consisting of full sentences, the functionality and data involved and some potential threats if it is fitting.

**Access Phone State and Identity (READ\_PHONE\_STATE)**

**Description:** Allows access to read unique identity numbers connected to your phone. May also read phone status (eg. ongoing calls).

**Threat Level:** High risk

**Reprioritise Running Applications (REORDER\_TASKS)**

**Description:** Allows access to change the order of running applications. Advertising applications may use this to show ads above other applications.

**Threat Level:** Medium risk

**Prevent Phone from Sleeping (WAKE\_LOCK)**

**Description:** Allows access to prevent the screen from turning off after inactivity. May drain the battery.

**Threat Level:** Low risk

The complete list of permission descriptions and corresponding threat levels can be found in Appendix A.

## 3.4 Android Permission Patterns

As we performed our previous experiment on permission descriptions (see Section 3.3), we experienced that some permissions alone were not dangerous in means of privacy. However, combined with other permissions, they should be considered a serious risk. Motivated by this, we did research on how to best identify such permission patterns based on the expertise we possess from our previous experiments.

### 3.4.1 Experimental Plan

This experiment aims to further answer research question one and two. Enck et al. [14] shows that creating patterns using required permissions, used broadcast receivers and content providers can be used to detect malicious applications.

They base their research on a rooted Android device not matching our requirements to use an unrooted device, however, their method for creating patterns (described below) will be useful. Such permissions combined with a proper explanation can also serve as a proper way of informing users of possible threats in their installed applications. The following questions will guide this experiment:

1. Which permission patterns can be used to detect potential risks for user's privacy?
2. What is the best way to describe these patterns such that the user understand the risks involved?

As a method, we will use security requirements engineering (see Figure 3.2), as proposed by Enck et al. [14], to create permission patterns that can identify possible malicious behaviour. We will use the patterns implemented in Kirin [14] as inspiration, but we will restrict our assets to required permissions only. The following approach will be used to create permission patterns:

1. **Identify Assets:** Extract features on the Android platform. This includes permissions, broadcasted intent messages, and components of system applications (Activities, etc.). For example, the RECORD\_AUDIO permission protects the audio recorder on a Android device. Hence, this asset is considered to be microphone input.
2. **Identify Functional Requirements:** Study each asset to specify corresponding functional descriptions. These descriptions indicate how the asset interacts with the rest of the phone and third-party applications. Audio can be recorded using the MediaRecorder API.
3. **Determine Assets Security Goals and Threats:** Which security goal (confidentiality, integrity, and availability) are appropriate to use for each of the assets must be determined. Next, how the functional requirements can be abused with respect to the defined security goals should be considered to provide threat descriptions: "spyware can breach the user's privacy by recording a conversation and sending it to the adversary via the Internet".
4. **Develop Asset's Security Requirements:** Define security requirements from the threat description (e.g. determining which sets of functionality are required to compromise a threat). "An application must not be able to record audio, and access the Internet".
5. **Determine Security Mechanism Limitations:** Map the practical limitations of the proposed enforcement mechanism. For example, as the goal of Kirin was to identify potentially dangerous configurations at install time,

it cannot ensure runtime support beyond what Android already provides. Adjustments of rules due to limitations are done by iterating back to step 4.

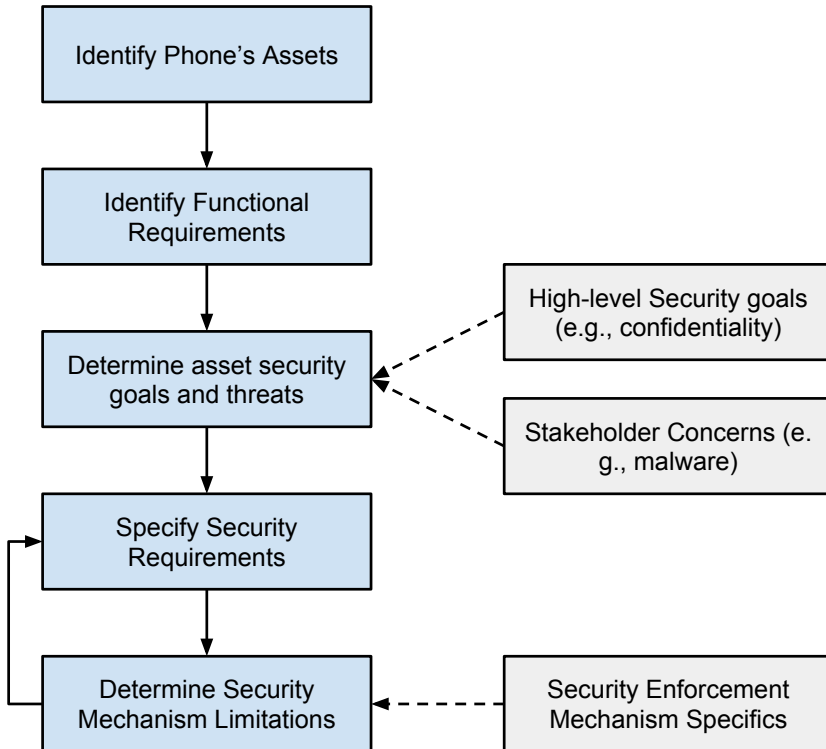


Figure 3.2: The figure illustrates a procedure for finding Android application security requirements proposed by Enck et al. [14]. We use this method to find patterns of permissions indicating dangerous functionality.

### 3.4.2 Experimental Setup

The setup of this experiment will be based on the results of our experiments in the previous sections, and the research done by Enck et al. [14] on malicious permission combinations.

### 3.4.3 Experimental Results

This section presents the developed permission patterns.

#### Relay Contact Data

An application can read information about contacts stored on a Android device and send this information to a online server. Contact information consists of full name, email address, date of birth, street address, photo and more.

1. android.permission.READ\_CONTACTS
2. android.permission.INTERNET

#### Relay SMS Messages

An application can read SMS messages received by an Android device and send these to an online server.

1. android.permission.READ\_SMS
2. android.permission.INTERNET

#### Covert Listening Device

An application can use the microphone on an Android device to record audio in the background. The application can be set to start recording as the device is turned on, without being in the foreground or user involvement. This audio can then be sent to an online server.

1. android.permission.RECEIVE\_BOOT\_COMPLETED
2. android.permission.RECORD\_AUDIO
3. android.permission.INTERNET

#### Covert Camera Surveillance

An application can capture images and video through both the front and back camera without being in the foreground or user involvement. The images and videos can be sent to an online server.

1. android.permission.RECEIVE\_BOOT\_COMPLETED
2. android.permission.CAMERA
3. android.permission.SYSTEM\_ALERT\_WINDOW
4. android.permission.INTERNET

### **Create Movement Profile**

An application can track the precise and/or coarse location and movement of the device. The application can start the tracking when the device is turned on, without being in the foreground or user involvement. The location data can be sent to an online server.

1. `android.permission.RECEIVE_BOOT_COMPLETED`
2. `android.permission.ACCESS_FINE_LOCATION` or  
`android.permission.ACCESS_COARSE_LOCATION`
3. `android.permission.INTERNET`

### **Eavesdrop on Phone Calls**

An application can record audio through the microphone while incoming and outgoing calls are made on the device. This audio can then be sent to an online server.

1. `android.permission.RECORD_AUDIO`
2. `android.permission.READ_PHONE_STATE` or  
`android.permission.PROCESS_OUTGOING_CALLS`
3. `android.permission.INTERNET`

### **Relay and Falsify SMS Messages**

An application can intercept and replace an incoming SMS message on the device. The SMS message can be sent to an online server.

1. `android.permission.RECEIVE_SMS`
2. `android.permission.WRITE_SMS`
3. `android.permission.INTERNET` or `android.permission.SEND_SMS`

### **Falsify SMS Messages**

An application can intercept and replace an incoming SMS message on the device.

1. `android.permission.RECEIVE_SMS`
2. `android.permission.WRITE_SMS`

### **Create Fake Shortcuts**

An application can create and replace shortcuts on the home screen pretending to be trusted applications.

1. `android.permission.INSTALL_SHORTCUT`
2. `android.permission.UNINSTALL_SHORTCUT`

### **Phish for Login Information**

An application can create an account on the device, and prompt the user for username and password while pretending to be a trusted account. Username and password can be sent to an online server.

1. `android.permission.AUTHENTICATE_ACCOUNTS`
2. `android.permission.USE_CREDENTIALS`
3. `android.permission.INTERNET`

## **3.5 Privacy Risk Score**

In the previous experiments, we have created better meta-data for permissions and also combined permission in patterns indicating malicious behaviour. The results of these experiments can be used to detect unwanted behaviour and educate users about threats to their privacy. However, it is probably to hard and time consuming for the novice user to use this information as a foundation to measure an applications privacy risk and comparing this risk to other applications. We want develop a method for automatic analysis of applications based on their required permissions and feedback from the user regarding his/her preferences towards these permissions.

### **3.5.1 Experimental Plan**

This experiment is related to our first and second research questions, and seeks to answer the following problems:

1. What is the best approach to rank Android applications based on their threat to privacy?
2. Is it possible to capture the user's opinion about permissions and use this to improve the application ranking?

### 3.5.2 Experimental Setup

The following information has been used to conduct this experiment:

- Permission risk scores obtained in Section 3.3.
- Permission patterns obtained in Section 3.4.
- 150 free applications downloaded from Google Play Store. The applications were selected with an even distribution from all categories on Google Play Store. Further, in each category, half the amount of applications were chosen from the top list, the remaining half were selected randomly.
- Android Studio
- Samsung Galaxy S5 with Android 5.0 Lollipop

First, we installed all 150 test applications on the Samsung Galaxy S5. Second, a test application was developed as a framework for testing solutions to the experimental questions. This application was set to run on Android versions 16 and higher - equal to the application described in Chapter 4. Finally, a stand-alone Java-application for visualising results was developed. This made it easier to adjust parameters for our algorithm.

### 3.5.3 Experimental Results

The following sections present answers to our experimental questions. We will first describe how user permission preferences are captured and then describe how we use these preferences together with results from previous experiments to generate an application privacy risk score.

#### Learning User Permission Preferences

Permission preferences are learned by interacting with the user. When viewing detailed information about a third-party application, users are presented with objective facts about what this application is able to do by having a certain permission. Each fact comes with a sad, neutral and happy emoticon (see Figure 4.6a) for the user to report his/her opinion on the information. Emoticons are preferred over textual feedback because they increase simplicity and hence, the probability of the user taking the time to answer. It is also preferred because we want to catch how the users "feel" about the fact that a particular application might take advantage of, or have the possibility to take advantage of the users' private data. Sad, happy and neutral captures state of mind in a simple, yet satisfiable manner.



Feedback on application permissions are used to generate permission weights representing the users permission preferences where a high weight value corresponds to the user not feeling good about the permission. A single permission weight are calculated by summarising replies to the objective fact connected to this permission. This fact can be present in different applications and can be answered multiple times on the same application. "Sad" reply gives +1 points, "Neutral" 0 points and "Happy" -1 points. We set initial weights to capture our knowledge on permissions (see Section 3.3) by rewarding high risk permissions a +4 point penalty, medium risk permissions a -1 point penalty and low risk permissions a -4 point penalty. Points gathered from user interaction are then added to these initial points.

To avoid a linear representation of user preferences and obtain normalised weights, we run the total point score for each permission through the Sigmoid function (see Equation 3.1 and Figure 3.3).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

where:  $e$  = the natural logarithm base

The initial weight for high risk permissions are then close to 1, for neutral permissions around 0.25 and for low risk permissions close to 0. By using the Sigmoid function, we can avoid some weights to be unreasonable high because they are connected to permissions requested by a large number of applications. Facts on popular permissions will have more user replies and could achieve very high values if the weight were calculated as a linear function. We merely state that as the weight approach 0 or 1, they are either preferred or not preferred. This makes the replies to permission facts count in a more natural way. "Happy" replies counteract high risk permissions, but we do not want having a single reply to adjust the weight to much. By using the Sigmoid function, replies counteracting the permission weight will count more if there has been many equal replies earlier.

### Privacy Risk Score

The privacy risk score for an application is composed of two separate scores, namely permission score (see Equation 3.2) and risk indicator score (see Equation 3.3).

$$permission\_score(x) = \sum_{i=1}^n 10.0w_i \quad (3.2)$$

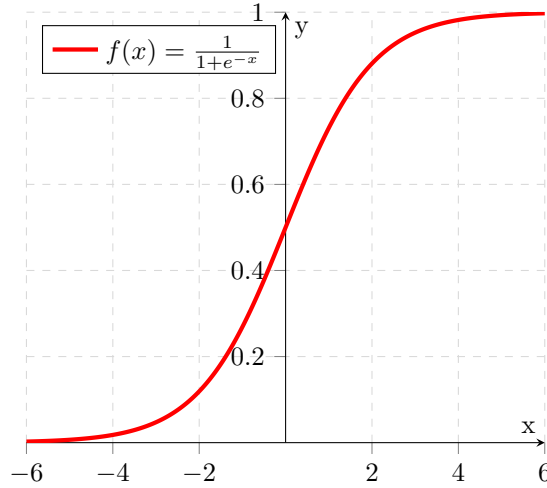


Figure 3.3: An illustration of the Sigmoid Function Curve. We use the Sigmoid function to normalise preference weights to a value that penalises negative feedback and rewards positive feedback.

where:  $w_i$  = the user preference weight for permission  $i$  in application  $x$

$$risk\_indicator\_score(x) = \left\langle \sum_{i=1}^n \left( \sum_{j=1}^r 10.0w_{ij} \right) \right\rangle \quad (3.3)$$

where:  $n$  = the number of risk indicators violated by application  $x$   
 $r$  = the number of permissions in the risk indicator permission pattern  
 $w_{ij}$  = the user preference weight for permission  $j$  in risk indicator  $i$

Both score functions are weighted sum functions based on permission weights described in Section 3.5.3. These weights are multiplied with 10.0 to increase the range of both scores. Based on user preferences, a low risk permission can be counted as high risk if sufficient negative preferences is present and counter-wise with high risk permissions and positive preference. Medium risk permissions can be weighted both ways based on preference. Hence, in the permission score, one permission can count maximum 10.0 points and minimum 0.0 points.

Defined risk indicators can be found in Section 3.5.3 presented as "permission patterns". The essence of these indicators are that a permission might not pose a high risk by itself, but in combination with one or more other permissions, they

may represent a threat. The risk indicator score is produced by calculating the average of permission weights of the permissions defined by the indicator. This might produce a very low value if the indicator consists of only low risk permissions or if the user preference on permissions in the combination are positive, and hence, may not follow the definition of risk indicators very well. To counteract this, the average value is normalised to a number between 5.0 and 10.0. The minimum score is then moved up to the exact middle of initial low risk and initial high risk.

Summarising the permission score and the risk indicator score gives us the raw privacy score:

$$raw\_privacy\_score(x) = permission\_score(x) + risk\_indicator\_score(x) \quad (3.4)$$

By applying this score to our application test set (described in Section 3.5.2), we found that most applications were in the range between 0 to 100, but also that some were "off the chart" by having values up to 260. This was expected due to the fact that the raw privacy score is a linear function without limits. We wanted all applications to be in the range 0 to 100 to give users of our application an intuitive way of knowing what is worst and what is best for their privacy. By putting all application scores within a range, we provide them with a better way of comparing applications. We also wanted applications in the mid-range to be easier to affect by replying to our permission facts (see Section 3.5.3) because they are in a "void" of uncertainty, not being classified as either good or bad. We wanted to make it easy for the user to create his/her incentives to keep or uninstall a mid-range application by nudge them up to high risk or down to low risk based on his/her preferences. To achieve the mentioned factors, we run the raw privacy score through a general logistic function (see Equation 3.5) fitted with parameters to match our test set of 150 applications.

$$final\_privacy\_score(x) = \frac{L}{1 + e^{-k(x-x_0)}} \quad (3.5)$$

where:  $x$  = the raw privacy score of application of an application  
 $e$  = the natural logarithm base  
 $x_0$  = the  $x$ -value of the sigmoid's midpoint  
 $L$  = the curve's maximum value  
 $k$  = the steepness of the curve

Figure D.45 gives a visual representation of our privacy score applied to the application test set described in Section 3.5.2. The curve steepness,  $k$ , are set to be very steep from the start making it easy to nudge low risk applications up to a higher risk level. Further,  $x_0$  are set so that no applications with more than

one high-risk permission should be classified as low risk. The high threshold is set to 60, which was found to be where many applications started to violate our risk indicators.

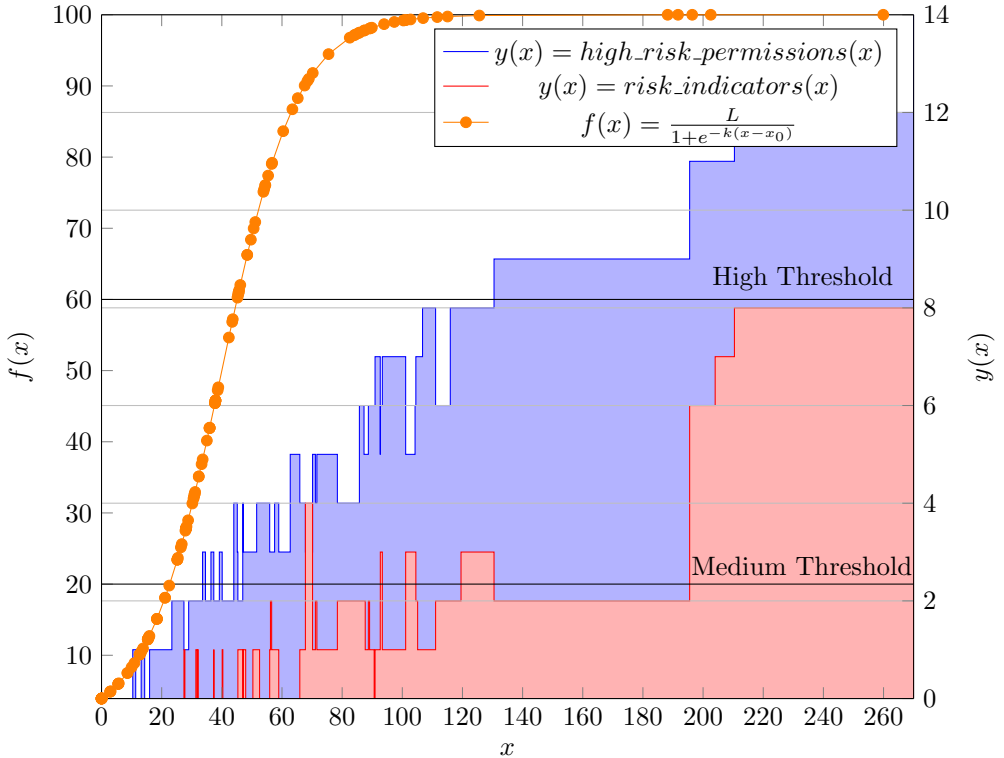


Figure 3.4: The figure illustrates the final privacy score  $f(x)$  where  $L = 100.0$ ,  $x_0 = 35.0$  and  $k = 0.08$  compared to the number of high risk permissions and risk indicators on the  $y$ -axis as a function of the raw privacy score ( $x$ ).

### 3.6 User Behaviour Analysis

Actions a user performs on the his/her device has the potential to reveal weaknesses in knowledge, and/or unfortunate habits related to privacy. By capturing key events, we could be able to categorise our users by attitude towards privacy and provide them with customised information and feedback to adjust negative

behaviour.

### 3.6.1 Experimental Plan

A user behaviour analysis of data collected locally on the device could help identify weaknesses in behaviour and map the current privacy awareness of the user. We envisioned that these behavioural elements could be used to place users in one of three categories: privacy fundamentalists, pragmatic majority or marginally concerned (see Section 2.3.2 for definitions). Each of these categories would entail customised content in the application. For privacy fundamentalists, who are already highly concerned about privacy, information about applications would be provided in an objective manner, not emphasising threats but providing insight to the applications' functionality. For the marginally concerned, this information would have a higher focus on informing about potential threats, comprehension, and education. Our aim is to guide both categories towards the pragmatic majority. Our experiment was conducted to answer the following experimental question:

1. Which actions made by the user in an Android application can be utilised to reveal privacy-related behaviour?

### 3.6.2 Experimental Setup

In this experiment we used elements from our research on users' relationship to privacy (see Section 2.3) to help classify our users. Otherwise, this experiment were based on finding ways to detect user behaviour on the device, and examine if it may be utilised to find weaknesses related to privacy.

### 3.6.3 Experimental Results

To establish a foundation for being able to properly categorise a user, we needed to create parameters which would reflect important aspect about the user's attitude towards privacy. These parameters would reflect the current attitude towards privacy, and would dynamically shift when behaviour contradicting them were detected. We recognised privacy awareness, privacy concern, and how much the user value utility over privacy as good candidates. We established a baseline for these parameters by developing a dialogue-view in the application containing questions which would be prompted to user the first time it were used. The questions were designed so the answers could easily be quantified and adopted into our parameters. Further, we investigated the possibility of presenting users with an end user license agreement (EULA) of our application, and measure the time the user spent reading it before accepting. This would provide us with insight

to the user's concern about privacy, revealing if it were read or not. We also examined the value of detecting if users visits key elements of the application. If a user never visits our application's view for third-party application permissions, it would suggest that the user has little concern towards privacy. We further found that the system broadcasts intents whenever an application is installed, updated or deleted. These intents can be captured by registering a broadcast receiver with the filters `PACKAGE_ADDED` and `PACKAGE_FULLY_REMOVED`. Since every application is given a privacy risk score when installed, a user which retains an application installed even after reviewing its privacy score could indicate a lack of concern to towards privacy.

Except the initial questions and the installation events, we found the other elements to only provide vague indications of a user's attitude towards privacy, and it would be hard to detect significant changes in user's awareness of privacy. This approach would also be hard to justify in explanations, as it would probably be considered as "creepy" by the user. However, we find our permission descriptions (see Section 3.3) and information about potential threats from combinations of permissions (see Section 3.4) to contain information which is suitable for every user category. The remainder of this section includes the features in which experiments shows successful results.

Detecting installations of new applications could be used to search for trends in the user's application install pattern. To correctly estimate a change in the trend, we use previous history and compare it to the recently installed application. We believe that the average user installs, at the most, 10 applications per month. Since we want the trend-analysis to reflect the user's current privacy awareness, we use only the last 10 installed applications. The trend will thus not get affected by outdated behaviour. This approach is inspired by the principles of Kalman Filtering, which recursively use measurements over time to provide better estimates of the underlying system state and future results. The privacy score calculated for each application at install-time will serve as our data basis. Whenever a new application is installed, the average privacy score of the last 10 installed applications is calculated and compared to the history of earlier calculations. If the new average is over our defined threshold for high-risk applications (an average risk score higher than 60 of 100 possible), and has increased since the last average was calculated, we conclude that the user shows a negative attitude towards privacy continuing to install high-risk applications. The user must therefore be notified about the increasing amount of potentially dangerous applications on his/hers device. If a trend averages above our threshold for dangerous applications, but is not increasing, it is interpreted as negative.

Answers collected on permission preferences could be analysed to detect disharmony. We define disharmony as the unbalance in users' preferences that occur when negative feedback is given on a permission (collected to create permission

---

weights, see section 3.5) and the application which owns the permission is kept installed on the device. A high presence of negative feedback in application-preferences strengthens the evidence of disharmony. This may also be used to indicate how users value utility over privacy, which defines the willingness to keep applications that disclose information installed if they provide high levels of utility or entertainment value.





## Chapter 4

# Android Watchdog Application

*This chapter will present our proposed solution to the thesis goal - an Android application named Android Watchdog. We will present our vision for the application, followed by an explanation of its architecture. Further, we elaborate around the graphical user interface and how principles for Explanation-Aware Computing is implemented. Finally, the architecture components and activities are outlined in more detail.*

### 4.1 Our Vision

Our vision for Android Watchdog is to provide a simple tool for users to review the functionality and privacy-threats in third-party applications installed on their Android device. The application conducts a static analysis of applications based on the permissions they require. It does not detect or prevent real-time events such as sensor usage or access to sensitive information. Instead, we have emphasised the importance of creating comprehensive explanations of permissions and permission patterns to ensure that users understand the risks an application constitutes to their privacy. We have also developed explanations of our application's behaviour to achieve transparency and thus leading to users trusting our application.

In order to provide a more accurate analysis of third-party applications, we have implemented a mechanism for users to report their opinions on application functionality. This mechanism is developed to compensate for the lack of meta-data available about applications for our analysis, and it also forms a rec-

ommender that encourages users to review and/or remove applications on their device.

Figure 4.1 below shows a simplified model of the underlying application architecture. Activities represent important screens in the application and Utility-Components provide the functionality and data basis. Each of the components and activities are described in more detail in Sections 4.4 and 4.5.

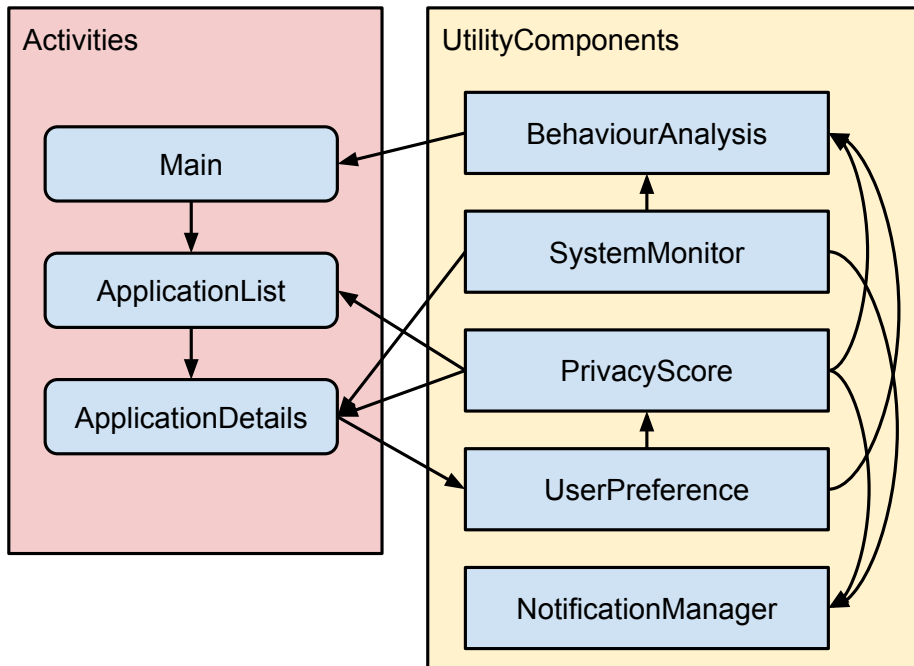


Figure 4.1: Overview of The Android Watchdog architecture. Activities are different screens in the application where the user interface is created. Utility-Components are external components that implement research done in Chapter 3.

## 4.2 Graphical User Interface

An important focus when creating the graphical user interface was to increase the usability of the application. This directly influences the user's confidence and trust in the application [19]. We also wanted the user to immediately understand the intention of our application. The Android Watchdog application is

designed following primarily the design guidelines for Android Material Design [26]. The most important graphical element is the risk factor combined with the corresponding colour code. This is the result of our analysis of the application, and the most visible signal the to attract the user's attention. Further, the application follows a light blue and white theme, with the occurring red, yellow and green colours indicating risk factors. Text elements are coloured either black or light grey, depending on the importance of the text. Text coloured black are elements that we want to direct the users attention to while grey text are often used for explanations or descriptions as an extension to an important component.

### 4.3 Explaining Application Behaviour

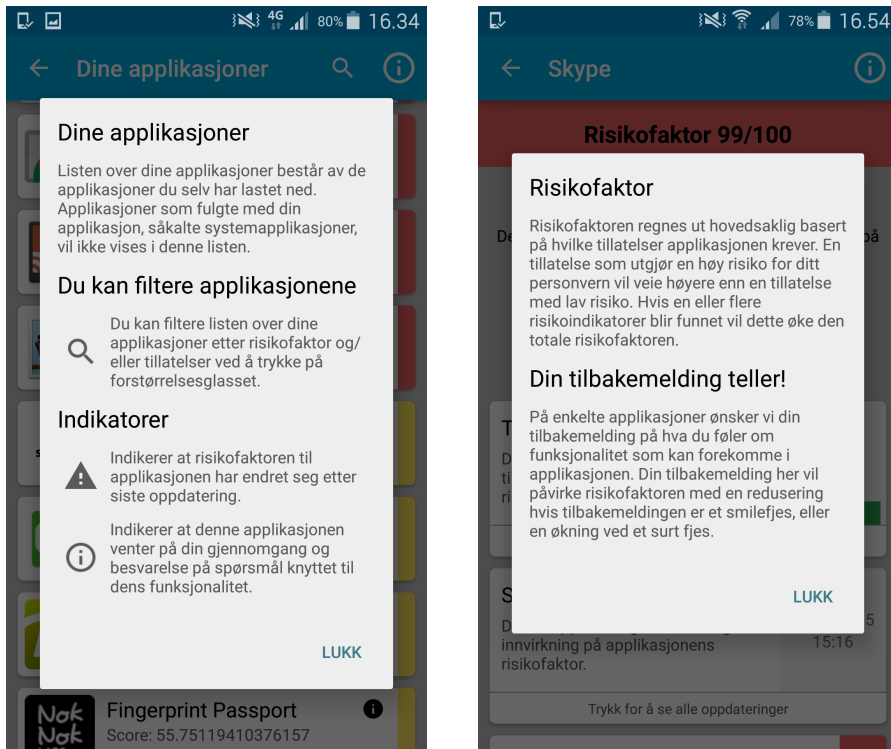
Explanations are a good way to provide information to users (see Section 2.1.7). Every activity in the application contains an information dialogue, which can be found in the top action bar. This dialogue contains detailed information about every component in the current activity (see Figure 4.2a). These explanations ensure that the user understand how results are calculated, and the function of components in the activity. If we label an application as a possible risk to the user, a justifying explanation of the underlying data for this is presented (see Figure 4.2b). To ensure that the information provided becomes valuable for even novice users, technical terms are avoided where possible. We have written new explanations to every permission to assure proper understanding of the functionality and the threats involved in accepting them (see Section 3.3). These new descriptions are written in a manner that is simple and short, but uses more precise words, in comparison to the ones that belong to the operating system itself. A selection of different explanations found in the application can be seen in Figures 4.3b, 4.2a, 4.2b, 4.7b, and 4.9b.

### 4.4 Implementing our Research in UtilityComponents

The following section contains descriptions of the different UtilityComponents in Android Watchdog. A UtilityComponents main concern is to provide important information to application activities.

#### 4.4.1 BehaviourAnalysis

The BehaviourAnalysis-component is used to generate information about the users' behaviour and actions.



(a) Explanations implemented in the ApplicationList-activity.

(b) Explanations implemented in the ApplicationDetail-activity.

Figure 4.2: Illustration of explanations in the ApplicationList-activity and the ApplicationDetail-activity.

## Trends

The trend analysis accumulates data from the SystemMonitor-component (see Section 4.4.2) about the recently installed and uninstalled applications to identify trends (explained in Section 3.6). It further uses an application's risk score, provided by the PrivacyScore-component as the basis for its calculations. The analysis uses the installation history of the ten latest installations for estimations, is meant as a reflection of the user's either improved or worsened concern towards privacy. How the trends are visualised is presented in Section 4.5.1.

## Disharmony and Utility-Privacy Feedback

The disharmony and utility-privacy feature uses data from the `UserPreference`-component (see Section 4.4.4) to search for negative attitudes towards application permissions. If a user has shown dissatisfaction towards one or more permission, we want to encourage the user to review the application. This is not just a detection of an unbalance between the user's preferences and the application, but it also reflects how the user values utility over privacy. How the disharmony is visualised is presented in 4.5.1.

### 4.4.2 SystemMonitor

The `SystemMonitor`-component receives information from the Android operating system and is used to detect installations, removals and updates of applications. Whenever a third-party application is installed or removed, this component receives a broadcast from the system. The `SystemMonitor` receive this broadcast and sends this data to the `BehaviourAnalysis`-component. Further, it detects when an application receives an update from Google Play Store and if any permissions were removed or added. This information is pushed to the `NotificationManager`.

### 4.4.3 PrivacyScore

The `PrivacyScore`-component has two main functions: (1) analyse user feedback on application functionality and calculate permission weights (see Section 3.5.3), and (2) calculate the privacy score for an application based on which permissions it requires and which risk indicators (see Section 3.5.3) it violates. User feedback are gathered from the `UserPreference`-component described in Section 4.4.4. The final score is calculated by summarising all permissions multiplied with its permission weight and risk indicators multiplied by the combined permission weight as described in Section 3.5.3.

### 4.4.4 UserPreferences

This component collects data about the user's attitude towards applications' permissions in the `ApplicationDetail`-activity. Further, it stores the replies and make them available for both the `BehaviorAnalysis`-component and the `PrivacyScore`-component to use.

### 4.4.5 NotificationManager

The NotificationManager creates and posts a notification to the Android status bar (see Figure 4.8b) if an update to a third-party application affected its privacy risk score. The notification provides the user of our application with a notice that his/her immediate attention is required. A touch on the notification launches the ApplicationDetail-activity of the updated application.

## 4.5 Application Activities

This section will give a brief description of the different activities in Android Watchdog.

### 4.5.1 Main

The Main-activity (see Figure 4.3a) is intended to provide an overview of the user's current privacy state. It provides a dashboard that shows important information from the BehaviourAnalysis-component and the SystemMonitor-component. The activity consists of an overall privacy risk score, which is the average of the risks scores awarded to all third-party applications. Further, the user is presented with the following elements containing both textual and graphical information:

#### **Overall Threat**

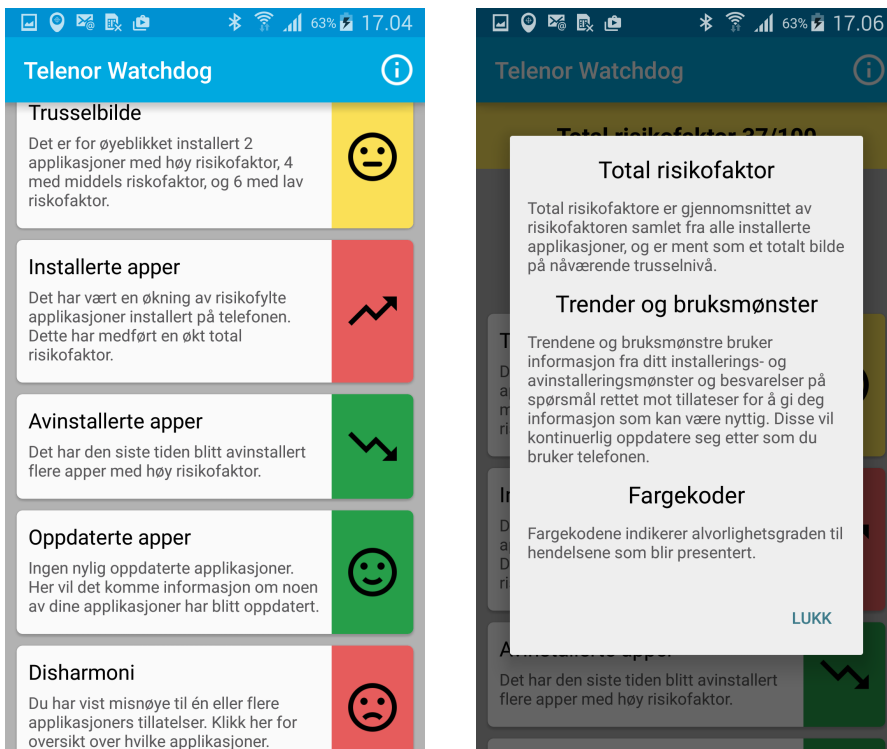
This card sums up the overall threat based on all the current installed applications. The card provides a textual summary of how many applications lies within each privacy threat level. Based on the severity of the overall risk involved, the colour code and the smiley face adjust accordingly.

#### **Install Trend**

Using the BehaviourAnalysis-component, this card shows trends in the user's latest application installations. A downward, upward or straight arrow indicates an increasing, decreasing, steady high or steady low risk-factor trend. A neutral smiley is shown if there are no valuable information to present. The colour code indicates the severity.

#### **Uninstall Trend**

Using the BehaviourAnalysis-component, this card shows trends in the users' latest application removals. A downward or straight arrow indicates an increasing or steady high risk-factor trend in the latest uninstalled applications. A neutral smiley is shown if there are no valuable information to present. The color code indicates the severity.



(a) The Main-activity of our application. Here, the user are made aware of good and/or bad habits concerning his/her applications.

(b) Explanations implemented in the Main-activity.

Figure 4.3: Illustration of the applications Main-activity.

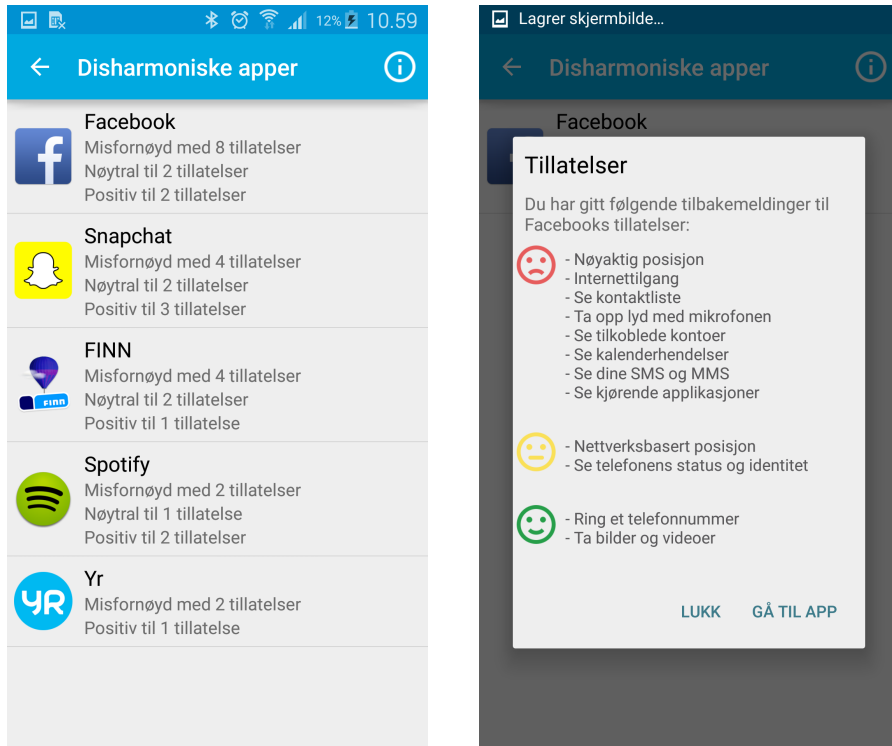
### Updated Applications

This card show information about the latest updated applications. If any applications have received any recent updates, it will be presented here to encourage the user to review it. A neutral face is shown if there are no valuable information to present.

### Disharmony

This card uses data from the BehaviourAnalysis-component and displays information about disharmony in user's preferences towards installed applications. If a user have shown dissatisfaction towards several of an application's permissions but still has it installed on his/her device, a warning

notification will be present in this card. If touched, the user will be directed to an activity that enables for review of his/her preferences (See figures 4.4a and 4.4b).



(a) Screenshot of the activity for reviewing disharmonic applications. This activity is accessed by touching the card for disharmony in the Main-activity. It provides an overview of the user's preferences towards applications.

(b) Screenshot of the dialogue for reviewing specific disharmonic permissions. This dialogue is presented when clicking on an application in the activity for disharmonic applications.

Figure 4.4: Illustration of how disharmonious applications are presented in the application.



## 4.5.2 ApplicationList

The ApplicationList-activity (see Figure 4.5a) makes it easy for the user to overview installed third-party applications. Applications are presented in two lists where the first contains the most recently updated applications, and the second contains all other applications. It is important to highlight newly updated applications because their risk to user privacy might have changed due to the latest update, and the user should be made aware of these changes. If an application's privacy risk score changed, a warning icon will show as an indicator.

As an incentive for users to explore applications, filtering of the application list is implemented (see Figure 4.5b). Applications can be filtered by privacy threat level and/or which permissions they require. This makes it easy for the user to find and review applications with certain functionality.

## 4.5.3 ApplicationDetail

The ApplicationDetail-activity (see Figures 4.6a and 4.6b) summarises important privacy information on a per application basis. It also gather user input to the UserPreference-component described in Section 3.5.3. Next, each activity element will be described in more detail:

### Required Permissions

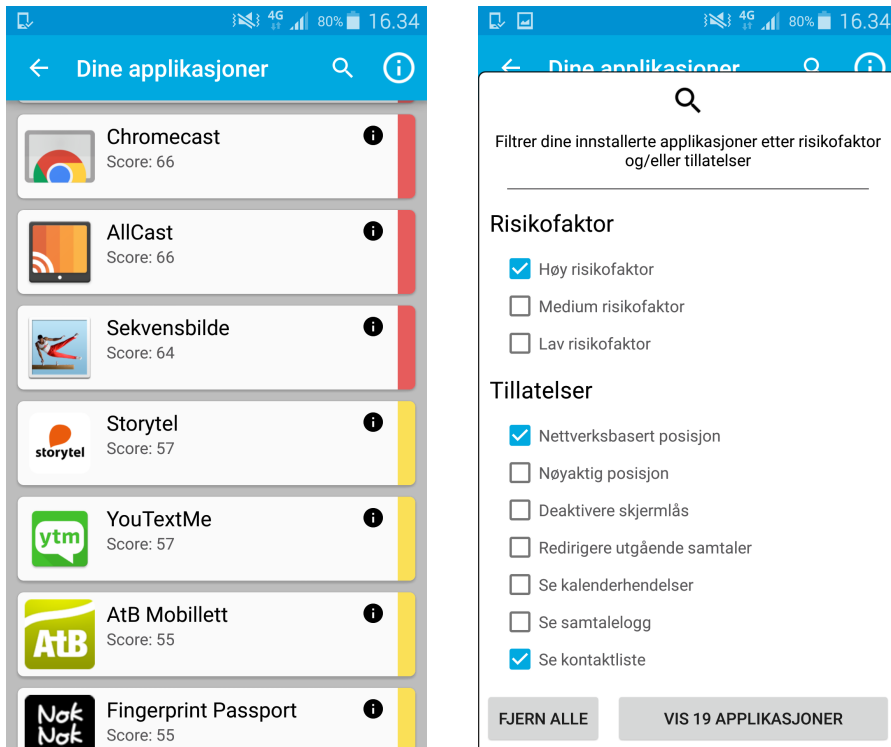
Informs about how many permissions this application requires and how many of them present a high privacy threat. It also visualises the distribution over permissions with high, medium and low privacy threat level required by the application. Touching the card opens a new activity for detailed description of each permission required by the application (see Figure 4.7a). The information presented in this activity is a result of our experiment on creating comprehensive descriptions of permissions (see Section 3.3).

### Application Updates

Informs about the impact of the latest application update had on the privacy risk score. Touching the card opens a new activity for detailed information about every application update (see Figure 4.8a). It shows which permissions was added and which was removed. It also shows the privacy risk score for each update, helping the user decide whether the developers are taking privacy into consideration when developing the application.

### Risk Indicators

Informs about how many risk indicators the application violates. Touching the card opens a new activity for detailed information about each violated



(a) The ApplicationList-activity of our application. The user are presented with all installed third-party applications on his/her phone and initially made aware of their privacy risk with colour codes.

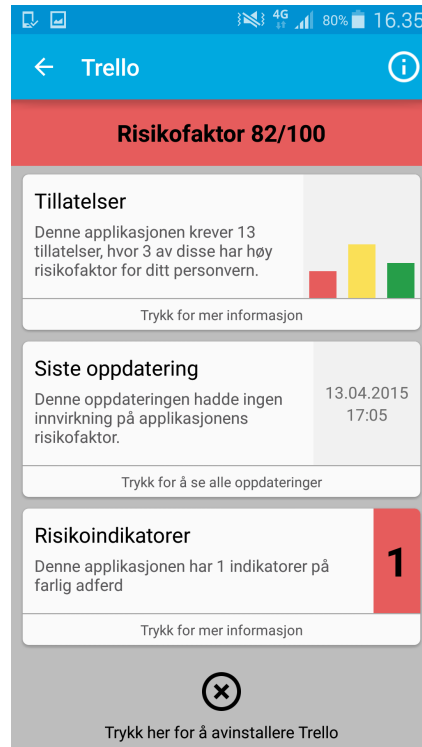
(b) In the ApplicationList-activity, filtering of applications based on threat level or required permissions are implemented.

Figure 4.5: Illustration of the ApplicationList-activity.

risk indicator (see Figure 4.9a). These indicators are a result of our experiments on permission patterns for detecting malicious behaviour (see Section 3.4).



(a) The top half of ApplicationDetail-activity. Users are presented with facts about permissions and asked how they feel about it. This is input to our user preferences system, described in Section 3.5.3.



(b) The bottom half of ApplicationDetail-activity. Users are presented with information about how the applications impact their privacy and a shortcut to uninstall the application.

Figure 4.6: Illustration of the ApplicationDetail-activity.

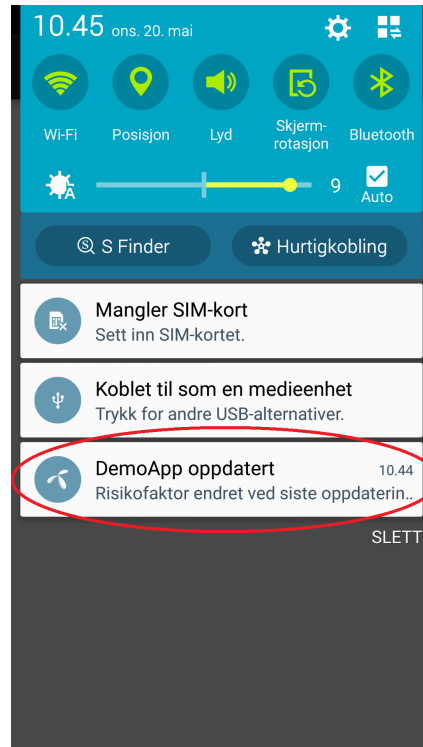
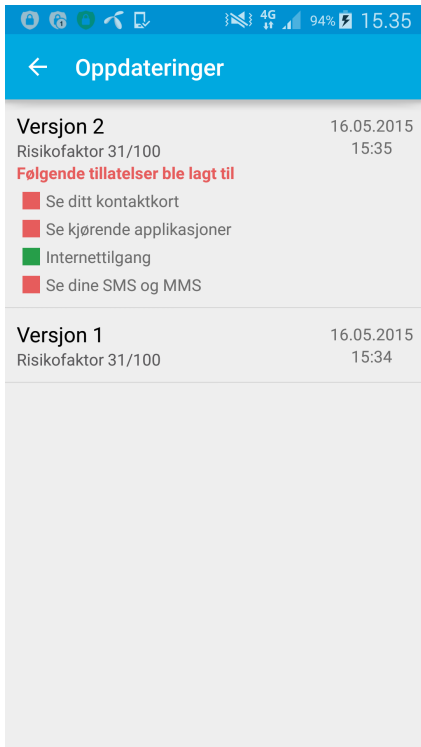


(a) This activity is shown when the user touches the card for required permissions in `ApplicationDetail`-activity. Users are presented with description of permissions (see Section 3.3.3) requested by the application.



(b) Explanations implemented in the activity explained in Figure 4.7a.

Figure 4.7: Illustration of the activity that is presented when the user touches the card for required permissions in `ApplicationDetail`-activity.



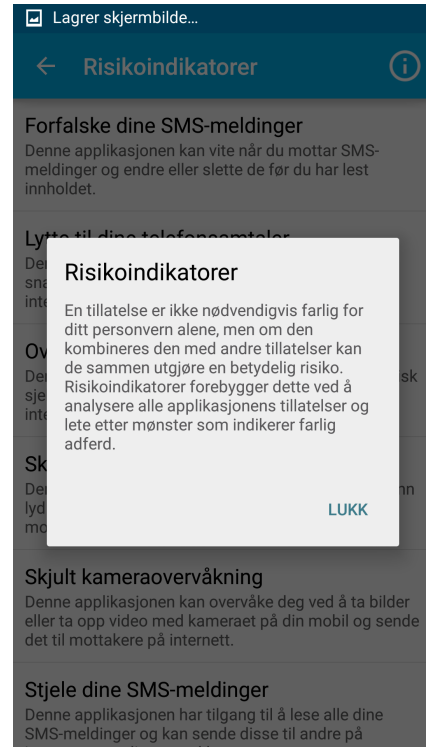
(a) This activity is shown when the user touches the card for application updates in ApplicationDetail-activity. How the latest application update affected the privacy threat level is visualised.

(b) A status bar notification is shown if an application update affects the application's privacy risk score.

Figure 4.8: Illustration of the activity that is presented when the user touches the card for application updates in ApplicationDetail-activity and how status bar notifications are implemented.



(a) This activity is shown when the user touches the card for risk indicators in ApplicationDetail-activity. Users are presented with detailed information regarding risk indicators (see Section 3.4) in the application.



(b) Explanations implemented in the activity explained in Figure 4.9a.

Figure 4.9: Illustration of the activity that is presented when the user touches the card for risk indicators in ApplicationDetail-activity.

# Chapter 5

## Application Evaluation

*This chapter contains the evaluation of our application. First, we will present our plan and hypothesis' for the evaluation. Second, the setup used and subject distribution are laid down. Finally, results from the different evaluations are presented and thoroughly analysed, followed by identified limitations.*

### 5.1 Evaluation Plan

The evaluation goal is to measure both changes in users' attitude towards privacy when using our application and the user acceptance of our application. We aim to recruit 20 subjects without or with a limited technical background. This group will probably serve our purpose because we want to investigate the impact of our research on "normal" people not daily surrounded by new technology. The following procedure defines our evaluation:

1. Recruit 20 subjects with Android devices from NTNU, Campus Dragvoll. There will be a 500 NOK price awarded to a randomly chosen participant after the evaluation is finished.
2. The students will reply to an initial questionnaire designed to measure their attitude towards privacy (see Appendix C).
3. After finishing (2), the subjects will install Android Watchdog on their Android devices and use the application for five days.
4. After finishing (3), the subjects will reply to a questionnaire equal to (2) to see if their attitude towards privacy has changed after using Android Watchdog.

5. Finally, the subjects will reply to a Mobile Services Acceptance Model (MTAM) Questionnaire (see Section 2.1.5 and Appendix B) to measure the user acceptance of our application.

Our application will be equipped with Google Analytics when being used by the evaluation subjects. This will enable us to also evaluate application usage statistics.

## 5.2 Evaluation Hypothesis

This section will present our hypothesis' for the different evaluations of our application.

### 5.2.1 Privacy Survey Hypothesis

We have developed the following null hypothesis for the privacy survey. This serves as the basis for our later calculation of the paired t-test.

*H<sub>0</sub>: The use of our application will not change evaluation participants' attitude towards privacy.*

### 5.2.2 Application Usage Hypothesis

The hypothesis below is our way to conclude on subjects engagement in the survey, and will be tested against the unique user count and usage statistics from Google Analytics.

*H<sub>1</sub>: The Watchdog application will be used by all evaluation participants during the evaluation period.*

### 5.2.3 Mobile Services Acceptance Model Hypothesis

This section presents MTAM hypothesis developed by Gao et al. [17]. Each hypothesis will be connected to its respective construct in MTAM.

#### Context

Android applications can be used in a large variety of different situations. Users' perception on mobile services varies in different contexts.

*H<sub>1</sub>: The appropriate context has a direct positive effect on Perceived Usefulness of mobile services.*

*H<sub>2</sub>: The appropriate context has a direct positive effect on Perceived Ease of Use of mobile services.*



### Personal Initiatives and Characteristics

Individual characteristics, such as curiosity and perceived enjoyment, may strongly enhance users' perception of mobile services. Personal initiative is depending on whether the individual has previous experience with or knowledge about mobile services.

*H<sub>4</sub>: Personal Initiative and Characteristics have a direct positive effect on Intention to Use of mobile services.*

### Trust

Users may feel threatened when the technology has the capability to track actions and store personal information outside the control of its users.

*H<sub>5</sub>: Trust has a direct positive effect on Intention to Use mobile services.*

### Perceived usefulness

Productivity, performance, and effectiveness affect how users find the technology useful.

*H<sub>6</sub>: Perceived usefulness has a direct positive impact on intention to adopt mobile services.*

### Perceived Ease of Use

The degree to which the prospective user expects the target system to be free of effort.

*H<sub>3</sub>: Perceived Ease of Use of mobile services has a direct positive effect on Perceived Usefulness of mobile services.*

*H<sub>7</sub>: Perceived ease of use has a direct positive impact on intention to use mobile service.*

## 5.3 Evaluation Setup

The evaluation process will include the following setup:

- Questionnaires developed using Google Forms.
- Subjects with the distribution shown in Table 5.1.
- SmartPLS to analyse MTAM results.
- Paired t-test to analyse privacy survey results.
- Google Analytics for application usage statistics.

For privacy reasons, we distributed ID-number to all test subjects and had these connected to their e-mail account at a separate location. The application was distributed after all subjects had submitted the initial privacy survey. The subjects received the application per e-mail and were given a description of the installation process and short notice declaring how we intended them to use our application.

<b>Distribution</b>	<b>Amount</b>	<b>Percent</b>
<b>Gender</b>		
Male	14	70
Female	6	30
<b>Age</b>		
20-25	17	85
Over 25	3	15
<b>Department</b>		
Science or Engineering	3	15
Other	17	85

Table 5.1: The table show different distributions over the subjects participating in our evaluation. Most of the subject were recruited from NTNU, Campus Dragvoll.

## 5.4 Privacy Survey Results

In this section, we will present key results from the privacy survey, and analyse changes in attitude towards privacy after using our application. Referred questions can be found in Appendix C and their results in Appendix D. The complete list of results for the paired t-test can be found in Appendix D.2. The limitations of this survey are discussed in Section 5.7.

Our initial privacy survey show that among our evaluation subjects, over 60% claim to be some or very concerned about privacy (see Figure 5.1). On question about their level of knowledge about personal information collected by third-party applications from their Android device, over 60% state that they have little to no control over this element (see Figure 5.2). Already at this point, we see some evidence of our underlying theory that the users that claim to be concerned about privacy, in fact, does little or no effort to maintain it. On specific questions on what information and data they are comfortable with sharing (Questions 19-31, see Figures D.19 to D.31 on pages 136 to 142), they show high dissatisfaction towards most of the elements. Further, subjects state that they will keep applications installed if they provide high utility value (Question 4, see

Figure D.4 on page 129), while also stating high agreement to the suspicion that companies misuse the collected data (Question 33, see Figure D.33 on page 143). This indicates inconsistency in attitude and poor privacy-related judgement.

When examining and comparing the results from both surveys, we conclude that the majority of answers have shifted towards a more positive trend after using our application. The results from these question can viewed in context with results from our MTAM survey on perceived usefulness (see Section 5.6). Our MTAM provides good indications of having increased the users ability to maintain privacy and made them more aware of privacy threats. Some of the more interesting results that can be highlighted from the survey are described below.

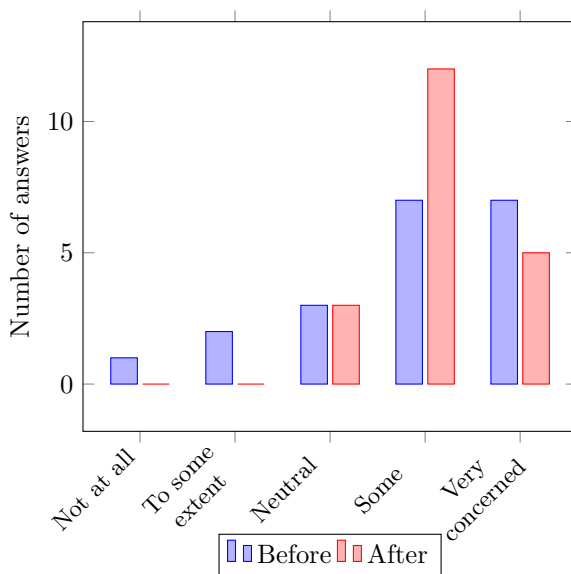


Figure 5.1: The figure show how our evaluation subjects responded when asked to grad how concerned they are with their Privacy. Further details can be found in Section D.2, question number 1.

Figure 5.1 show an increase in privacy concern among our participants. We can see from our initial survey that there were some subjects that stated little or no concern at all towards privacy. The second survey shows no answers in these categories, but a significant spike in the category "some". This may indicate that the information provided in our application has impacted the users in some way, leading to a change in concern for privacy. This can be further supported by the increased suspicion towards Android applications our participants reports in

Question 3 (see Figure D.3 on page 128).

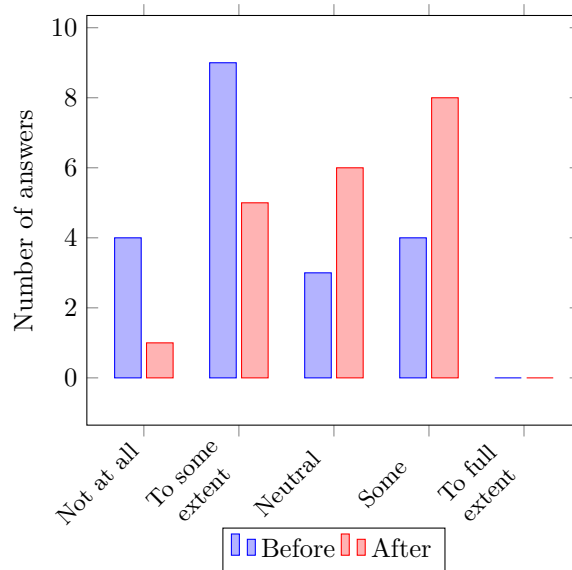


Figure 5.2: The figure show how our evaluation subjects responded when asked if the knew to what extent third-party applications collect and use their personal information. Further details can be found in Section D.2, question number 2.

Figure 5.2 shows a significant change in users' claimed knowledge about how much personal information third-party applications collect from their device. We can see a halving of the number of participants in the categories for "not at all" and "to some extent". Further, a doubling in the category "some" has occurred. This may indicate that our permission descriptions have lead to an increased understanding of the possible disclosure of information from third-party applications.

Another interesting observation is the loss of trust towards social media among our participants. We know social media applications, for example, Facebook and Instagram, to require a substantial amount of permissions. They have also been known to receive a high risk-score in our application. Figure 5.3 indicates that some of our participants were not aware of to what extent these applications can collect personal information, and show more scepticism towards them after using our application. Question 43 (see Figure D.43 on page 148) asks about the willingness to share data with social media. Also here we can see an improvement with a doubling in the category "unwilling".

Our statistical analysis from the paired t-test shows that only five questions

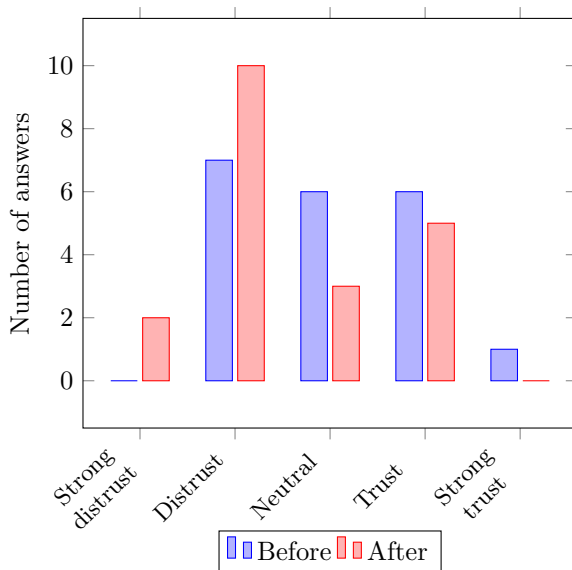


Figure 5.3: The figure show how our evaluation subjects responded when asked to grade how they trust Social Media Applications. Further details can be found in Section D.2, question number 39.

(Questions 2, 28, 37, 43, and 44) contain valid results with significance level below 0.05 (see Appendix D.2 for the complete paired t-test analysis). Question 2 (see Figure 5.2) is the most significant of these and gives us confidence that our application, in fact, provides information that leads to increased overview of potential threats by our users. Based on this analysis we can not prove that our application has enough significant evidence to discard the null hypothesis. The low amount of significant results may indicate that the paired t-test model requires a more significantly correlated dataset and may not have been suitable for our evaluation. This could be due to participants not providing an accurate answer and have replied with a certain amount of randomness, leading to uneven variations among the paired differences. However, based on the several indications described above, we choose to conclude that our application have had a positive influence on our subjects.

## 5.5 Application Usage Results

Table 5.2 presents data collected from Google Analytics in the period of our evaluation. `ApplicationListActivity` (see Section 4.5.2) are not surprisingly visited the most. This activity gives an overview over all installed third-party applications on the subjects Android device sorted after risk score. `ApplicationDetailActivity` (see Section 4.5.3) is the second most visited activity. This is a positive trend because the two activities are closely related. It shows that users frequently wish to get more details about the privacy risks connected to applications after inspecting them in `ApplicationListActivity`. `ApplicationDetail.RequiredPermissionsActivity`, `ApplicationDetail.RiskIndicatorsActivity`, and `ApplicationDetail.ApplicationUpdatesActivity` (all described in Section 4.5.3) are all activities that branch out from `ApplicationDetailActivity`. They present an even more detailed view of present risk factors. These activities are not visited as frequently. `ApplicationDetail.ApplicationUpdatesActivity` has few views. This is probably because few applications were updated during the relatively short evaluation time. The low view count of `ApplicationDetail.RequiredPermissionsActivity` and `ApplicationDetail.RiskIndicatorsActivity` can indicate that the application summary given in `ApplicationDetailActivity` is enough for most users to get an understanding of the applications risk to their privacy. It also naturally a bit lower because many applications do not have violated risk indicators (see Section 3.4) and do not require permissions. The low amount of views for `Main.DisharmonyActivity` tells us that users may need to receive stronger encouragement to use this functionality.

Activity Name	View Count
<code>ApplicationListActivity</code>	357
<code>ApplicationDetailActivity</code>	348
<code>MainActivity</code>	225
<code>ApplicationDetail.RequiredPermissionsActivity</code>	83
<code>ApplicationDetail.RiskIndicatorsActivity</code>	46
<code>ApplicationDetail.ApplicationUpdatesActivity</code>	15
<code>Main.DisharmonyActivity</code>	5
<b>Total</b>	<b>1079</b>

Table 5.2: The table show the number of views for each activity in our application during the evaluation period. The data is collected from Google Analytics. Not surprisingly, the main views described in application architecture (see Figure 4.1) got most views. These views show general information about applications.

Further, the application usage result shows that all subjects in our evaluation used the application one or more time during the evaluation time.

## 5.6 Mobile Services Acceptance Model Results

The following sections will discuss results from our MTAM questionnaire both in a descriptive manner, and in terms of data analysis using SmartPLS.

### 5.6.1 Descriptive Results

This section present results obtained from the MTAM questionnaire by inspecting the values in Table 5.3. For each question in the MTAM questionnaire, subjects were asked to answer on a scale from 1 to 7, 7 being highly positive towards the question.

The construct Perceived Usefulness is based on the grade users perceive our application to raise their ability to maintain personal privacy. This includes understanding of potential threats and limiting disclosure of information. Our results show mean values close to 5 on all questions in this category. Especially, Question 2 shows that the subjects have great confidence in our application, making them more aware of threats from third-party applications installed on their device. This indicates that our experiments on creating new permission descriptions (see Section 3.3) have been successful. Perceived Usefulness might be affected by our subject distribution. Students with a non-technical studies, the major group in our study, will probably grade our application to be more useful than students with a technical background.

The reported trust in our application is high. We believe this to be mainly because of our extensive work on having transparency, justification, and conceptual explanations present (see Section 4.3). Our subjects report that they understand the purpose of the application, that the information presented is valid, and most importantly, that it does not violate their privacy and that it is a low risk connected with using it. These values prove both high cognition-based trust and affect-based trust, and hence, result in a high overall perceived trust as explained in Section 2.1.4.

A high value on question number 16 is interesting because it states that users of our applications tend to use it when they are downloading new applications. This answers directly to our research question number two (see Section 1.2) because it indicates that our application supplies more useful information than Google Play Store presents when new applications are installed. Hence, it is used as a tool to gather more information about applications and by that, has an educational effect on its users.

Further, we notice that our test subjects show a high level of agreement on Intention to Use, and indicate that they would download the application if it were available in Google Play Store. This supports the fact that there is a market for this kind of application. On an overall basis, our test subjects have shown great interest and positivity towards our application, and more than half of the answers

show mean values well above 5 (see Table 5.3). Based on this, we conclude our application to be useful for the group of people which participated in the survey.

Question Number	Average Answer
<b>Perceived Usefulness</b>	
1	4,74
2	5,74
3	4,47
<b>Perceived Ease of Use</b>	
4	5,53
5	5,37
6	5,79
<b>Trust</b>	
7	5,95
8	5,58
9	5,63
10	5,47
11	5,53
<b>Personal Initiatives and Characteristics</b>	
12	5,37
13	4,79
14	4,95
<b>Context</b>	
15	3,42
16	5,47
17	3,26
18	4,68
<b>Intention to Use</b>	
19	5,21
20	4,89

Table 5.3: The table show average response values for questions in our MTAM Questionnaire. Responses to each question are a numerical value in the range 1 to 7. Question numbers can be matched with the questions presented in Appendix B.

### 5.6.2 Data Analysis

To test the reliability and validity of each construct in our MTAM, Internal Consistency of Reliability of each construct was tested with Cronbach's Alpha



coefficient (see Table 5.4). The Cronbach's Alpha values range from 0.594 to 0.936. All the constructs except Context is above 0.7. The lower reliability for the construct Context can be partly attributed to the high number of measurement items. According to previous research by Robinson et al. [49], a reliability coefficient of 0.6 is marked as the lowest acceptable limit for Cronbach's Alpha for exploratory research.

<b>Construct</b>	<b>Cronbachs Alpha</b>
Context	0,594
Intention to Use	0,701
Perceived Ease of Use	0,847
Perceived Usefulness	0,783
Trust	0,936

Table 5.4: The table show Cronbach's Alpha value for each construct in our MTAM analysis. Cronbach's Alpha is a measurement for testing reliability and validity of each construct. A value of 0.6 is considered to be the lowest acceptable value [49].

For the purposes of testing the research hypotheses, Partial Least Squares (PLS) analysis was used. PLS allowed us to do a combined regression and principal components factor analysis within the same statistical technique. In this study, the collected data was analysed using the statistical software SmartPLS. Figure 5.4 presents the structural measurement model using the PLS algorithm. The number inside the nodes means  $R^2$  (R-square), which denotes the coefficient of determination.  $R^2$  provides a measure of how well future outcomes are likely to be predicted by the model, the amount of variability of a given construct. In our PLS analysis, the  $R^2$  coefficient of determination is a statistical measure of how well the regression coefficients approximate the real data point. Table 5.5 shows the path coefficients, which are standardised regression coefficients, generated from the PLS analysis, among other key values.

As the key values show, only one of our hypothesis, namely  $H_3$ , is supported with a significance level of 0.05. Adjusting this level up to 0.1 includes  $H_6$  as a supported hypothesis. We especially believe that  $H_4$  and  $H_5$  should have a greater impact on intention to use our application. They are probably insignificant due to the nature of our subject group. Most of them had a non-technical background and by that, do not have the natural personal initiative to use new technology. Further, our application requires a minimum of permissions (these are permissions required by Google Analytics) which might reduce trust as a factor when participants evaluated their acceptance of the application. Another possibility regarding the insignificance of trust is, as reported in Section 2.3, peo-

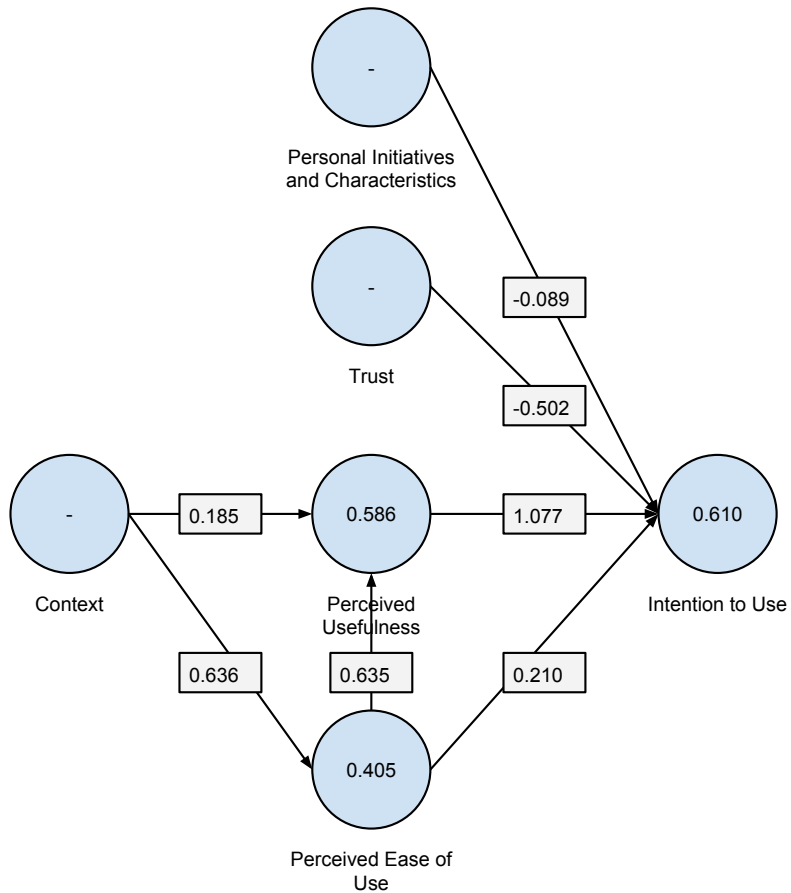


Figure 5.4: The figure show PLS analysis results of our MTAM questionnaire. Values in nodes are the  $R^2$  (R-square) value, while values on edges are the Path Coefficient between the connected nodes. The figure should be seen in comparison with Table 5.5.

ple do not understand the means Android provide them with to evaluate the dangerous functionality of third-party applications.

Perceived Usefulness proves to be the most important construct for users intention to use our Android application ( $H_6$ ). This hypothesis is also significant at a 0.1 significance level. Further, Ease of Use prove to be an important factor for Perceived Usefulness ( $H_3$ ). The high weight on  $H_6$  is not surprising because

Hyphothesis	Path Coefficient	T-value	P-value
$H_1$ : Context $\rightarrow$ Perceived Usefulness	0,185	0,607	0,544
$H_2$ : Context $\rightarrow$ Perceived Ease of Use	0,636	1,234	0,218
$H_3$ : Perceived Ease of Use $\rightarrow$ Perceived Usefulness	0,635	3,234	0,001
$H_4$ : Personal Initiative and Characteristics $\rightarrow$ Intention to Use	-0,089	0,211	0,833
$H_5$ : Trust $\rightarrow$ Intention to Use	-0,502	0,972	0,331
$H_6$ : Perceived Usefulness $\rightarrow$ Intention to Use	1,077	1,854	0,064
$H_7$ : Perceived Ease of Use $\rightarrow$ Intention to Use	0,210	0,577	0,564

Table 5.5: The table show key values acquired by PLS analysis of our MTAM questionnaire. The table should be seen in comparison with Figure 5.4.

we have developed an informative an educational application inspired by research question number two (see Section 1.2), and usefulness of such applications need to be high for them to have any purpose.  $H_3$  is probably an significant factor partly because the large amount of applications in the same category on Google Play Store creates a high level of user expectation when it comes to basics such as ease of use. An application that is easy to use will also provide incentives to use it more regularly, hence, increasing its usefulness.

## 5.7 Evaluation Limitations

Students constitute a large group of people, but are not representative for the general population. We did not have enough participants to use our results as a statistical foundation. Thus, our results can only be used as indicators and not in context with people outside this group. Our findings may also be somewhat biased by the fact that we chose a group of people which probably would record high usefulness since they come from a non-technical background and the information provided in our application would in any case be educational and useful for them.

Since our surveys were taken online, we can not confirm that any of the submitted information is correct. From this, it follows that information collected in the surveys may include submissions of wrong data, either on an intentional or unintentional basis.

In the Application Usage study, recording of specific user actions was not performed. We could, for example, have recorded when users uninstalled a certain

application and also what privacy risk score was connected with this application. Doing so would have given us more accurate indicators on user behaviour when using our application. Further, the study was conducted over a period of five days. In retrospect, we do not think this is long enough for users of the application to fully comprehend the large amount of different information presented in the application. A longer period of use may also have revealed more significant changes in attitude towards privacy.

## Chapter 6

# Conclusion and Future Work

*This chapter starts with a discussion about our research approach and our proposed solution to this thesis' goal. Second, the answers we have found to our research questions are presented, followed by a list of contributions to the field. Finally, we present future work motivated by a paragraph containing our final remarks.*

### 6.1 Discussion

The following sections will include discussions where we wish to highlight factors that have influenced our research approach and proposed solution.

#### 6.1.1 Research Approach

Our literature study in Chapter 2 has revealed that the average user do not understand privacy to a level sufficient enough to preserve it, nor does the Android permission system work as originally intended. In our study of these subjects, we have become motivated to nudge people towards managing their privacy better. The permission descriptions provided by Google Play Store and the Android operating system has in several studies been proven to not be comprehensible and too coarse-grained. We considered this in our approach towards creating our application and also explain potential risks involved in accepting certain permissions. Not all applications are malicious, but we have chosen to take a subjective position for mainly two reasons. The first being that we want to formulate our information in a way that makes an impact on the user. We believe

that the lack of understanding, interest and concern towards privacy is because users do not receive enough information - information that gets the user concerned and interested in privacy. The second reason is that the application must achieve trust among its users. We believe there is a need for an application that works in the users' interest, not just providing a static and objective description of permissions. We want our users to be confident in our information and learn the real nature of Android permissions and the risks involved in accepting them. We believe this approach adds a new dimension to existing solutions. It might also cause users to lose trust in legitimate applications since users are not accustomed to this type of information. However, it is the real nature of the permission, whether a third-party application uses it for malicious purposes or not. This lays the basis for our approach and has guided us in our further development and choices made in our project process.

Initially, we wanted to create a privacy breach detection system for non-rooted Android devices. We did a series of experiments both trying to detect when phone sensors are being used (see Section 3.1), and to retrieve sensitive information protected by permissions (see Section 3.2). Results from these experiments show that extracting information using permissions are way easier than trying to detect when a third-party application are requesting this information. We discovered methods to detect when the use of GPS, microphone and camera occur, but we were not able to precisely identify which application was using it at the time of detection. By monitoring running applications and filter them based on the fact that using camera requires the camera permission, we were able to narrow the list of suspects down. This method could be further refined by saving statistics of running applications when the use of sensors occurs and then search for patterns. However, the detection mechanisms used are not fit for use by normal Android users. Using these mechanisms for detection might interfere with other third-party applications by either reducing their functionality or, in the worst case, grant them impossible to use. These approaches are fit for research purposes, but when developing an application to be distributed through Google Play Store, it may not be a good idea to make use of them as they most likely will be interpreted as malicious behaviour. Having a system signed application would make it possible to access a larger portion of the Android System API, making sensor usage detection and identifying application behaviour easier and more accurate. This is further explained in our proposed Future Work, under Section 6.5.3. Because these restrictions limit our initial goal of detecting third-party application behaviour, we changed focus to include education and informing users about potential risks, solely based on permissions. This focus has added more weight on providing accurate and understandable information, and will be discussed in the next section.

## 6.1.2 Android Watchdog Application

We have re-written all descriptions and evaluated threat level for every Android permission. Our justifying reasons for being able to do so has basis in our extensive literature study, our examination of the Android documentation, and our experiments. In Section 2.3.3 we present research that propose different approaches to overcome the problem that users do not understand permissions. Fang et al. [15] deduces that all the information a permission provides access to should be a part of its explanation. Following this approach would probably lead to explanations being too exhaustive to read, but we have kept it in mind when writing our permission descriptions. We have written our descriptions short, but in complete sentences, and in a language form more suitable for novice users. We also combine the explanation of the permissions natural functionality with its possible disclosure of information where it is suitable.

Another approach, found in a website article presenting "The 12 Most Abusive Android Permissions" [53], gives a threefold explanation of each permission. First, an objective explanation of the permission, presenting its natural function. Second, an example of how and what it may steal of information. Finally, a list of which types of applications that would normally require this permission (see Figure 6.1). This approach may have some advantages in comparison to our descriptions because our biased information might influence the user too much. It will also make it easier to separate legit and malicious permission functionality.

### 2. GPS Location

**What it's for:** It grants apps access to your exact location through the Global Positioning System (GPS) and other location sources like cell sites and Wi-Fi. Like network-based location, GPS location can also be used by app developers to gain profit from location-based ads.

**How it can be abused:** Malicious apps use it to load location-based attacks or malware.

**Apps that need this permission:** location apps, check-in apps, social media apps

Figure 6.1: The figure shows an example of threefold explanations of permission developed by Trendmicro [53]. We believe this approach may have advantages over our permissions when it comes to understandability.

When deciding the threat level for each permission (see Section 3.3.3), we found that some, by themselves, do not present a privacy risk, however, when reviewed in combination with other permissions, revealed threats. To cover this in our application, we created patterns of permissions that together pose a high level of threat. Inspired by the research performed by Enck et al. [14], several permission patterns were developed (see Section 3.5.3). Our approach is likely

less accurate because we do not use registered broadcast receivers and content providers in our patterns as our application do not have root privileges. However, our goal is to inform and educate rather than to detect, so the fact that our approach is not as accurate, do not limit its usability. We use these patterns to inform the users of our application that they need to see the list of required permissions as a whole, not treating a permission as an isolated unit, and thus helping them in reviewing privacy risks in third-party applications.

Our initial vision was to incorporate a dynamic risk factor, indicating what type of functionality could be implemented in an application based on its required permissions. However, this does not cover the fact that some applications have legitimate reasons to use certain permissions. For example, a map-based application would probably need to access user location. Further, we do not evaluate the developer of the application, number of downloads, user rating and description. All these factors present valuable information together with what permissions are required. As we use the Android Developer API as our basis for application development, this sort of context information is not available. As a step towards overcoming this challenge, we decided to capture users preferences on permissions by implementing permission weights. This was done to incorporate the users opinion together with our predefined permission threat level (explained in Section 3.5.3). Further, this led to the issue of choosing between letting the weights impact one application (which were our initial idea for the weight system), or every application having the specific permission. Letting a weight impact only one application's permission would have demanded that each weight had more impact on the privacy score of that application more. An approach like this may have led to ignorant users removing the value of our predefined weights. This approach might work in combination with a centralised server, providing statistics and decision support (see Future Work 6.5.1), but as a local feature on the device, we recognise more pitfalls than positive effects. Therefore, our chosen approach seeks to capture the users overall attitude towards permission functionality. However, if we detect disharmony in their answers towards specific application permissions, we notify and encourage them to review this inconsistency.

So far, we have discussed our permission descriptions and how we rank them by privacy risk. We also mentioned permission patterns as a way for the user to interpret required permissions as a whole, and our system for capturing the user preference of permissions. All these factors are taken into account when we calculate the total privacy risk score for each application. The details of this calculation are explained in Section 3.5. As discussed earlier, we want the users opinion about permissions to be a part of our score. However, it should not be able to influence it too much, opening the possibility for uninformed users to overrule our expert opinion. To overcome this issue, we run our score through a



general logistic function fitted to our problem (see Section 3.5.3). This creates a scenario where applications we initially evaluated to be extremely dangerous are very hard to for the user to adjust towards a low threat-level. Further, the immediate steepness of our fitted function makes it easy to adjust initially low-risk applications towards a higher threat-level. Hence, the user preferences have more impact on applications with a lower initial risk score. This covers our integrity as privacy experts as we implicitly state that if our opinion is strong about an application, it is hard to argue against us.

## 6.2 Answers to Research Questions

This section will summarise answers to each research question.

*RQ1: Which techniques can be used to detect possible malicious behaviour of third-party applications based on real-time system monitoring and application analysis on an unrooted Android device?*

For a third-party application without root privileges, we have found that the best way to detect the possible malicious behaviour of other applications to be static analysis of required permissions. This is a simple way of measuring risks involved in using a certain application. To improve our permission analysis, we have developed a set of permission patterns to help further identify potential threats. Permission over-privileged applications and lack of application context are factors that make this method less precise.

We have found ways of detecting when phone sensors (e.g. camera, GPS, and microphone) are being used. However, we were not able to precisely decide which applications was using it at detection time due to the strict Android application sandboxing and other system limitations. Also, utilising these detection techniques have proven to increase battery usage and have a negative impact on other applications' functionality - sometimes making them crash.

*RQ2: What is the best way to inform users about threats in installed third-party applications on an Android device and provide them with incentives to uninstall these applications?*

We have used results from our research on RQ1 to develop a system for rating applications by their potential threat to users' privacy. This system is built on our experiments on creating comprehensive descriptions of permissions and grading them by our privacy risk, and our research on permission patterns. This information is used together with our expertise on privacy to create an initial privacy score for each application installed on the users' Android device. This

score provides users with an incentive to either uninstall a potential dangerous application or further investigate its behaviour. Because our system is based on static analysis of required permissions, we have created an interface that lets the user give feedback on his/her opinion about applications' functionality. We incorporate this feedback in our scoring system to compensate for our lack of context awareness regarding applications. Hence, our system incorporates recommender functionality for nudging users towards uninstalling dangerous applications.

The mentioned permission descriptions and permission patterns are also used to inform and educate users of what installing a new application may lead to in a privacy-related aspect.

The evaluation of our work shows a general increase in privacy awareness and knowledge about potential threats and disclosure of personal information among the participants after using our application for a short period.

*RQ3: Which user interaction patterns can be employed to make users aware of their privacy-related behaviour?*

We have identified the benefits of providing users with feedback about their privacy-related behaviour to nudge users towards better privacy awareness. This behaviour analysis encapsulates installation trends, while also alerting users about disharmony in opinions about applications. We believe this to give valuable information, and help build confidence in the application and establishing a trustworthy relationship with the user.

## 6.3 Thesis Contributions

This section summarises the contributions made by this thesis. We believe that our work can be useful for others trying to develop privacy-preserving Android applications and aim to make it open source when possible. We want to highlight (1), (2), and (5) in the below list because we evaluate the significance of these contributions to be high.

1. Identification of people's attitude towards privacy, especially how they in most cases are willing to share information with applications that provides high utility.
2. Identification of how Android sensors can be used maliciously and possible methods to detect such behaviour through an unrooted third-party application.
3. We have gathered Android permissions in groups corresponding to their privacy threat-level and also written more comprehensive descriptions of

these permissions. This provides a better way of educating users about what impact acceptance of certain permissions may have on their privacy.

4. We have proposed a model for ranking Android applications on risk to user privacy. Our approach is unique in that it uses requested permissions, patterns in requested permissions, and user feedback on how these permissions fit the application context.
5. We have implemented (3) and (4) in an Android application that also follows conventions for Explanation-Aware Computing. Our evaluation show that the application has a positive effect on users' attitude towards privacy after being used for a short period.
6. Through our evaluation of (5), we have found indicators that use of Explanation-Aware Computing in Android applications increases overall perceived trust.
7. By our usage of The Mobile Services Acceptance Model for evaluating user acceptance of (5), we have contributed to the work performed by Gao et al. [17] on forming an extension of TAM more fit for evaluating mobile services.

## 6.4 Final Remarks

We started out trying to solve a problem, searching for guidance in literature and solutions through experiments. These experiments did uncover solutions, however not satisfiable ones. As stated by Coehn and Howe [8], evaluation opens new avenues of research because experiments often raise new questions as others are answered and because it identifies deficiencies and, thus, problems for further studies. After evaluating our initial experiments, we found our project to take a new path, leading us towards new problems and new experiments. The aspects mentioned in the above discussion can not be compared to what we set out trying to create. However, we believe it answers to our goal (see Section 1.2) maybe even better. Continuous evaluation still leads to new questions. The ones we could not follow further are written down in the next section.

## 6.5 Future Work

This section presents ideas for future work. It includes both improvements to our current application and abstract ideas for future development.

### 6.5.1 Centralised Backend for Calculation of User Preferences

As discussed in Section 6.1.2, our current calculation of user preferences are affected by the fact that it is based only on one person. This person may be too concerned about privacy or have no cares at all. This will lead to deviations in their preferences compared to common sense. Having a centralised system for calculating user preferences based on all users would remove these deviations. The same method for calculating preference weights (see Section 3.5.3) could be used. However, since they are acquirement from an online server, and requires remote transit and storage of data, concerns about symmetric privacy should be taken strongly into consideration to obtain a transparent system and maintain user trust.

### 6.5.2 Detection of "Proxy Applications"

Our experiments on permission patterns have inspired the idea that two or more applications, which alone may seem harmless, may combine their permissions and potentially form a privacy risk. More research should be done to detect the internal communication between applications on the device. This includes intents, shared databases (Content Providers) and broadcast receivers. These elements lay the foundation for internal traffic of data, and can to some extent be detected. It is possible to read which broadcast receivers an application have registered, and thereby which intents it will intercept. It is also possible to read which content providers an application offers. This should serve as the starting point for this research.

### 6.5.3 System Signed Application to Access Root Privileges

A specification for our application was that it should be downloadable from Google Play Store. This limited the functionality we were allowed to request from the Android API since we could not create an application with root access. Creating an application signed with a system key, would open the possibility to request permissions granting system functionality. For the application to be signed with the system key, it has to be bundled on the system image on the device (e.g. it has to be pre-installed on the device by the phone distributor). A system signed application can request the permission `READ_LOGS` that enables it to read low-level system logfiles. These logfiles contains information about, for example, when an image is captured with the camera and which application it was that used the camera to capture it. It would enable the possibility to detect usage of device sensors by other applications in a precise manner.

#### 6.5.4 Permission Threat Level Based on Application Category

A limitation in our current permissions threat levels (see Section 3.3.3) is that they do not dynamically change based on which application are requiring them. A solution to this might be to use the third-party Android Market API to acquire application categories as this is not available in the standard Android API. Further, research should be done to find patterns of common and less common permissions of "secure" applications in each category. By applying this information, it should be possible to adjust threat level on single permissions based on the category of the application requiring them. We believe this will further improve the privacy score system by having knowledge about normal and abnormal functionality among third-party applications.

#### 6.5.5 Privacy Risk Score Evaluation

It would be valuable to evaluate the performance of our privacy risk score algorithm to be able to further enhance it and see if it is sufficient enough to be used as a method for detection. This should be done by downloading a test set of over 100 confirmed malicious applications and see how they are scored compared to the current privacy score. By doing this, it may give us an indication of the accuracy of our algorithm, or if new methods for detecting dangerous behaviour should be introduced.

#### 6.5.6 Watchdog Recommender

Another closely related project has been under development the recent year at NTNU. A privacy-oriented application recommender that aims to provide a substitute to potentially dangerous applications by crawling the Google Play store, and finding similar applications. The recommendation is based on privacy and will choose applications that have the same functionality, but with lower privacy risk. We see great potential in integrating this service into our application, giving users a simple way of replacing applications categorised as threats to their privacy.

#### 6.5.7 Multiparty Differential Privacy for Permission Weights

The idea of Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers, discussed by Pathak et al. [46], researches the idea of securing users' privacy in centralised data collection and data repositories. In their work, they propose a privacy-preserving protocol that is built upon the principle of Differential Privacy. The principle states that there is a high probability of getting

the same results from a query that is run on two almost equal data sets, and therefore nothing can be learned about any individual entry in the data set. Further, the paper researches how classifiers can be trained based on differentially private aggregated data from separate mutually untrusting parties. In our proposed future work in Section 6.5.1 we see the value in collecting user preferences towards permissions to a centralised source. This poses a threat to privacy if these preferences might be connected to the user on a server. Our potential future work in this section is based upon the idea of differential privacy together with aggregating the vector of permission weight from each user. This may be utilised to learn a classifier that further will be used to provide community-based permission weights back to the user. The realistic implementation of this feature needs to be further researched.

If we are to implement functionality that requires the application to communicate and transfer data to external sources, we need to protect the privacy of the users' identity and data. A new type of encrypting are being researched, which, in theory, shall remove the possibility of detecting the identity of data contributors, called Homomorphic Encryption. This enables the possibility of processing data without encrypting it. It enables all data to be fully encrypted at all times, during searches, in databases and operations on the server. Data would never be decrypted from the moment it leaves the user's device to the time the user views the response from the server on the device again. This technique is still in a early development phase and causes a major increase in computation time. However, improvements are constantly being made, and researchers predict it to be the future of encryption. This is a very futuristic element, but very interesting indeed.

# Bibliography

- [1] Acquisti, A. and Grossklags, J. (2005). Privacy and rationality in individual decision making. *Security Privacy, IEEE*, 3(1):26–33.
- [2] Adams, D. A., Nelson, R. R., and Todd, P. A. (1992). Perceived usefulness, ease of use, and usage of information technology: A replication. *MIS Q.*, 16(2):227–247.
- [3] Au, K. W. Y., Zhou, Y. F., Huang, Z., and Lie, D. (2012). Pscout: Analyzing the android permission specification. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12*, pages 217–228, New York, NY, USA. ACM.
- [4] Balebako, R., Jung, J., Lu, W., Cranor, L. F., and Nguyen, C. (2013). "little brothers watching you": Raising awareness of data leaks on smartphones. In *Proceedings of the Ninth Symposium on Usable Privacy and Security, SOUPS '13*, pages 12:1–12:11, New York, NY, USA. ACM.
- [5] Benbasat, I. and Barki, H. (2007). Quo vadis tam? *Journal of the Association for Information Systems*, 8(4).
- [6] Bitdefender Inc (2014). Clueful privacy advisor for android. <http://www.bitdefender.com/solutions/clueful-android.html>. Visited February 2015.
- [7] Cassens, J. and Kofod-Petersen, A. (2007). Designing explanation aware systems: The quest for explanation patterns. In *ExaCt*, pages 20–27.
- [8] Coehn, P. and Howe, A. (1988). How evaluation guides ai research. *AI Mag.*, 9(4):35–43.
- [9] Computerworld (2011). Snooping: It's not a crime, it's a feature. <http://www.computerworld.com/article/2507554/data-privacy/snooping-it-s-not-a-crime-it-s-a-feature.html?page=2>. Visited May 2015.

- [10] Cranor, L. F., Reagle, J., and Ackerman, M. S. (1999). Beyond concern: Understanding net users' attitudes about online privacy.
- [11] Criniti, C. (2012). Permissionexplorer. [https://play.google.com/store/apps/details?id=com.carlocriniti.android.permission\\_explorer](https://play.google.com/store/apps/details?id=com.carlocriniti.android.permission_explorer). Visited February 2015.
- [12] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.*, 13(3):319–340.
- [13] Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., and Sheth, A. N. (2010). Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, OSDI'10*, pages 1–6, Berkeley, CA, USA. USENIX Association.
- [14] Enck, W., Ongtang, M., and McDaniel, P. (2009). On lightweight mobile phone application certification. In *Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS '09*, pages 235–245, New York, NY, USA. ACM.
- [15] Fang, Z., Han, W., and Li, Y. (2014). Permission based android security: Issues and countermeasures. *Computers Security*, 43(0):205 – 218.
- [16] Felt, A. P., Chin, E., Hanna, S., Song, D., and Wagner, D. (2011). Android permissions demystified. In *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11*, pages 627–638, New York, NY, USA. ACM.
- [17] Gao, S., Krogstie, J., and Siau, K. (2014). Adoption of mobile information services: An empirical study. *Mob. Inf. Syst.*, 10(2):147–171.
- [18] Geneiatakis, D., Fovino, I. N., Kounelis, I., and Stirparo, P. (2014). A permission verification approach for android mobile applications. *Computers Security*, (0):-.
- [19] Glass, A., McGuinness, D. L., and Wolverton, M. (2008). Toward establishing trust in adaptive agents. In *Proceedings of the 13th International Conference on Intelligent User Interfaces, IUI '08*, pages 227–236, New York, NY, USA. ACM.
- [20] Good, N., Dhamija, R., Grossklags, J., Thaw, D., Aronowitz, S., Mulligan, D., and Konstan, J. (2005). Stopping spyware at the gate: A user study of privacy, notice and spyware. In *Proceedings of the 2005 Symposium on Usable Privacy and Security, SOUPS '05*, pages 43–52, New York, NY, USA. ACM.



- [21] Google Inc (2015a). Android api documentation. <http://developer.android.com/reference/packages.html>. Visited May 2015.
- [22] Google Inc (2015b). Android manifest permissions. <https://developer.android.com/reference/android/Manifest.permission.html>. Visited February 2015.
- [23] Google Inc (2015c). Application security. <http://source.android.com/devices/tech/security/overview/app-security.html>. Visited December 2014.
- [24] Google Inc (2015d). Broadcastreceivers. <http://developer.android.com/reference/android/content/BroadcastReceiver.html>. Visited February 2015.
- [25] Google Inc (2015e). Intents and intent filters. <http://developer.android.com/guide/components/intents-filters.html>. Visited February 2015.
- [26] Google Inc (2015f). Material design guidelines. <http://www.google.com/design/spec/material-design/introduction.html>. Visited May 2015.
- [27] Google Inc (2015g). Services. <http://developer.android.com/guide/components/services.html>. Visited February 2015.
- [28] Hull, G., Lipford, H., and Latulipe, C. (2011). Contextual gaps: privacy issues on facebook. *Ethics and Information Technology*, 13(4):289–302.
- [29] IEEE Spectrum (2014). Which mobile apps are worst privacy offenders? <http://spectrum.ieee.org/tech-talk/consumer-electronics/portable-devices/new-website-shows-lists-which-mobile-apps-are-worst-privacy-offenders/>. Visited May 2015.
- [30] International Data Corporation (2015). Android market share. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Visited May 2015.
- [31] Jiang, X., Hong, J. I., and Landay, J. A. (2002). Approximate information flows: Socially-based modeling of privacy in ubiquitous computing. In *Proceedings of the 4th International Conference on Ubiquitous Computing, UbiComp '02*, pages 176–193, London, UK, UK. Springer-Verlag.
- [32] John, M. (2015). D-vasive. <https://play.google.com/store/apps/details?id=com.dVasivehl=en>. Visited February 2015.

- [33] Kelley, P. G., Consolvo, S., Cranor, L. F., Jung, J., Sadeh, N., and Wetherall, D. (2012). A conundrum of permissions: Installing applications on an android smartphone. In *Proceedings of the 16th International Conference on Financial Cryptography and Data Security, FC'12*, pages 68–79, Berlin, Heidelberg. Springer-Verlag.
- [34] Lacave, C. and Diez, F. J. (2000). A review of explanation methods for bayesian networks. *Knowledge Engineering Review*, 17:2002.
- [35] Legris, P., Ingham, J., and Colletette, P. (2003). Why do people use information technology?: A critical review of the technology acceptance model. *Inf. Manage.*, 40(3):191–204.
- [36] Lin, J., Amini, S., Hong, J. I., Sadeh, N., Lindqvist, J., and Zhang, J. (2012). Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12*, pages 501–510, New York, NY, USA. ACM.
- [37] Madsen, M. and Gregor, S. (2000). Measuring human-computer trust. In *Proceedings of the 11 th Australasian Conference on Information Systems*, pages 6–8.
- [38] Mayer, R. C., Davis, J. H., , and Schoorman, F. D. (1995). An integrative model of organizational trust. *The Academy of Management Review*.
- [39] McAllister, D. J. (1995). Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations. *The Academy of Management Journal*, 38(1):pp. 24–59.
- [40] MIT Technology Review (2015). The truth about smartphone apps that secretly connect to user tracking and ad sites. <http://www.technologyreview.com/view/537186/the-truth-about-smartphone-apps-that-secretly-connect-to-user-tracking-and-ad-sites/>. Visited May 2015.
- [41] Moonsamy, V., Rong, J., and Liu, S. (2014). Mining permission patterns for contrasting clean and malicious android applications. *Future Generation Computer Systems*, 36(0):122 – 132. Special Section: Intelligent Big Data Processing Special Section: Behavior Data Security Issues in Network Information Propagation Special Section: Energy-efficiency in Large Distributed Computing Architectures Special Section: eScience Infrastructure and Applications.
- [42] Muir, B. M. (1992). Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. In *Ergonomics*.

- [43] Network World (2014). Popular android apps fail basic security tests, putting privacy at risk. <http://www.networkworld.com/article/2603901/popular-android-apps-fail-basic-security-tests-putting-privacy-at-risk.html>. Visited May 2015.
- [44] Noam, E. (1997). Privacy and self-regulation: Markets for electronic privacy, in privacy and self-regulation in the information age.
- [45] Parasuraman, R. and Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. human factors. *The Journal of the Human Factors and Ergonomics Society*.
- [46] Pathak, M., Rane, S., and Raj, B. (2010). Multiparty differential privacy via aggregation of locally trained classifiers. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1876–1884. Curran Associates, Inc.
- [47] Peng, H., Gates, C., Sarma, B., Li, N., Qi, Y., Potharaju, R., Nita-Rotaru, C., and Molloy, I. (2012). Using probabilistic generative models for ranking risks of android apps. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12*, pages 241–252, New York, NY, USA. ACM.
- [48] Rebecca Balebako, Pedro G. Leon, H. A. P. G. K. J. M. A. A. L. F. C. and Sadeh., N. (2011). Nudging users towards privacy on mobile devices. In *Workshop on Persuasion, Influence, Nudge and Coercion Through Mobile Devices (PINC at CHI-11)*.
- [49] Robinson, J., Shaver, P., and L.S., W. (1991). Criteria for scale selections and evaluation. *Academic Press*.
- [50] Sarma, B. P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., and Molloy, I. (2012). Android permissions: A perspective combining risks and benefits. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, SACMAT '12*, pages 13–22, New York, NY, USA. ACM.
- [51] Schank, R. C. (1986). *Explanation Patterns: Understanding Mechanical and Creatively*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.
- [52] Spiekermann, S., Grossklags, J., and Berendt, B. (2001). E-privacy in 2nd generation e-commerce: privacy preferences versus actual behavior.
- [53] Trendmicro (2014). 12 most abusive permissions. <http://about-threats.trendmicro.com/us/library/image-gallery/12-most-abused-android-app-permissions>. Visited May 2015.

- 
- [54] UNM Newsroom (2015). Smart phones do not keep secrets. <http://news.unm.edu/news/smart-phones-do-not-keep-secrets>. Visited May 2015.
- [55] Ur, B., Leon, P. G., Cranor, L. F., Shay, R., and Wang, Y. (2012). Smart, useful, scary, creepy: Perceptions of online behavioral advertising. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 4:1–4:15, New York, NY, USA. ACM.
- [56] USC News (2015). ‘free’ apps may not be so free after all: They take a big toll on your phone. <https://news.usc.edu/79081/beware-of-an-ads-hidden-costs-in-free-mobile-apps/>. Visited May 2015.
- [57] Vigneri, L., Chandrashekar, J., Pefkianakis, I., and Heen, O. (2015). Taming the android appstore: Lightweight characterization of android applications. *CoRR*, abs/1504.06093.

## Appendix A

# Permission Descriptions and Threat Levels

Name	Designation	Risk	Description
ACCESS_COARSE_LOCATION	Nettverksbasert posisjon	3	Gir tilgang til tilnærnet posisjon ved hjelp av Wi-Fi eller triangulering av mobilsignaler
ACCESS_FINE_LOCATION	Nøyaktig posisjon	3	Gir tilgang til nøyaktig posisjon ved hjelp av GPS, Wi-Fi eller triangulering av mobilsignaler.
DISABLE_KEYGUARD	Deaktivere skjermlås	3	Gir tilgang til å deaktivere telefonens skjermlås (pinkode, mønster, fingeravtrykk etc.)
PROCESS_OUTGOING_CALLS	Redirigere utgående samtaler	3	Gir tilgang til å se informasjon om en utgående samtale, bl.a. telefonnummer. Gir også muligheten til å redirigere utgående samtaler til et annet nummer enn det du tastet inn eller fullstendig avslutte samtalen.
READ_CALENDAR	Se kalenderhendelser	3	Gir tilgang til å se detaljer om hendelser i din kalender.
READ_CALL_LOG	Se samtalelogg	3	Gir tilgang til å se telefonens samtalelogg.

READ_CONTACTS	Se kontaktliste	3	Gir tilgang til informasjon om alle kontakter lagret på telefonen. Denne informasjonen kan bl.a. inkludere navn, telefonnummer, husadresse, e-mailadresse.
READ_PROFILE	Se ditt kontaktkort	3	Gir tilgang til å se personlige opplysninger lagret i ditt kontaktkort.
READ_SMS	Se dine SMS og MMS	3	Gir tilgang til å lese dine sendte og mottatte SMS og MMS i sin helhet. Enkelte applikasjoner bruker denne funksjonaliteten for å automatisk mottat bekreftelseskoder sendt på SMS.
READ_SOCIAL_STREAM	Se nyhetsstrøm på sosiale medier	3	Gir tilgang til å se dine og dine venners oppdateringer på sosiale medier, f.eks. Facebook, Google+ eller Twitter.
RECEIVE_MMS	Oppdage mottatt MMS	3	Gir tilgang til å få beskjed når en MMS blir mottatt på din telefon. Gir også tilgang til å lese innholdet.
RECEIVE_SMS	Oppdage mottatt SMS	3	Gir tilgang til å få beskjed når en SMS blir mottatt på din telefon. Gir også tilgang til å lese innholdet.

RECORD_AUDIO	Ta opp lyd med mikrofonen	3	Gir tilgang til å ta opp lyd med mikrofonen på din telefon. Lyd kan spilles inn uten at du får vite om det.
CAMERA	Ta bilder og videoer	3	Gir tilgang til å ta bilder og ta opp video med alle telefonens kameraer. Dette kan gjøres uten at du får vite det.
GET_TASKS	Se kjørende applikasjoner	3	Gir tilgang til å til enhver tid se hvilke applikasjoner som kjører på telefonen.
READ_HISTORY_BOOKMARKS	Se din nettleserhistorie og dine bookmarks	3	Gir tilgang til å se hvilke nettsteder du har besøkt i din nettleserhistorie, samt hvilke sider du har lagret som bookmarks.
READ_PHONE_STATE	Se telefonens status og identitet	3	Gir tilgang til å se unike identitetsnummer forbundet med din telefon. Kan også se telefonens status, f.eks. at en telefonsamtale er gående.
USE_CREDENTIALS	Hente påloggingsinformasjon fra tilkoblede kontoer	3	Gir tilgang til å bruke påloggingsinformasjon fra tilkoblede kontoer (Google, Facebook, Dropbox etc.) på din telefon.



READ_EXTERNAL_STORAGE	Se data på en ekstern lagringsenhet	3	Gir tilgang til å se data lagret på en ekstern lagringsenhet, f.eks. ekstern harddisk koblet til med USB.
AUTHENTICATE_ACCOUNTS	Opprette tilkoblede kontoer	2	Gir tilgang til å opprette en ny tilkoblet konto med brukernavn og passord på din telefon.
CALL_PHONE	Ring et telefonnummer	2	Gir tilgang til å ringe et telefonnummer (utenom nummerene) uten ditt samtykke. Kan f.eks. brukes til å ringe betaljenester.
KILL_BACKGROUND_PROCESSES	Slukke andre applikasjoner	2	Gir tilgang til å lukke en kjørende applikasjon. Kan f.eks. brukes til å lukke telefonens antivirus-applikasjon.
USE_SIP	Tillate telefoni over internett	2	Gir tilgang til å utføre inter-nettsamtaler.
WRITE_CALENDAR	Legge til hendelser i kalenderen	2	Gir tilgang til å legge til hendelser i din kalender.
WRITE_CALL_LOG	Endre eller slette innhold i samtaleloggen	2	Gir tilgang til å endre telefonens samtalelogg, inkludert data om inkommande og utgående samtaler. Farlige applikasjoner kan bruke dette til å skjule uønsket adferd ved å f.eks. slette logg om en spesifikk samtale.

WRITE_CONTACTS	Endre kontaktlisten	2	Gir tilgang til å endre din kontakliste, herunder legge til nye kontakter.
WRITE_PROFILE	Endre ditt kontaktkort	2	Gir tilgang til å endre personlig informasjon lagret i ditt kontaktkort.
WRITE_SMS	Skrive SMS eller MMS	2	Gir tilgang til å skrive, men ikke sende SMS eller MMS.
WRITE_SOCIAL_STREAM	Skrive innlegg til din nyhetsstrøm på sosiale medier	2	Gir tilgang til å poste innlegg sosiale medier registrert på telefonen, f.eks. Facebook, Google+, Twitter.
ACCESS_MOCK_LOCATION	Falsk posisjon	2	Gir mulighet for å utgi seg på en annen posisjon enn der telefonen faktisk befinner seg.
ACCESS_NETWORK_STATE	Se nettverkstilkoblinger	2	Gir tilgang til informasjon om nettverk, inkludert hvilke nettverk telefonen er koblet til og hvilke nettverk som er tilgjengelige i området der telefonen befinner seg.
ACCESS_WIFI_STATE	Se Wi-Fi-nettverkstilkoblinger	2	Gir tilgang til informasjon om Wi-Fi-nettverk, inkludert hvilke Wi-Fi-nettverk telefonen er koblet til og hvilke Wi-Fi-nettverk som er tilgjengelige i området der telefonen befinner seg.

ADD_VOICEMAIL	Legg til talepost	2	Gir en applikasjon mulighet til å legge til meldinger i din talepostkasse.
BLUETOOTH	Paring med andre Bluetooth-enheter	2	Gir tilgang til å koble til parede Bluetooth-enheter.
BLUETOOTH_ADMIN	Bluetooth innstillinger	2	Gir tilgang til å søke etter og pare med andre Bluetooth-enheter.
CHANGE_CONFIGURATION	Endre telefoninnstillinger	2	Gir tilgang til å endre enkelte av dine telefoninnstillinger (f.eks. standard språk på mobilen).
CHANGE_NETWORK_STATE	Endre nettverkstilkobling	2	Gir tilgang til å slå av og på bruk av nettverk. Kan også koble seg til eller fra et nettverk.
CHANGE_WIFI_STATE	Koble til eller koble fra Wi-Fi	2	Gir tilgang til å slå av og på bruk av Wi-Fi. Kan også koble seg til eller fra et Wi-Fi-nettverk.
CLEAR_APP_CACHE	Slette korttidsminne	2	Gir tilgang til å slette korttidsminnet for alle innstallerte applikasjoner på telefonen. I korttidsminnet ligger data som nylig er blitt brukt av en applikasjon slik at den skal kunne hente det raskere.

GET_ACCOUNTS	Se tilkoblede kontoer	2	Gir tilgang til å se informasjon om dine tilkoblede kontoer på telefonen, f.eks. Facebook eller Google.
MANAGE_ACCOUNTS	Administrere tilkoblede kontoer	2	Gir tilgang til å legge til eller fjerne tilkoblede kontoer, f.eks. Facebook eller Google.
MODIFY_AUDIO_SETTINGS	Endre lydinnstillinger	2	Gir tilgang til å endre telefonens lydinnstillinger, f.eks. sette på lydløs eller justere volumet.
NFC	Kontrollere Near Field Communication (NFC)	2	Gir tilgang til å sende eller motta data over NFC. For at NFC skal kunne fungere må enhetene som kommuniserer være kun få centimeter fra hverandre.
READ_SYNC_STATS	Se synkroniseringsstatistikk	2	Gir tilgang til å se når synkronisering av en applikasjon skjedde og hvor mye data som ble synkronisert.
REORDER_TASKS	Reorganisere kjørende applikasjoner	2	Gir tilgang til å endre rekkefølgen på kjørende applikasjoner. Reklameapplikasjoner kan f.eks. bruke dette til å kjøre en reklame over alle andre applikasjoner.

SEND_SMS	Sende SMS	2	Gir tilgang til å sende SMS på dine vegne. Dette kan f.eks. brukes til å sende SMS til dyre betalingsnummere.
SUBSCRIBED_FEEDS_READ	Se RSS nyhetsstrøm	2	Gir tilgang til å se hvilken RSS nyheter du abonnerer på og innholdet i disse. Ikke farlig hvis du ikke abonnerer på RSS.
SUBSCRIBED_FEEDS_WRITE	Endre RSS nyhetsstrøm	2	Gir tilgang til å endre din RSS nyhetsstrøm.
WRITE_EXTERNAL_STORAGE	Endre eller slette innhold på SD-kortet	2	Gir tilgang til å endre eller slette eksisterende filer eller opprette nye filer på telefonens SD-kort.
WRITE_HISTORY_BOOKMARKS	Legge til nettleserhistorie eller bookmarks	2	Gir tilgang til å legge til sider i din nettleserhistorie. Gir også tilgang til å opprette nye bookmarks i din nettleser.
WRITE_SYNC_SETTINGS	Skru synkronisering av applikasjoner av eller på	2	Gir tilgang til å skru av eller på synkronisering av applikasjoner.
ACCESS_WIFI_STATE	Se nettverkstilkoblinger	1	
BODY_SENSORS	Kroppssensorer	1	Gir applikasjonen tilgang til å benytte tilgjengelige kroppssensorer som f.eks. pulsmåler.

CHANGE_WIFI_MULTICAST_STATE	Åpne WiFi multicast	1	Gir tilgang til å motta og sende nettværkspakker på multicast modus. Kan påvirke din batteritid negativt.
CHANGE_WIMAX_STATE	Koble til eller koble fra WiMAX	1	Gir tilgang til å slå av og på bruk av WiMAX. WiMAX er en type 4g-nettverk.
EXPAND_STATUS_BAR	Åpne eller lukke statuslinjen	1	Gir tilgang til å enten åpne eller lukke statuslinjen på din telefon.
FLASHLIGHT	Bruke lommelykt	1	Gir tilgang til å bruke LED-lyset på telefonen som lommelykt. Kan øke telefonens batteribruk drastisk.
GET_PACKAGE_SIZE	Måle størrelse på en app-likasjon	1	Gir tilgang til å måle hvor mye plass en applikasjon tar opp på telefonen.
INSTALL_SHORTCUT	Opprette snarvei	1	Gir tilgang til å opprette en snarvei på telefonens hjemmeskjerm. Kan brukes til å opprette falske snarveier som utgir seg for å åpne noe annet enn det de faktisk gjør.
INTERNET	Internettilgang	1	Gir tilgang til å bruke internett. Kan laste opp og ned data uten at du får vite det.

READ_SYNC_SETTINGS	Se synkroniseringsinnstillinger	1	Gir tilgang til å se om automatisk synkronisering er skrudd på eller av for innstilte app-likasjoner.
READ_USER_DICTIONARY	Se ord i din ordliste	1	Gir tilgang til å se hvilke ord som er lagret i din personlige ordliste.
RECEIVE_BOOT_COMPLETED	Oppdage oppstart av telefonen	1	Gir tilgang til å få beskjed om at telefonen akkurat ble slått på. Brukes ofte av app-likasjoner som skal starte når telefonen starter.
RECEIVE_WAP_PUSH	Oppdage mottatt WAP push-melding	1	Gir tilgang til å få beskjed når en WAP push-melding blir mottatt på din telefon. WAP push forekommer f.eks. når din telefon mottar informasjon om at en MMS er klar for å lastes ned.
SET_ALARM	Opprette en alarm	1	Gir tilgang til å opprette en alarm til et bestemt tidspunkt på din mobil.
SET_TIME_ZONE	Endre tidssone	1	Gir tilgang til å endre tidssonen som blir brukt på din telefon.
SET_WALLPAPER	Endre bakgrunn på hjem-skjerm	1	Gir tilgang til å endre bakgrunnsbilde eller bakgrunnsanimasjon på din hjemskjerm.

SYSTEM_ALERT_WINDOW	Kjøre over andre applikasjoner	1	Gir tilgang til å kjøre "popup"-vinduer over andre applikasjoner. Reklameapplikasjoner kan typisk bruke denne tillatelsen til å vise reklame. I utgangspunktet ikke ment for vanlige applikasjoner.
TRANSMIT_IR	Bruke infrarød (IR) sender	1	Gir tilgang til å bruke telefonens infrarøde sender dersom en slik er tilgjengelig.
UNINSTALL_SHORTCUT	Fjerne snarveier	1	Gir tilgang til å fjerne snarveier på telefonens hjemskjerm.
VIBRATE	Kontroll over vibrator	1	Gir tilgang til å kontrollere telefonens vibrator.
WAKE_LOCK	Forhindre skjermen i å skru seg av	1	Gir tilgang til å forhindre at skjermen skrur seg av etter inaktivitet. Kan i stor grad påvirke telefonens strømforbruk.
WRITE_USER_DICTIONARY	Legg til ord i din ordliste	1	Gir tilgang til å legge til ord i din persolige ordliste. Auto-correct bruker denne ordlisten når du skriver f.eks. SMS.
WRITE_VOICEMAIL	Endre eller fjerne talepost	1	Gir tilgang til å endre eller fjerne eksisterende talepost på din telefon.



## Appendix B

# Mobile Services Acceptance Model Questionnaire

# Spørreundersøkelse

Takk for at du har brukt appen vår. Vennligst svar på denne avsluttende spørreundersøkelsen.

I spørsmålene kan svarene graderes fra 1 (mest negativ) til 7 (mest positiv)

**\*Må fylles ut**

## 1. Vennligst skriv inn ditt ID-nummer \*

.....

## 2. Nytteverdi \*

Markér bare én oval per rad

	1	2	3	4	5	6	7
(1) Å bruke appen har gjort meg bedre i stand til å ivareta mitt personvern	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(2) Å bruke appen har gjort meg mer oppmerksom på personvernstrulser	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(3) Ved å bruke appen har jeg lært hvordan jeg kan begrense informasjonsstjeling fra telefonen min	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 3. Brukervennlighet \*

Markér bare én oval per rad

	1	2	3	4	5	6	7
(4) Det var lett å bruke appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(5) Jeg fant lett frem til den informasjonen jeg lette etter i appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(6) Brukergrensesnittet i appen var enkelt å forstå	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 4. Personlige faktorer \*

Markér bare én oval per rad

	1	2	3	4	5	6	7
(7) Jeg synes det var spennende å bruke appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(8) Å bruke appen gir meg en fordel over de som ikke bruker den	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(9) Jeg finner det givende å bruke appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**5. Tillit \***

Jeg syns...

*Markér bare én oval per rad*

	1	2	3	4	5	6	7
(10) jeg fikk et klart bilde over hensikten til appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(11) jeg stoler på utgiveren av appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(12) appen ikke bryter mitt personvern	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(13) jeg føler at informasjonen i appen stemmer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(14) jeg føler at det er risikofritt å bruke appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**6. Kontekst**

Jeg ville brukt appen...

*Markér bare én oval per rad*

	1	2	3	4	5	6	7
(15) når jeg sitter på bussen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(16) når jeg laster ned en ny app	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(17) før jeg legger meg	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(18) når jeg får notifikasjon fra appen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**7. Intensjon om bruk \****Markér bare én oval per rad*

	1	2	3	4	5	6	7
(19) Jeg ville brukt appen hvis jeg hadde hatt mulighet til å laste den ned fra Google Play	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(20) Jeg ville brukt denne appen hvis den ble utgitt av teleoperatøren min	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Drevet av





## Appendix C

# Privacy Survey Questionnaire

# Spørreundersøkelse

\*Må fylles ut

## 1. Vennligst skriv inn ditt ID-nummer \*

.....

## 2. I hvilken grad passer følgende utsagn om deg? \*

Markér bare én oval per rad

	Ikke i det hele tatt	I liten grad	Nøytral	Noe	I stor grad
(1) Jeg er opptatt av personvern	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(2) Jeg har full oversikt over hvilke personlige data som benyttes og samles inn av appene på telefonen min	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(3) Jeg er kritisk til apper jeg installerer på telefonen min	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(4) Så lenge en app har stor nytte- eller underholdningsverdi, så beholder jeg den på telefonen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(5) Jeg føler jeg mangler kontroll over hva som foregår på telefonen min	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 3. I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen? \*

Markér bare én oval per rad

	Aldri	Sjeldent	Av og til	Ofte	Alltid	Vet ikke
(6) Tillatelser	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(7) Utgiver	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(8) Beskrivelse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(9) Kategori	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(10) Vurderinger	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(11) Antall nedlastinger	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om? \*

Markér bare én oval per rad

	Svært dårlig	Dårlig	Nøytral	Godt	Svært godt
(12) Enhets- og apploggen: Lar appen se noe av eller all denne informasjon: informasjon om aktivitet på enheten, apper som kjører, nettleserloggen og bokmerker	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(13) Identitet: Bruker minst en av disse: kontoer på enheten, profildata	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(14) Kontakter: Bruker kontaktinformasjonen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(15) Posisjon: Bruker enhetens posisjon	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(16) Kamera: Bruker enhetens kamera(er)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(17) Tekstmelding: Bruker minst en av disse: tekstmelding, multimediamelding. Kan medføre kostnader	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(18) Enhets-ID og anropsinformasjon: Lar appen fastslå telefonnummeret og enhets-ID-er, om en samtale pågår og det eksterne nummeret det opprettes forbindelse med under et anrop	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen? \*

Markér bare én oval per rad

	Meget uvillig	Uvillig	Nøytral	Villig	Meget villig
(19) Alder	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(20) Kjønn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(21) Din internetlogg og bokmerker	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(22) Hjemmeadresse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(23) Mobilnummer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(24) Kreditkortnummer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(25) Kontaktlisten på telefonen din	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(26) Din lokasjon	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(27) Din venneliste på sosiale medier	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(28) Informasjon om andre installerte apper på telefonen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(29) Innholdet i dine sendte og mottatte SMS-meldinger	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(30) Din samtalelogg	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(31) Bilder	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 6. Er du enige med de følgende uttalelsene? \*

Markér bare én oval per rad

	Veldig uenig	Uenig	Nøytral	Enig	Veldig enig
(32) Jeg er bekymret for at apper samler for mye informasjon om meg	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(33) Jeg er bekymret for at selskaper bruker min informasjon til andre hensikter enn det de i utgangspunktet sier de skal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(34) Forbrukere har mistet kontroll over hvordan og hvor mye informasjon som blir samlet inn av selskaper	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 7. Er du komfortabel med bruk av din personlige data til.. \*

Markér bare én oval per rad

	Veldig ukomfortabel	Ukomfortabel	Nøytral	Komfortabel	Veldig komfortabel
(35) Apps- og Internettjenester tilpasset dine preferanser og behov	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(36) Personalisert kundeservice	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(37) Personalisert reklame	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 8. I hvilken grad stoler du på følgende bedrifter/institusjoner? \*

Markér bare én oval per rad

	Sterk mistillit	Mistillit	Nøytral	Tillit	Sterk tillit
(38) Offentlige organer(f.eks. Skatteetaten)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(39) Sosiale medier(f.eks. Facebook)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(40) Globale Internett-bedrifter(f.eks. Google)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(41) Din mobiloperatør(f.eks. NetCom)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 9. Ville du gitt følgende bedrifter/institusjoner tillatelse til å bruke din personlige informasjon for å i gjengjeld få personaliserte apps- og Internettjenester? \*

Markér bare én oval per rad

	Meget uvillig	Uvillig	Nøytral	Villig	Meget villig
(42) Offentlige organer (f.eks. Skatteetaten)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(43) Sosiale medier (f.eks. Facebook)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(44) Globale Internetbedrifter (f.eks. Google)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
(45) Din mobiloperatør (f.eks. NetCom)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



## Appendix D

# Privacy Survey Questionnaire Results

### D.1 Questionnaire Results Before and After Application Usage

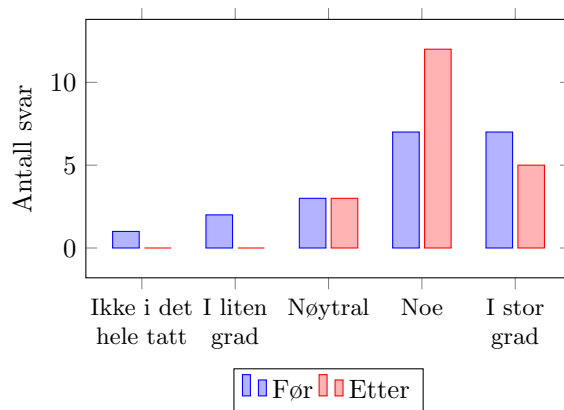


Figure D.1: (Q1) Jeg er opptatt av personvern [I hvilken grad passer følgende utsagn om deg?]

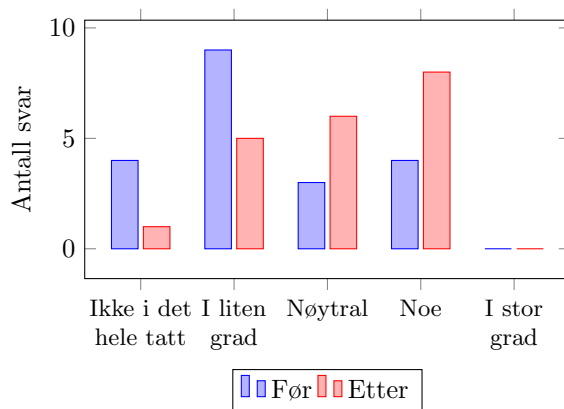


Figure D.2: (Q2) Jeg har full oversikt over hvilke personlige data som benyttes og samles inn av appene på telefonen min [I hvilken grad passer følgende utsagn om deg?]

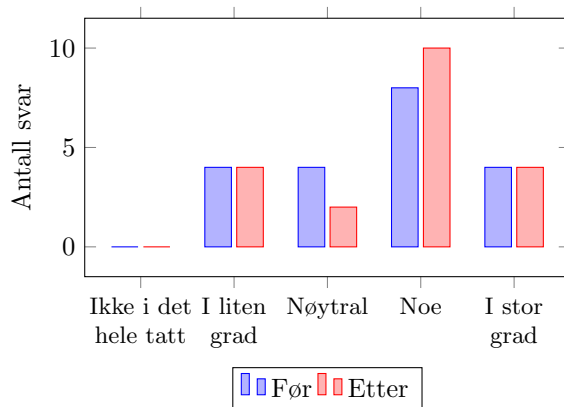


Figure D.3: (Q3) Jeg er kritisk til apper jeg installerer på telefonen min [I hvilken grad passer følgende utsagn om deg?]

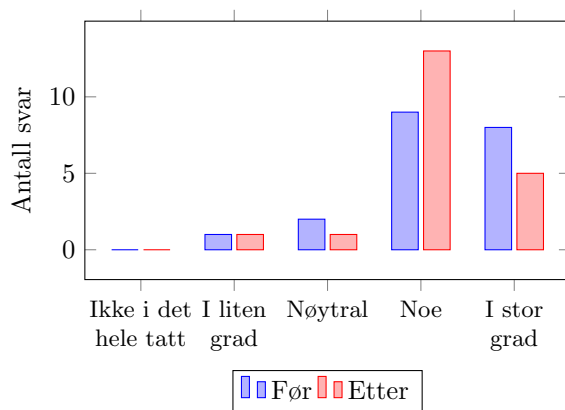


Figure D.4: (Q4) Så lenge en app har stor nytte- eller underholdningsverdi, så beholder jeg den på telefonen [I hvilken grad passer følgende utsagn om deg?]

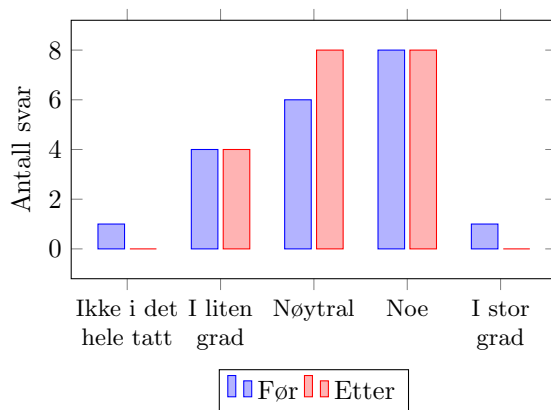


Figure D.5: (Q5) Jeg føler jeg mangler kontroll over hva som foregår på telefonen min [I hvilken grad passer følgende utsagn om deg?]

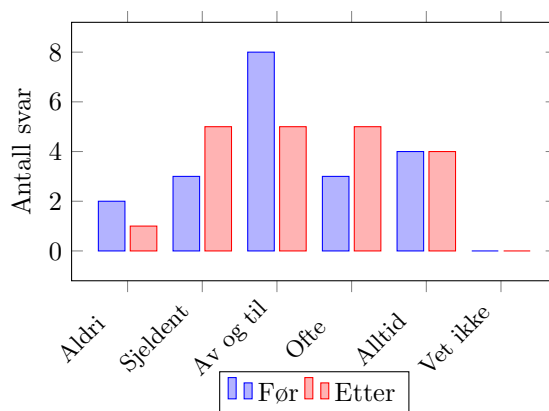


Figure D.6: (Q6) Tillatelser [I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen?]

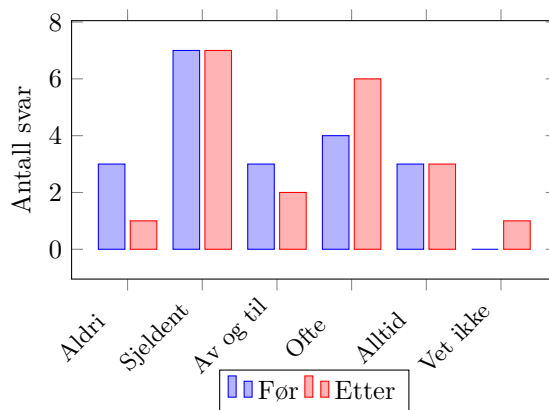


Figure D.7: (Q7) Utgiver [I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen?]

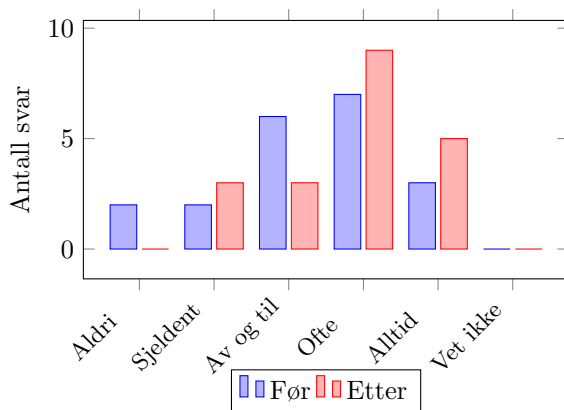


Figure D.8: (Q8) Beskrivelse [I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen?]

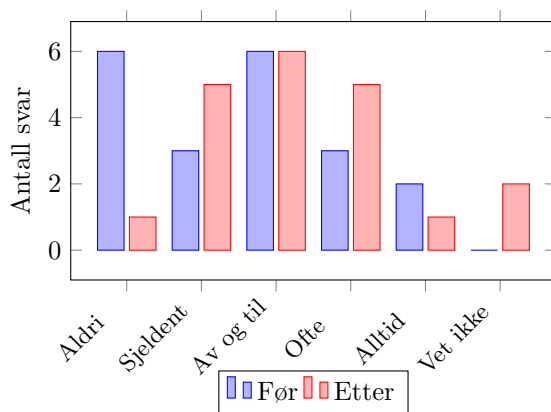


Figure D.9: (Q9) Kategori [I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen?]

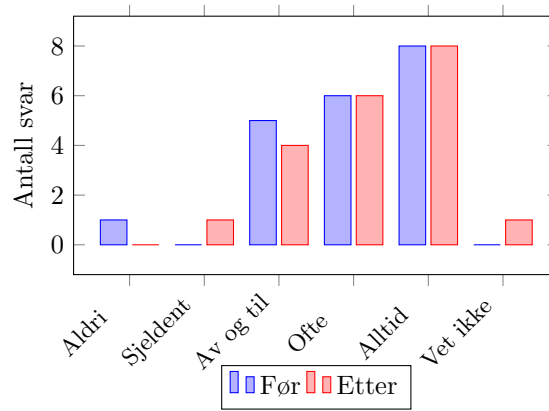


Figure D.10: (Q10) Vurderinger [I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen?]

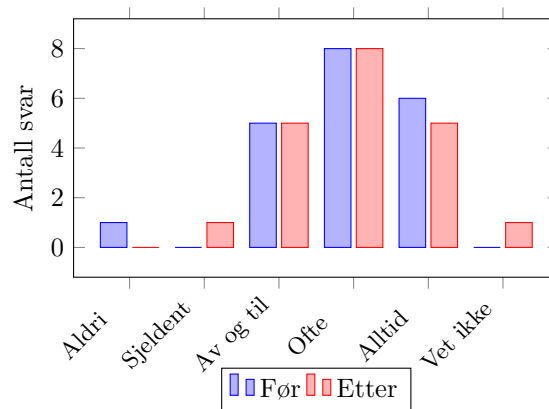


Figure D.11: (Q11) Antall nedlastinger [I hvilken grad benytter du følgende kriterier når du installerer en ny app på telefonen?]

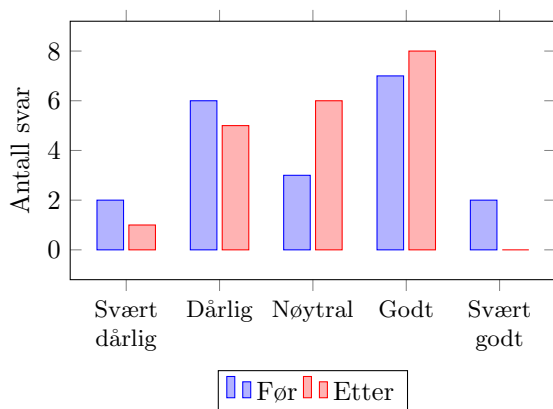


Figure D.12: (Q12) Enhets- og apploggen: Lar appen se noe av eller all denne informasjon: informasjon om aktivitet på enheten, apper som kjører, nettleserloggen og bokmerker [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]

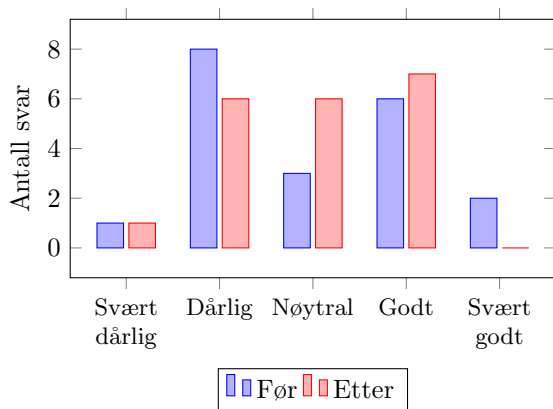


Figure D.13: (Q13) Identitet: Bruker minst en av disse: kontoer på enheten, profildata [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]

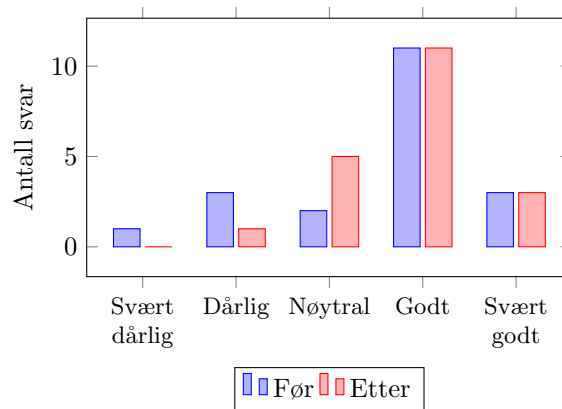


Figure D.14: (Q14) Kontakter: Bruker kontaktinformasjonen [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]

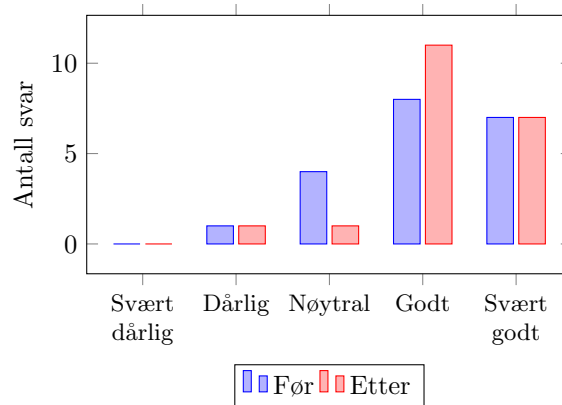


Figure D.15: (Q15) Posisjon: Bruker enhetens posisjon [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]



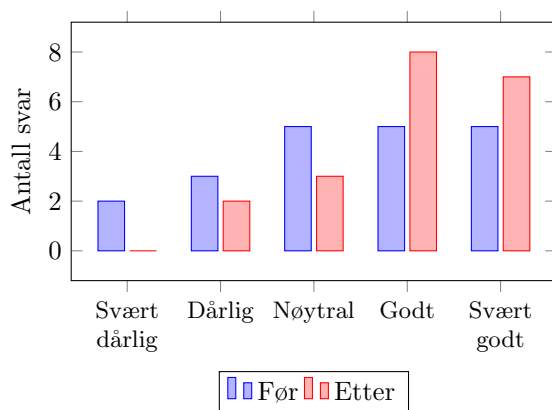


Figure D.16: (Q16) Kamera: Bruker enhetens kamera(er) [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]

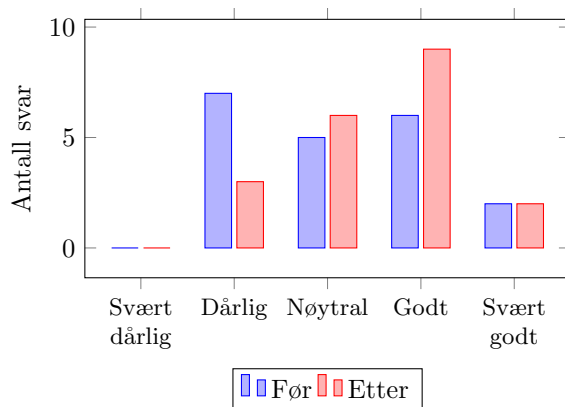


Figure D.17: (Q17) Tekstmelding: Bruker minst en av disse: tekstmelding, multimediamelding. Kan medføre kostnader [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]

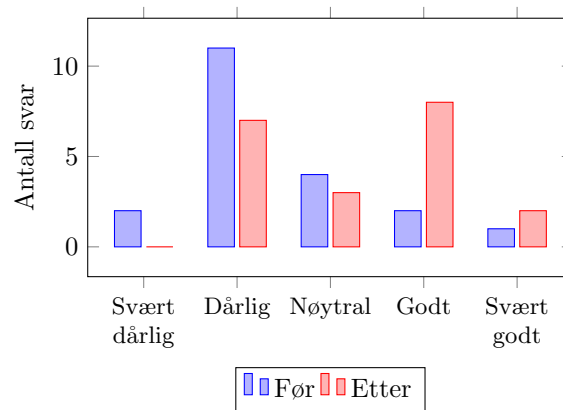


Figure D.18: (Q18) Enhets-ID og anropsinformasjon: Lar appen fastslå telefonnummeret og enhets-ID-er, om en samtale pågår og det eksterne nummeret det opprettes forbindelse med under et anrop [Du installerer en ny app på telefonen, og den ber om følgende tilganger. I hvilken grad forstår du hva den ber om?]

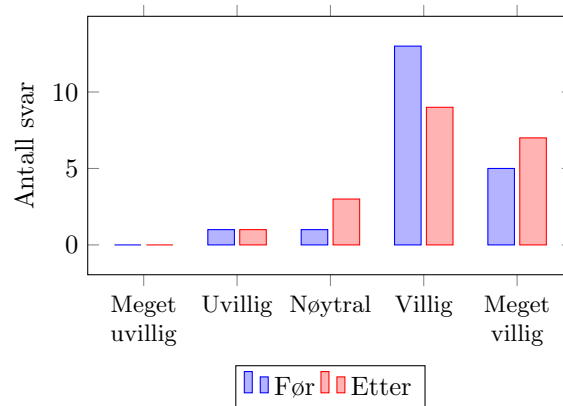


Figure D.19: (Q19) Alder [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

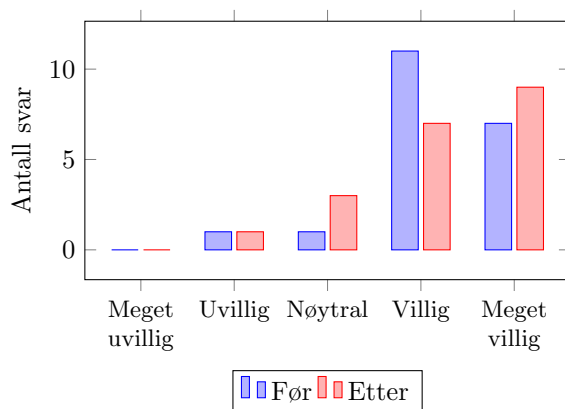


Figure D.20: (Q20) Kjønn [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

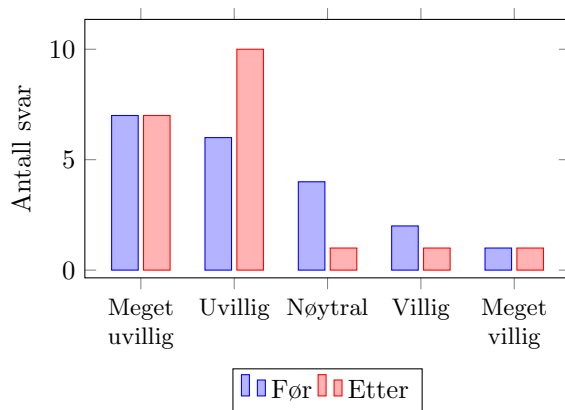


Figure D.21: (Q21) Din internetlogg og bokmerker [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

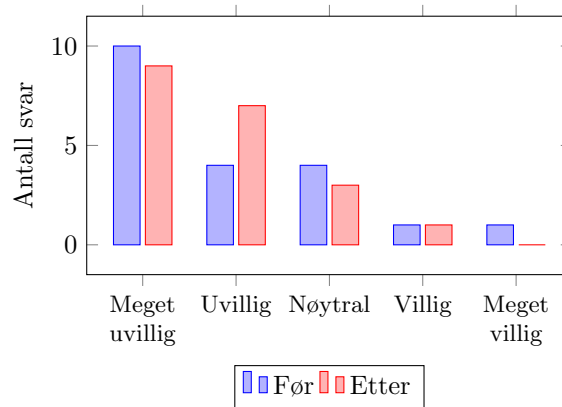


Figure D.22: (Q22) Hjemmeadresse [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

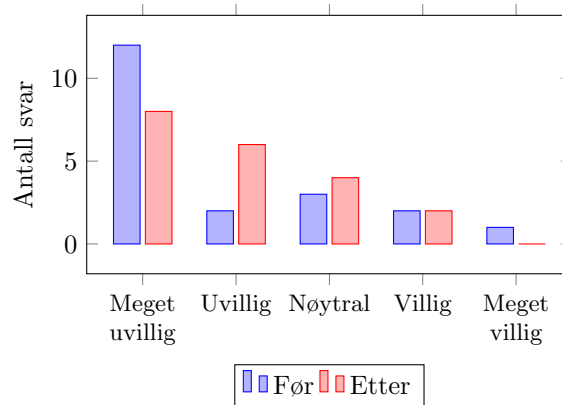


Figure D.23: (Q23) Mobilnummer [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

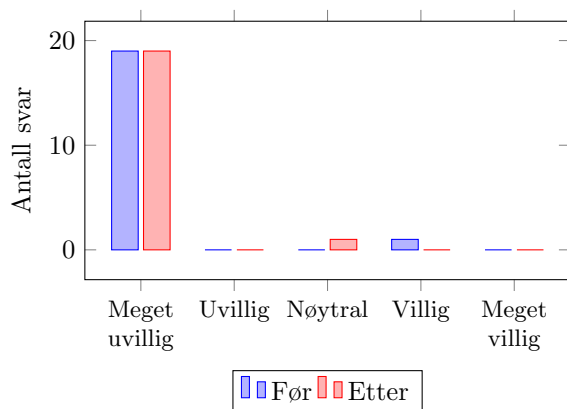


Figure D.24: (Q24) Kreditkortnummer [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

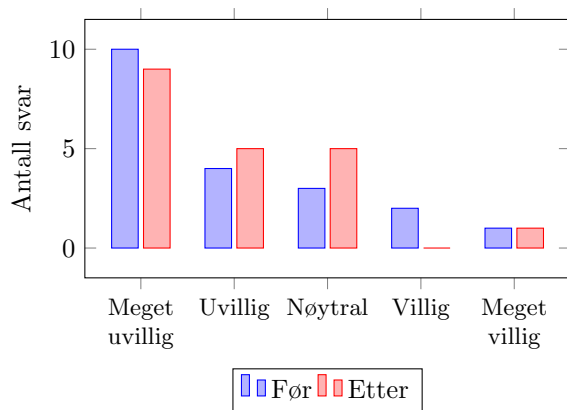


Figure D.25: (Q25) Kontaktlisten på telefonen din [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

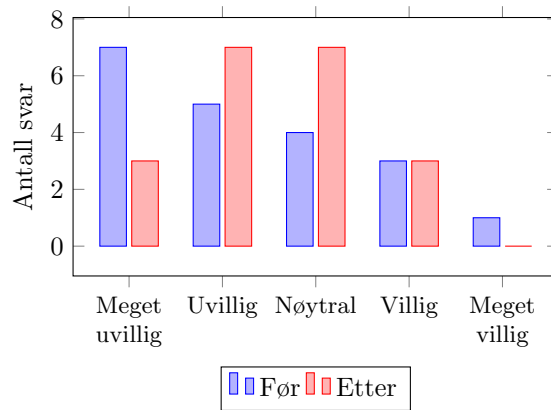


Figure D.26: (Q26) Din lokasjon [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

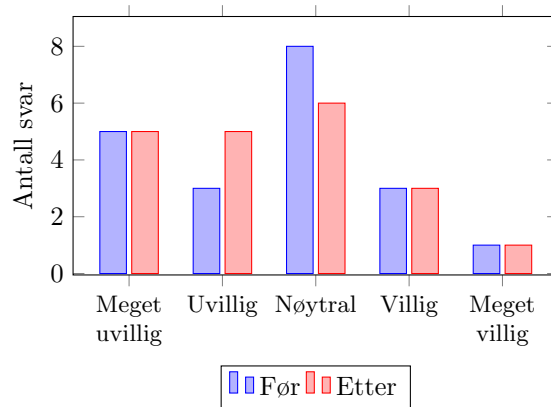


Figure D.27: (Q27) Din venneliste på sosiale medier [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

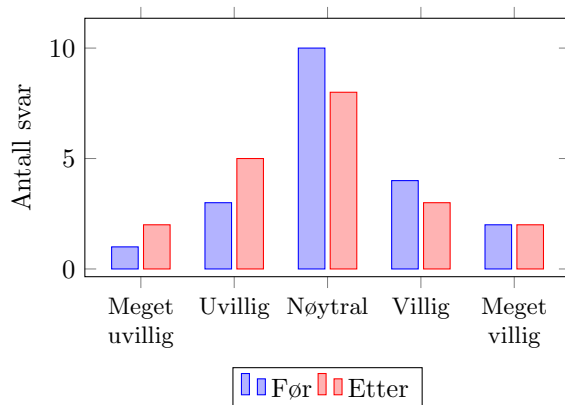


Figure D.28: (Q28) Informasjon om andre installerte apper på telefonen [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

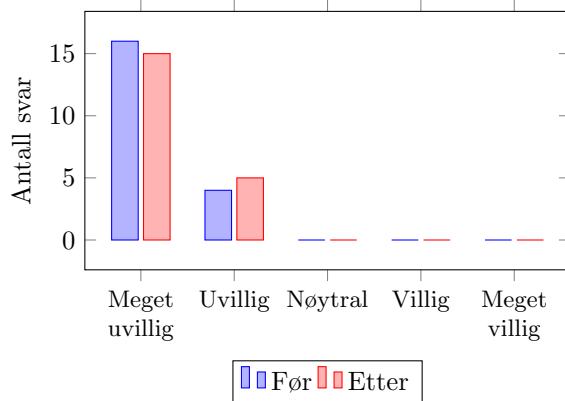


Figure D.29: (Q29) Innholdet i dine sendte og mottatte SMS-meldinger [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

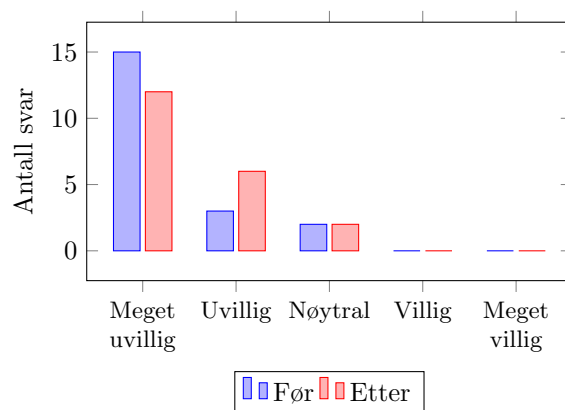


Figure D.30: (Q30) Din samtalelogg [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]

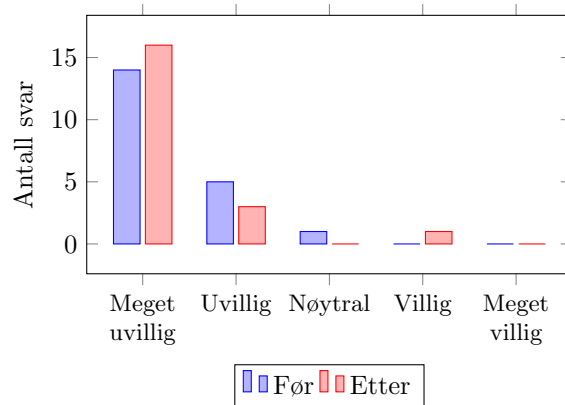


Figure D.31: (Q31) Bilder [Du skal laste ned en ny app. Den koster penger, men du kan få den billigere hvis du deler personlig informasjon. Hvilke av de følgende personlige data er du villig til å dele for å få avslag på prisen?]



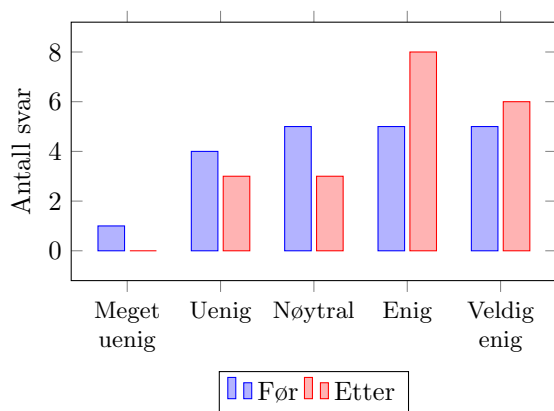


Figure D.32: (Q32) Jeg er bekymret for at apper samler for mye informasjon om meg [Er du enige med de følgende uttalelsene?]

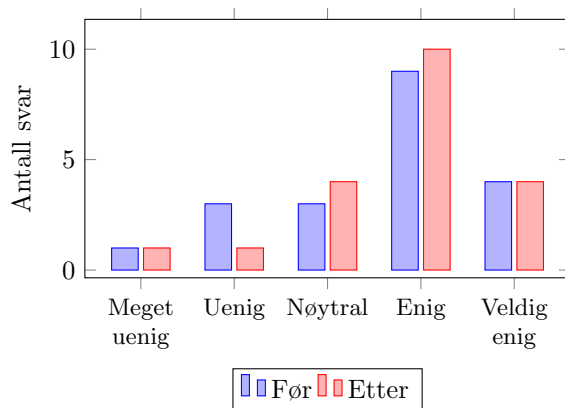


Figure D.33: (Q33) Jeg er bekymret for at selskaper bruker min informasjon til andre hensikter enn det de i utgangspunktet sier de skal [Er du enige med de følgende uttalelsene?]

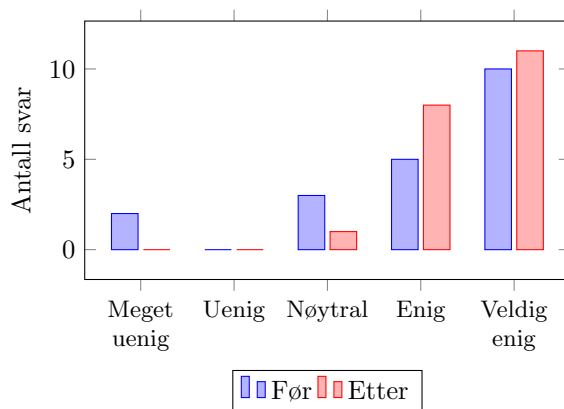


Figure D.34: (Q34) Forbrukere har mistet kontroll over hvordan og hvor mye informasjon som blir samlet inn av selskaper [Er du enige med de følgende uttalelsene?]

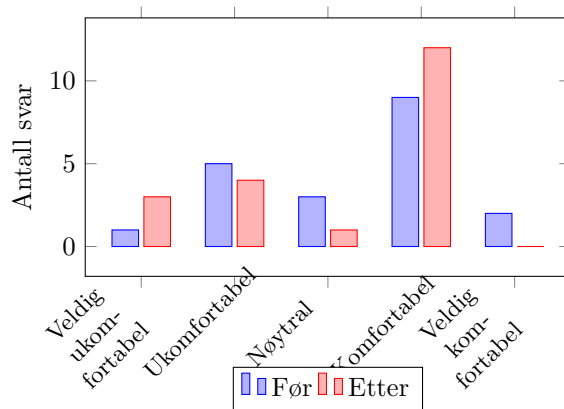


Figure D.35: (Q35) Apps- og Internettjenester tilpasset dine preferanser og behov [Er du komfortabel med bruk av din personlige data til..]

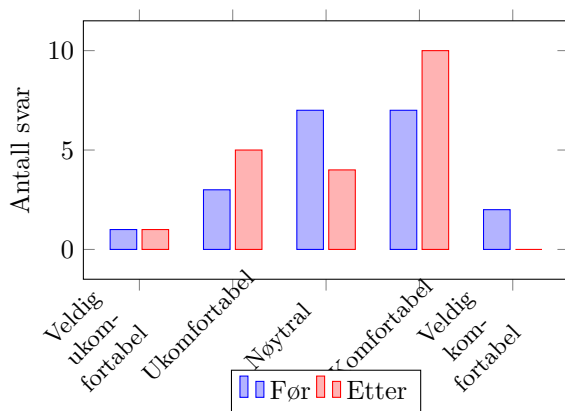


Figure D.36: (Q36) Personalisert kundeservice [Er du komfortabel med bruk av din personlige data til..]

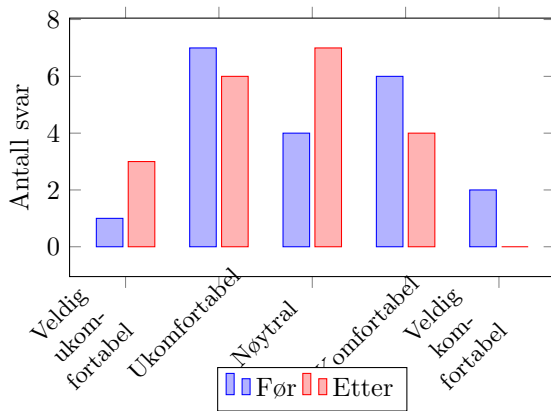


Figure D.37: (Q37) Personalisert reklame [Er du komfortabel med bruk av din personlige data til..]

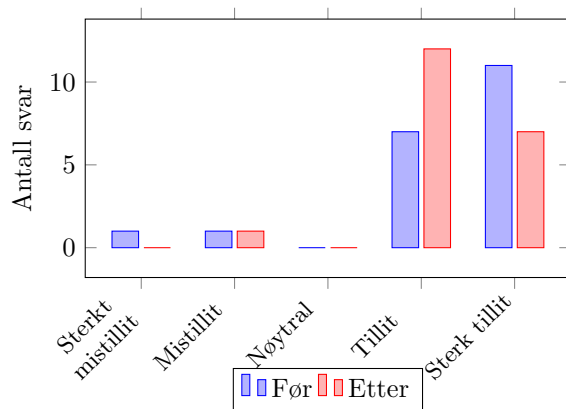


Figure D.38: (Q38) Offentlige organer(f.eks. Skatteetaten) [I hvilken grad stoler du på følgende bedrifter/institusjoner?]

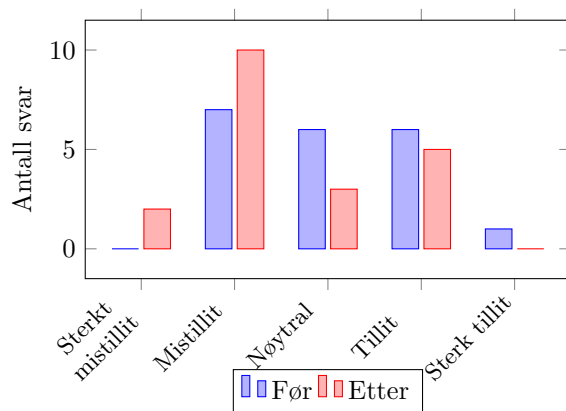


Figure D.39: (Q39) Sosiale medier(f.eks. Facebook) [I hvilken grad stoler du på følgende bedrifter/institusjoner?]

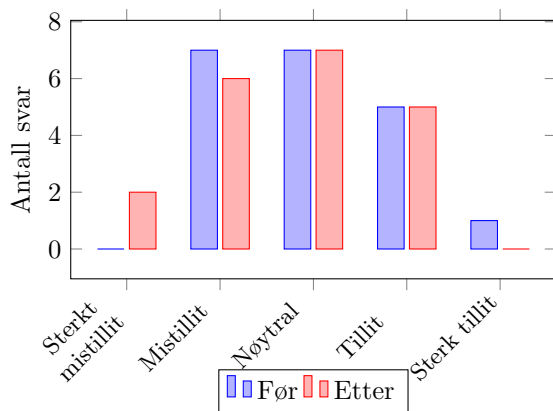


Figure D.40: (Q40) Globale Internett-bedrifter(f.eks. Google) [I hvilken grad stoler du på følgende bedrifter/institusjoner?]

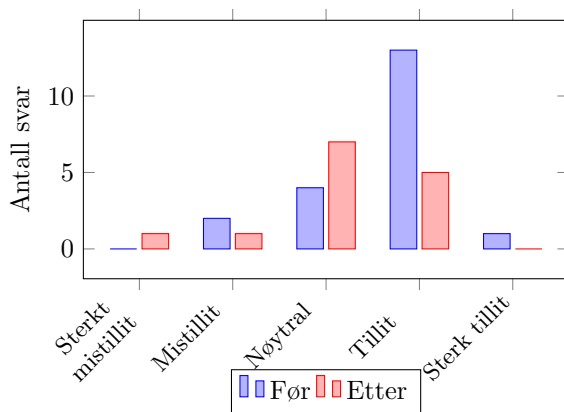


Figure D.41: (Q41) Din mobiloperatør(f.eks. NetCom) [I hvilken grad stoler du på følgende bedrifter/institusjoner?]

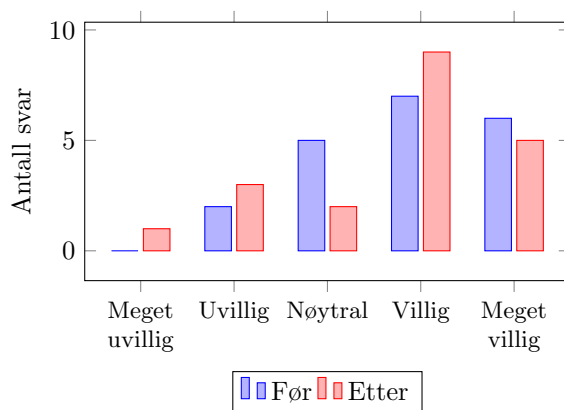


Figure D.42: (Q42) Offentlige organer (f.eks. Skatteetaten) [Ville du gitt følgende bedrifter/institusjoner tillatelse til å bruke din personlige informasjon for å i gjengjeld få personaliserte apps- og Internettjenester?]

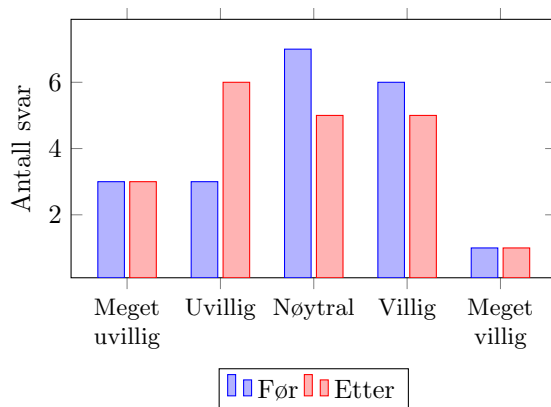


Figure D.43: (Q43) Sosiale medier (f.eks. Facebook) [Ville du gitt følgende bedrifter/institusjoner tillatelse til å bruke din personlige informasjon for å i gjengjeld få personaliserte apps- og Internettjenester?]

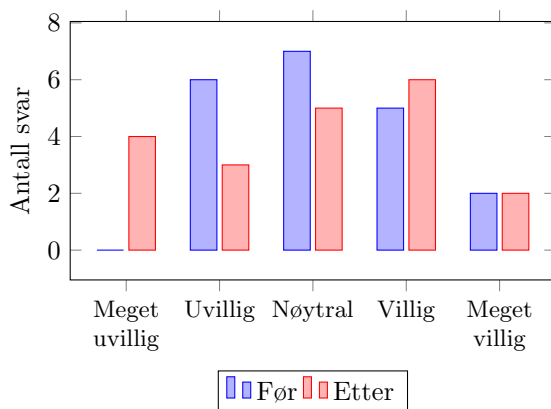


Figure D.44: (Q44) Globale Internetbedrifter (f.eks. Google) [Ville du gitt følgende bedrifter/institusjoner tillatelse til å bruke din personlige informasjon for å i gjengjeld få personaliserte apps- og Internettjenester?]

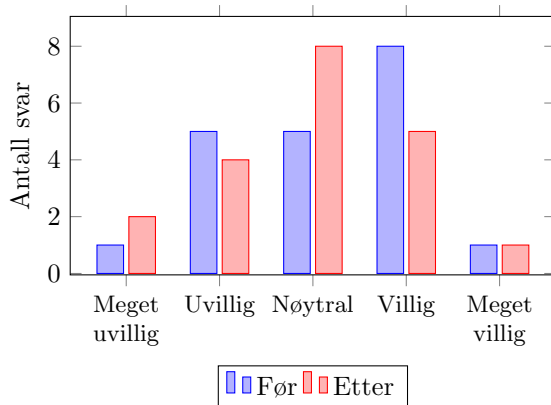


Figure D.45: (Q45) Din mobiloperatør (f.eks. NetCom) [Ville du gitt følgende bedrifter/institusjoner tillatelse til å bruke din personlige informasjon for å i gjengjeld få personaliserte apps- og Internettjenester?]

## D.2 Significance measurement ( $p < 0.05$ )

Question Number	Mean Value	T-Value	P-Value	Significant
1	0,3	1,592	0,128	FALSE
2	0,7	2,735	0,013	TRUE
3	0,2	0,83	0,417	FALSE
4	-0,05	-0,278	0,784	FALSE
5	-0,05	-0,209	0,837	FALSE
6	0,1	0,474	0,641	FALSE
7	0,25	0,887	0,386	FALSE
8	0,3	0,864	0,398	FALSE
9	0,25	0,739	0,469	FALSE
10	-0,5	-2	0,06	FALSE
11	-0,4	-1,675	0,11	FALSE
12	-0,25	-1,072	0,297	FALSE
13	-0,1	-0,379	0,709	FALSE
14	0,1	0,317	0,755	FALSE
15	-0,1	-0,503	0,621	FALSE
16	0,5	1,429	0,169	FALSE
17	0,25	1,348	0,194	FALSE
18	0,6	1,879	0,076	FALSE
19	0,15	1,172	0,256	FALSE
20	0,1	0,716	0,483	FALSE
21	0,25	1,459	0,161	FALSE
22	0,15	0,662	0,516	FALSE
23	0	0	0,999	FALSE
24	0,05	1,026	0,317	FALSE
25	0,1	0,639	0,53	FALSE
26	-0,05	-0,218	0,829	FALSE
27	0,15	0,632	0,534	FALSE
28	0,45	2,328	0,031	TRUE
29	0,05	0,582	0,567	FALSE
30	0	0	0,999	FALSE
31	0,1	1,026	0,317	FALSE
32	0,35	1,474	0,156	FALSE
33	0,05	0,278	0,784	FALSE
34	0,4	1,606	0,124	FALSE
35	0,35	1,158	0,261	FALSE
36	0,25	1,072	0,297	FALSE



---

37	0,55	2,403	0,026	TRUE
38	0	0	0,999	FALSE
39	0,4	2,236	0,037	TRUE
40	0,4	1,754	0,095	FALSE
41	0,25	1,601	0,125	FALSE
42	0,3	1,406	0,175	FALSE
43	0,35	2,155	0,044	TRUE
44	0,35	2,155	0,044	TRUE
45	0,2	0,869	0,395	FALSE