UNIVERSIDADE D
COIMBRA

Ricardo Da Silva Carvalho Mendes

Automated Privacy Protection for Mobile Devices

Tese no âmbito do Doutoramento em Engenharia Informática, Especialidade em Arquiteturas, Redes e Cibersegurança orientada pelo Professor Doutor João P. Vilela, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Março de 2022

UNIVERSIDADE D
COIMBRA

Department of Informatics Engineering
Faculty of Sciences and Technology
University of Coimbra

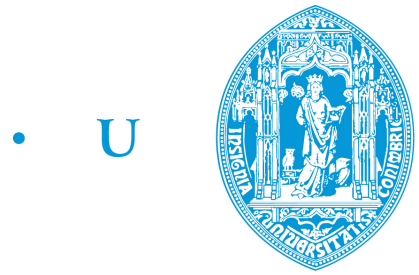
Automated Privacy Protection For Mobile Devices

Ricardo Da Silva Carvalho Mendes

PhD in Informatics Engineering
PhD Thesis submitted to the University of Coimbra

Advised by Prof. Dr. João P. Vilela

March, 2022



Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Protecção Da Privacidade Automática Em Dispositivos Móveis

Ricardo Da Silva Carvalho Mendes

Doutoramento em Engenharia Informática
Tese de Doutoramento apresentada à Universidade de Coimbra

Orientado pelo Prof. Dr. João P. Vilela

Março, 2022

This work was partially supported by the Portuguese Foundation for Science and Technology (FCT) under the PhD grant SFRH/BD/128599/2017. Additionally, through project: SWING2 (PTDC/EEI-TEL/3684/2014), funded by Fundos Europeus Estruturais e de Investimento (FEEI) through Programa Operacional Competitividade e Internacionalização - COMPETE 2020, by National Funds from FCT - Fundação para a Ciência e a Tecnologia, through project POCI-01-0145-FEDER-016753; MobiWise (P2020 SAICTPAC/001/2015) co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), European Union's ERDF (European Regional Development Fund), and the Portuguese Foundation for Science and Technology (FCT); COP-MODE, that has received funding from the European Union's Horizon 2020 research and innovation programme under the NGI_TRUST grant agreement no 825618; SNOB-5G with Nr. 045929 (CENTRO-01-0247-FEDER-045929) supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Centre (CENTRO 2020) of the Portugal 2020 framework and FCT under the MIT Portugal Program; Theia with Nr. 047264 (POCI-01-0247-FEDER-047264) supported by the European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) and Portugal 2020.

Cofinanciado por:

Acknowledgments

THE first thing that I have to acknowledge is that it a huge pleasure to write this section. I know that starting with such statement is unconventional, but reaching a state where the thesis is (almost) ready and having nothing but thanks to write is a relief. Having taken that out of the way, let me start with the proper acknowledgments.

I would like to start by deeply thanking my advisor Prof. João Paulo Vilela without whom this thesis would not exist. From proposing this opportunity to his full support at every stage of this work, I am forever grateful. You are, without a doubt a great advisor and a mentor.

I extend my gratitude to Prof. Alastair Beresford, which not only enabled me to visit Cambridge as a scholar, but also supported and integrated me throughout the experience. During that time, which I will forever cherish, I met some great friends: Jovan Powar, Stephan Kollmann, Jiexin "Stan" Zhang and Yuelin "Evelyn" Zhou.

I would also like to thank everyone that directly or indirectly helped my thesis or my well-being. First, I am grateful for the team behind the LCT laboratory, particularly Prof. Marília Curado and Prof. Edmundo Monteiro, for fostering a great environment that extends beyond work. Second, I thank my colleagues, that I now have the pleasure to call friends, that directly collaborated with me. Namely, Mariana Cunha and André Brandão. I hope that in the future we will continue to work together. Finally, I thank the old friends (from "Maleita" and Electrical and Computer Engineering) that I can always count on and the new friends that I made along the way, particularly in the LCT. I refrain from listing names in this later group as the size of it would put me at risk of forgetting someone. However, I do need to shout-out two particular persons from this latter group with whom I luckily shared most of this journey: David Abreu and Karima Velasquez. You two are awesome people and I will always consider you as brother and sister.

Finally, I would like to thank my family and Bruna for always being available, supportive and loving. This adventure would not have been meaningful without them, and I will always be grateful.

Thank you!

Abstract

The pervasiveness of smart devices and the always connected paradigm has fostered applications that benefit from sensing the environment to provide contextualized services to users. This paradigm has undeniably made lives easier by breaking language barriers, providing effective navigation routing and constant communication and availability, to name a few. For all of these services however, a significant amount of information is exchanged with service providers, some of which can be considered private and sensitive. Furthermore, after being collected, users have limited control over their data.

To preserve privacy before the data is sent to service providers, mobile devices employ permission managers. These mechanisms allow users to control access to sensitive resources and data by the installed applications. However, currently deployed managers have been shown inefficient at both protecting and warning users against the possible risks. Specifically, the main drawback of current systems lies in the number of permissions that are automatically accepted. After being allowed once, applications can generally access the same resource at any time, without user consent or even awareness. These automatically accepted permissions can violate the privacy preferences of the user at each current context, i.e., they violate privacy's contextual integrity, and therefore contradict users expectations.

Automation in permission managers is paramount as the number installed applications and respective permissions renders inefficient constantly asking the user. In fact, it would lead users to became fatigued and therefore to promptly dismiss the privacy notices. Hence, the automation must be *smart* by taking into account the intrinsic nature of privacy, namely, privacy's subjectiveness to each individual and the contextual dependency of such preferences.

The main goal of this thesis is to improve the state-of-the-art in privacy for mobile devices through personalized and context-aware automation. Towards this end, we start by performing a field study to collect permission decisions, their surrounding context and respective user expectations, a dataset that we make available to the community. This data shows the ineffectiveness of current permission managers based on runtime permissions, as this would have resulted in a violation of privacy for 15% of requests. Additionally, almost 50% of requests were unexpected to users, thus highlighting a strong misalignment between apps' practices and user expectations. Furthermore, privacy decisions see the strongest correlation with user expectation, however, both the expectation and its importance in the decision is subjective to each individual.

Using the collected data, we train personalized and context-aware models for the prediction of privacy decisions by taking into consideration user expectation and the context of the user and of the phone. Our best model achieves an Area Under the Receiving Operation Curve (ROC AUC) of 0.957 and an F1 score of 0.924.

Furthermore, such model reduces the number of privacy violations by 59.5%, when compared to a standard Android handset. Without user expectation, we achieve a ROC AUC of 0.898 and an F1 score of 0.886, a model that reduces the privacy violations by 27.9%.

Another crucial drawback of existing permission managers is the limited control over the trade-off between privacy and utility. Specifically, the binary option of allowing or denying permissions corresponds to extreme situations where the user either has maximum utility and no privacy, or maximum privacy and no utility, respectively. Obfuscation can be added to the permission manager to provide users with a fine-grained control over this trade-off. Two challenges arise in this subject: obfuscation techniques are data type dependent, and therefore different techniques would be required for each sensitive permission; tuning the obfuscation mechanism for each situation at each permission request, or using static configurations could result in ineffective privacy and/or utility depending on each situation/context.

Focusing on location data, a prevalent and sensitive type of data in mobile devices, we performed an empirical evaluation on the effect of varying frequency of reports on location privacy mechanisms based on differential privacy, the de facto privacy standard. This empirical study reveals that under sporadic release of location data, reports can be considered independent. However, under continuous location sharing, correlations between successive reports degrade the user privacy, thus requiring Location Privacy-Preserving Mechanisms (LPPMs) that take this aspect into consideration. Another finding from this study is that a poorly configured LPPM can result in no effective privacy. These two results served as motivation to propose a novel formal notion for the continuous release of location data based on differential privacy and termed Velocity-Aware Geo-Indistinguishability (VA-GI).

A VA-GI LPPM is presented that automatically adjusts for privacy or utility depending on the velocity of the user and frequency of reports. This automated adjustment is essential for the integration of such mechanism in a permission manager, while requiring minimal interaction from the user, e.g., for tuning parameters. Furthermore, this proposal simplifies its configuration by requiring only two user-set parameters, the privacy budget and a multiplier, and allows for the personalization of the LPPM by using data from a specific driver or from all drivers in a particular area, thus enabling personalization from a fine-grained user-level up to more general region-level (e.g. city or district). Our empirical simulations with real data show the effectiveness of the VA-GI LPPM in automatically adjusting the privacy and utility, in fact outperforming existing differentially private LPPMs.

Keywords: Mobile Devices, Permission Managers, Personalized Privacy, Context-Awareness, Location Privacy, User Expectation.

Resumo

A adoção em massa de dispositivos móveis inteligentes e o paradigma da conectividade permanente levaram ao desenvolvimento de aplicações que oferecem serviços personalizados com base em informação que recolhem sobre o contexto do utilizador (p.e. localização). Este paradigma facilitou o quotidiano dos utilizadores através de serviços como navegação e identificação de pontos de interesse, bem como ao ajudar a quebrar barreiras linguísticas, entre outros. No entanto, uma quantidade significativa de informação é enviada para os fornecedores destes serviços, parte da qual pode ser considerada privada e sensível. Além disso, os utilizadores têm, em geral, um controlo limitado sobre os seus dados após estes serem recolhidos.

Para preservar a privacidade dos utilizadores antes de os dados serem enviados para os fornecedores de serviços, os dispositivos móveis possuem gestores de permissões que permitem ao utilizador controlar o acesso das aplicações aos recursos e dados sensíveis. No entanto, os gestores de privacidade atuais são pouco eficazes a proteger e a notificar os utilizadores sobre os potenciais riscos de privacidade. Existe um elevado número de permissões que são automaticamente concedidas, em particular, após terem sido autorizadas uma primeira vez pelo utilizador, as aplicações podem, em geral, aceder ao mesmo recurso a qualquer momento, sem consentimento ou mesmo perceção por parte do utilizador. Estas permissões automaticamente concedidas podem violar as preferências dos utilizadores, contradizendo as suas expectativas que podem variar de acordo com o contexto de utilização.

A automação dos gestores de permissões é fulcral, uma vez que o elevado número de aplicações instaladas e respetivas permissões torna a sua gestão individual inviável, caso o utilizador tivesse que responder manualmente a todos os pedidos, o que teria como consequência a dessensibilização do utilizador para com os avisos de privacidade. Desta forma, a automação destes sistemas deve ser inteligente, garantindo que as características intrínsecas à noção de privacidade sejam respeitadas, nomeadamente, a sua subjetividade em relação a cada indivíduo e a sua dependência do contexto.

O objetivo principal desta dissertação é melhorar o estado-da-arte da privacidade em dispositivos móveis, através de automação personalizada e ciente do contexto. Para tal, começámos por realizar uma campanha de recolha de dados para coletar informação acerca das decisões de acesso a permissões pelos utilizadores, bem como o respetivo contexto e as expectativas dos utilizadores. Devido à inexistência de um *dataset* público semelhante, disponibilizamos os dados recolhidos à comunidade científica. O nosso *dataset* demonstra a ineficácia dos atuais gestores de permissões baseados em *runtime*, i.e. que concede permissões às aplicações da primeira vez que são pedidas, mantendo-as para futuras utilizações. Este modo de gestão de permissões que é o gestor predefinido do

Android resulta numa violação da privacidade em 15% das permissões respondidas pelos nossos participantes. Além disso, quase 50% de todos os pedidos de permissões foram considerados como inesperados pelos utilizadores, evidenciando uma forte divergência entre as práticas das aplicações e as expetativas do utilizador. Adicionalmente, a *feature* da expectativa do utilizador foi identificada como aquela com a maior correlação com as decisões de privacidade dos utilizadores, realçando a sua importância nas decisões de privacidade adotadas. No entanto, tanto a expectativa como a sua importância para a decisão foi identificada como sendo subjetiva a cada indivíduo.

Com os dados recolhidos, treinámos modelos personalizados e cientes do contexto para previsão das decisões de privacidade (aceitar ou rejeitar acesso às permissões), tendo em consideração *features* de expectativa e contexto do utilizador, bem como do contexto do dispositivo. O nosso melhor modelo de predição atinge um área abaixo da curva *Receiver Operator Characteristic* (ROC AUC) de 0.957 e um F1 score de 0.924. Mais ainda, este modelo reduz a quantidade de violações de privacidade em 59.5% em comparação com o gestor de permissões predefinido do Android baseado em permissões em *runtime*. Sem utilização da *feature* da expectativa (que requer *input* do utilizador), o modelo atinge ainda assim uma ROC AUC de 0.898, um F1 score de 0.886 e uma redução no número de violações de privacidade de 27.9%.

Outra importante limitação dos gestores de privacidade existentes é o controlo limitado sobre o compromisso entre a privacidade e a utilidade. Especificamente, a opção binária de permitir ou negar permissões corresponde aos extremos onde o utilizador tem máxima utilidade e nenhuma privacidade, ou máxima privacidade e nenhuma utilidade, respetivamente. A ofuscação de dados é uma medida válida para possibilitar aos utilizadores um controlo mais fino sobre este balanço. Dois desafios aparecem neste contexto: as técnicas de ofuscação são tipicamente específicas a cada tipo de dados e, portanto, técnicas diferentes são necessárias para cada tipo de permissão/dados; a configuração das técnicas de ofuscação para cada situação/contexto pode resultar num nível de proteção da privacidade ou num ajuste da utilidade, ineficazes.

Focando nos dados de localização, face à sua prevalência em dispositivos móveis e à sensibilidade dos mesmos, realizámos uma avaliação empírica do efeito da variação da frequência da partilha de dados de localização na eficácia dos mecanismos de proteção de privacidade de localização baseados em privacidade diferencial, a atual noção de privacidade de informação dominante. Este estudo revelou que a independência dos dados de localização pode ser efetivamente assumida no caso da partilha esporádica. No entanto, sob a partilha contínua da localização, a correlação entre localizações sucessivas degrada a privacidade do utilizador, requerendo assim mecanismos de preservação da privacidade de localização (LPPMs) que tenham em conta essa mesma correlação. A análise demonstrou ainda que uma inadequada configuração de um LPPM pode resultar numa perda significativa do nível de privacidade. Estes dois resultados serviram de motivação para a proposta de um nova noção formal de privacidade para a partilha contínua de dados de localização, baseada em privacidade diferencial, designada Velocity-Aware Geo-Indistinguishability (VA-GI).

Com base na análise referida, foi então desenvolvido o novo LPPM VA-GI que ajusta automaticamente o nível de privacidade ou utilidade em função da velocidade do utilizador e da frequência da partilha de dados de localização. Este ajuste automático é essencial para a integração de um LPPM num gestor de privacidade, para que a interação requerida ao utilizador, por exemplo para o ajuste de parâmetros, seja mínima. Para além disso, esta proposta simplifica a configuração do LPPM, requerendo apenas dois parâmetros: o orçamento de privacidade e um multiplicador, que servem para definir os limites máximos e mínimos do nível de privacidade e utilidade. Este LPPM permite ainda a personalização do mecanismo através do uso de dados de um único condutor ou de todos os condutores de uma dada área geográfica (por exemplo, uma cidade ou distrito). A avaliação com trajetórias reais demonstra a eficácia do VA-GI LPPM no ajuste automático dos níveis de privacidade e utilidade, resultando num desempenho superior face a outros LPPMs baseados em privacidade diferencial.

Palavras-chave: Dispositivos Móveis, Gestores de Privacidade, Privacidade Personalizável, Ciente do Contexto, Privacidade de Dados de Localização, Expectativa do Utilizador.

Foreword

The work detailed in this thesis was performed at the Laboratory of Communications and Telematics (LCT) of the Center for Informatics and Systems of the University of Coimbra (CISUC), and at the Computer Laboratory of the Department of Computer Science and Technology of the University of Cambridge. The work undertaken in the course of this thesis resulted in the following publications:

Journal papers:

- Mendes, R. and Vilela, J. P. (2017). Privacy-preserving data mining: Methods, metrics, and applications. *IEEE Access*, 5:10562–10582
- Mendes, R., Cunha, M., and Vilela, J. P. (2020). Impact of frequency of location reports on the privacy level of geo-indistinguishability. *Proceedings on Privacy Enhancing Technologies*, 2020(2):379 – 396

Conference papers:

- Mendes, R., Cunha, M., Vilela, J. P., and Beresford, A. R. (2022b). Enhancing user privacy in mobile devices through prediction of privacy preferences. In *Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, pages 153–172. Springer
- Mendes, R., Brandão, A., Vilela, J. P., and Beresford, A. R. (2022a). Effect of user expectation on mobile app privacy: A field study. In *2022 IEEE international conference on pervasive computing and communications (PerCom)*, pages 207–214. IEEE
- Mendes, R., Cunha, M., and Vilela, J. P. (2023). Velocity-aware geo-indistinguishability. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY)*. ACM. In press
- Mendes, R. and Vilela, J. P. (2018). On the effect of update frequency on geo-indistinguishability of mobility traces. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '18, page 271–276, New York, NY, USA. Association for Computing Machinery

The work undertaken by the candidate has sparked further research that was pursued by other students in collaboration with the candidate, culminating in two master thesis and several collaboration papers as identified below.

Supervisor of MSc Thesis:

- Brandão, A. (2021). Prediction of privacy preferences with user profiles: A federated learning approach. Master's thesis, Universidade do Porto

- (Unofficial Supervision) Cunha, M. (2019). Privacy-preserving mechanisms for location traces. Master's thesis, Universidade de Coimbra

Cooperation papers:

- Brandão, A., Mendes, R., and Vilela, J. P. (2022). Prediction of mobile app privacy preferences with user profiles via federated learning. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*. ACM. In press
- Brandão, A., Mendes, R., and Vilela, J. P. (2021). Efficient Privacy Preserving Distributed K-Means for Non-IID Data. In *Advances in Intelligent Data Analysis XIX*, pages 439–451, Cham. Springer International Publishing
- Cunha, M., Mendes, R., and Vilela, J. P. (2021). A survey of privacy-preserving mechanisms for heterogeneous data types. *Computer Science Review*, 41:100403
- Cunha, M., Mendes, R., and Vilela, J. P. (2019). Clustering geo-indistinguishability for privacy of continuous location traces. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, pages 1–8. IEEE

This work was framed and conducted within the context of the following projects and grants:

SWING2 - Securing Wireless Networks with Coding and Jamming; funded by Fundos Europeus Estruturais e de Investimento (FEEI) through Programa Operacional Competitividade e Internacionalização - COMPETE 2020 (PTDC/EEI-TEL/3684/2014). The main goal of the SWING2 (Securing Wireless Networks with Coding and Jamming) project is to advance the state-of-the art on physical-layer techniques to secure wireless networks under eavesdropper adversaries that aim to overhear unintended information.

SNOB-5G - financed by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Centre (CENTRO 2020) of the Portugal 2020 framework and FCT under the MIT Portugal Program. SNOB-5G with Nr. 045929 (CENTRO-01-0247-FEDER-045929). SNOB-5G will carry out the research and development of a self-optimized, intelligent and fault-tolerant wireless backhaul solution for 5G networks that will empower cities, as neutral hosts, in the 5G business, by promoting total connectivity with a high-bandwidth, capacity and latency requirements, capable of accommodating new and innovative urban services. The new and robust technological solution for 5G deployment may use existing urban furniture to overcome the current limitations related to the availability and installation costs of wired connections that support the backhaul communication.

MobiWise - financed by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework, and by the Portuguese Foundation for Science and Technology (FCT)(OE); Project, MOBIWISE,

POCI-01-0145-FEDER-016426. MobiWise aims to enhance mobility in the cities, both for commuters and for tourists, through the development of a 5G platform that encompasses an access infrastructure filled with sensors, people and vehicles.

COP-MODE - COntext-aware Privacy protection for MObile DEvices; funded through the European Union's Horizon 2020 research and innovation programme under the NGI_TRUST grant agreement no. 825618. The COP-MODE project is a research initiative led by the University of Coimbra, the University of Porto and the University of Cambridge aiming at advancing the state-of-the art on privacy protection mechanisms for mobile devices operating in ubiquitous computing environments.

Theia - financed by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) and Portugal 2020; Project nº 047264, POCI-01-0247-FEDER-047264. Theia aims to research and develop new algorithms for autonomous driving.

Ph.D. grant - Foundation for Science and Technology (FCT) (SFRH/BD/128599/2017).

Contents

List of Figures

List of Tables

Acronyms

ADWIN ADaptive WINdowing

AE Adversary Error

app Application

API Application Programming Interface

CA Context-Awareness

CCPA California Consumer Privacy Act

CDF Cummulative Density Function

DDM Drift Detection Method

Geo-Ind Geo-Indistinguishability

GDPR General Data Protection Regulation

GPS Global Positioning System

IID Independent and Identically Distributed

IoT Internet of Things

KDE Kernel Density Estimation

LBS Location-Based Service

LPPM Location Privacy-Preserving Mechanism

ML Machine Learning

MM Map-Matching

NPM Naive Permission Manager

OS Operating System

P2P Peer-to-Peer

PM Permission Manager

PL Planar Laplace

PoI Points-of-Interest

PPDM Privacy-Preserving Data Mining

PIR Private Information Retrieval

PDF Probability Density Function

PEBA Profile-Estimation Based Attack

RBF Radial Basis Function

ROC AUC Area Under the Receiving Operation Curve

SVM Support Vector Machines

VA-GI Velocity-Aware Geo-Indistinguishability

Chapter 1.

Introduction

Contents

The pervasiveness of smart devices and the always on and always connected paradigm has fostered applications that benefit from sensing the environment to provide contextualized services to its users. However, this constant collection and flow of information presents severe privacy and security risks, such as the possibility of disclosure through data breaches. This calls for systems that empower users with control over their data, allowing them to benefit from the technology and at the same time retain a certain degree of privacy. This thesis tackles the development and enhancement of privacy mechanisms through automation, personalization and context-awareness. This chapter introduces this subject by presenting some background, motivation and the objectives of this work, and enumerating the contributions.

1.1 Background and Motivation

Designing privacy-preserving mechanisms is challenging due to the non-existence of a single universal privacy definition [Langheinrich, 2009] and its strong dependence on people's preferences, beliefs and on the context of privacy decisions [Acquisti et al., 2015]. Often, users are unaware of data collection risks [Felt et al., 2012] and intrusive practices [Balebako et al., 2013; Shklovski et al., 2014], thus trading privacy for small benefits [Acquisti et al., 2015]. An illustrative yet pervasive example is that of loyalty cards where users receive monetary discounts at the expenses of revealing buying patterns [Acquisti and Grossklags, 2008].

Due to their inherent capacity to collect high quantities of sensitive data, smartphones have implemented permission managers to give users control over which applications can access certain device resources, including sensors and data. While researchers have proposed several improvements, current industry privacy managers are still ineffective at protecting users' privacy [Felt et al., 2012; Wijesekera et al., 2015; Mendes et al., 2022a]. For example, permissions requests are prompted at the first time an application requires such access. While this allows to contextualize the request by the need for a certain functionality, after being accepted once, applications can generally access the resources at any time and for any purpose even without users noticing [Almuhimedi et al., 2015; Wijesekera et al., 2015]. These subsequent automatically accepted permissions thus potentially violate the preferences of the user at these newer contexts [Mendes et al., 2022a].

A naive solution is to request permission access on every use. However, applications can make hundreds of permission requests per day [Wijesekera et al., 2015; Mendes et al., 2022b], thus leading to warning fatigue and consequently, poor privacy choices [Felt et al., 2012]. Towards solving this problem, researchers proposed automated solutions through either personalization [Liu et al., 2016], context-awareness [Zavala et al., 2011], and more recently both [Olejnik et al., 2017]. Personalized permission managers take into consideration users' personal

preferences toward privacy. However, these approaches are often static and do not take into account the user and device context, which has been shown to greatly influence privacy decisions [Acquisti et al., 2015]. Personalized and context-aware permission managers are thus required. Nevertheless, due to the high context dynamism of mobile devices, precisely define context and subsequently enforce personalized privacy is challenging.

The vast majority of research proposals in permission managers focus on either allowing or denying access to resources. An essential problem arises from this lack of control over the trade-off between privacy and utility. Denying permissions has maximum privacy but no utility, that is, it prevents users from accessing functionality. Conversely, allowing access has maximum utility, but no privacy, and after being collected, possibly untrustworthy data collectors get full access and control over such data [Mendes and Vilela, 2017]. Towards achieving a better trade-off between privacy and utility at collection time, that is, before the data is sent to service providers, obfuscation techniques can be used. Obfuscation is the purposefully degradation of data quality in order to retain a certain degree of privacy, thus still allowing for some disclosure, and consequently, benefit from the functionality.

Obfuscation is, however, highly data dependent [Cunha et al., 2021]. Specifically, different types of data require different methods, and in the context of utility, even applications using the same data type may have different data quality requirements. In this thesis, a focus on location data is given due to the relevance and sensitiveness of this type of data in mobile devices [Huang et al., 2018].

The attractiveness of sharing location data is related to Location-Based Services (LBSs), which have proliferated with the pervasiveness of mobile devices [Huang et al., 2018]. LBS providers rely on users' current location to provide a geo-temporal contextualized service. However, mobility traces are highly sensitive as this type of information can disclose habits, social connections, points of interest and even health conditions [Krumm, 2009]. In fact, it has been shown that mobility traces are highly unique [De Montjoye et al., 2013], that Points-of-Interests (PoIs) act as quasi-identifiers [Bettini et al., 2005; Primault et al., 2014], and that individual's traces are extremely predictable given past location history [Song et al., 2010]. The sensitiveness of this type of data must thus be taken into account when designing location privacy obfuscation techniques.

This thesis intends to contribute to the state of the art in privacy in mobile devices by proposing privacy protection mechanisms that are automated, to avoid warning fatigue and minimize user interaction, personalized, to take into consideration users' privacy preferences, and context-aware, to account for privacy's context dependency. Such system shall also resort to obfuscation to increase the control over the trade-off between privacy and utility. Since obfuscation is highly data dependent and disclosure (utility) is application specific, a focus on an essential type of data in mobile devices will be given, namely, location data. The integration of obfuscation in a permission manager further requires simple and automatic configuration to varying situations, as to avoid misconfigurations and poor privacy or utility depending on the context

surrounding the data practice.

1.2 Objectives

The main objective of this research is to enhance privacy in mobile devices. Towards this general goal, the following objectives have been established:

- Analyze privacy decisions in mobile devices within each surrounding context. Validate the need and potential venues for the development of *smarter* automated permission managers;
- Develop methods for an automated, personalized and context-aware permission manager that improves on existing limitations;
- Analyze the characteristics of location data towards evaluating the potential incorporation of location obfuscation in a permission manager. Namely, measure the potential privacy disclosure that occurs in dynamic contexts, for instance, the difference between releasing location data sporadically or continuously; and
- Develop a novel Location Privacy-Preserving Mechanism (LPPM) that automatically adapts to varying contexts/situations at which the data is collected.

The last two objectives focused on location data as each type of data requires different obfuscation techniques, specially since their nature can impact the attacks or the amount of data that is disclosed. However, similar methodology to the one undertaken in this work can be followed to develop effective obfuscation mechanisms for each of the sensitive data types that mobile devices collect.

1.3 Contributions

Taking into consideration the previously described goals, this thesis has produced the following contributions:

- Collected a dataset of permission decisions, surrounding context and respective user expectations from 93 participants. We made this dataset available to interested researchers [Mendes, 2021a]. This dataset validated our premise that the permission manager in current Android devices (up to Android 9) is ineffective at protecting user privacy. It further showed a strong misalignment between user expectations and app practices.
- Development of personalized and context-aware predictive models that automate privacy decisions with higher performance than the current Android permission manager. Specifically, by taking into consideration the expectation and the phone and user contexts we achieve a Area Under the Receiving Operation Curve (ROC AUC) of 0.96 and an F1 score of 0.92. Without user expectation, we achieve a ROC AUC of 0.9 and an F1 score of 0.88. These two solutions achieve a reduction on the the number of privacy violations of approximately 60% and 28%, respectively.

- We performed a study on the impact of the frequency of location reports on the privacy-utility trade-off. To empirically motivate the need for automatic dynamic adjustment of the privacy-utility trade-off in accordance with varying frequency of location reports. Specifically, as the frequency increases, so does the correlation between subsequent reports and, consequently, the privacy loss. Furthermore, according to our analysis, a misconfigured parameter can result in no effective privacy.
- We propose Velocity-Aware Geo-Indistinguishability (VA-GI), a generalization of geo-indistinguishability to location traces. VA-GI allows for the development of LPPMs that automatically adjust their privacy and utility in accordance with the frequency of reports and user velocity, thus using these two variables as a metric for the correlation between points. Furthermore, we propose a VA-GI LPPM that requires only 2 user-set parameters, the privacy budget and a multiplicative factor, as to facilitate tuning and therefore mitigate misconfigurations. Our simulations with real data show that the VA-GI LPPM outperforms existing geo-indistinguishable LPPMs regarding the privacy-utility adaptability.

These contributions have been published in five international conference papers and one journal papers as well as motivated further research that was supervised by the candidate and culminated in two masters' thesis and four cooperation papers.

1.4 Bibliographic Note

The vast majority of the work detailed in this thesis has been published in international conferences and journals as follows.

The literature review presented in Chapter 2 was based on the following publications:

- Overview on privacy, privacy-preserving mechanisms in the data life-cycle, and the trade-off between privacy and utility: Mendes, R. and Vilela, J. P. (2017). Privacy-preserving data mining: Methods, metrics, and applications. *IEEE Access*, 5:10562–10582
- State of the art in permission managers, including context-awareness and personalization:
  - Mendes, R., Brandão, A., Vilela, J. P., and Beresford, A. R. (2022a). Effect of user expectation on mobile app privacy: A field study. In *2022 IEEE international conference on pervasive computing and communications (PerCom)*, pages 207–214. IEEE
  - Mendes, R., Cunha, M., Vilela, J. P., and Beresford, A. R. (2022b). Enhancing user privacy in mobile devices through prediction of privacy preferences. In *Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, pages

153–172. Springer

- State of the art in location privacy and respective privacy-preserving mechanisms:
  - Mendes, R., Cunha, M., and Vilela, J. P. (2023). Velocity-aware geo-indistinguishability. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY)*. ACM. In press
  - Mendes, R., Cunha, M., and Vilela, J. P. (2020). Impact of frequency of location reports on the privacy level of geo-indistinguishability. *Proceedings on Privacy Enhancing Technologies*, 2020(2):379 – 396
  - Mendes, R. and Vilela, J. P. (2018). On the effect of update frequency on geo-indistinguishability of mobility traces. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '18, page 271–276, New York, NY, USA. Association for Computing Machinery

The contributions from Chapter 3 have been reported in the following publications:

- Presentation of the dataset and detailed analysis on the effect of user expectation on privacy decisions: Mendes, R., Brandão, A., Vilela, J. P., and Beresford, A. R. (2022a). Effect of user expectation on mobile app privacy: A field study. In *2022 IEEE international conference on pervasive computing and communications (PerCom)*, pages 207–214. IEEE.
- Analysis on the impact of the context in privacy decisions and the development of prediction models towards enhancing automated privacy in mobile devices: Mendes, R., Cunha, M., Vilela, J. P., and Beresford, A. R. (2022b). Enhancing user privacy in mobile devices through prediction of privacy preferences. In *Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, pages 153–172. Springer.

Additionally, the work in the referred chapter has sparked the idea of creating privacy profiles and the prediction models with privacy guarantees, even against the entity that trains these models. Tackling this problem, a master thesis was supervised by the candidate, flourishing in the following two publications:

- Brandão, A., Mendes, R., and Vilela, J. P. (2021). Efficient Privacy Preserving Distributed K-Means for Non-IID Data. In *Advances in Intelligent Data Analysis XIX*, pages 439–451, Cham. Springer International Publishing. This paper proposes a novel privacy-preserving clustering algorithm that is efficient and robust to non-Independent and Identically Distributed (IID) Data. While the idea of building privacy profiles while retaining the privacy of users was the source for this proposal, the technique is generic in the sense that it can be applied to any use case. However, the robustness to non-IID data is vital for the profiling use-case as the

privacy preferences of each user is considered a single point in the clustering (profiling) algorithm. Therefore, since privacy preferences vastly diverge [Liu et al., 2014], the distribution of points (preferences) would potentially result in an extreme non-IID case.

- Brandão, A., Mendes, R., and Vilela, J. P. (2022). Prediction of mobile app privacy preferences with user profiles via federated learning. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*. ACM. In press. This paper evaluates the feasibility of building privacy profiles with privacy guarantees using the proposed clustering algorithm from the previous paper, and using federated learning to build the prediction models with the profiles. This approach preserves privacy even against the collecting entity, while achieving similar performance to the non-private approach described in this chapter.

The work detailed in Chapter 4 originated the following publications:

- Mendes, R. and Vilela, J. P. (2018). On the effect of update frequency on geo-indistinguishability of mobility traces. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '18, page 271–276, New York, NY, USA. Association for Computing Machinery
- Mendes, R., Cunha, M., and Vilela, J. P. (2020). Impact of frequency of location reports on the privacy level of geo-indistinguishability. *Proceedings on Privacy Enhancing Technologies*, 2020(2):379 – 396

It has additionally served as motivation for the LPPM proposed in Chapter 5 and for a master thesis entitled "Privacy-Preserving Mechanisms for Location Traces" that was unofficially co-advised by the candidate and culminated in the publication of the following paper: Cunha, M., Mendes, R., and Vilela, J. P. (2019). Clustering geo-indistinguishability for privacy of continuous location traces. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, pages 1–8. IEEE.

Finally, Chapter 5 culminated in the paper: Mendes, R., Cunha, M., and Vilela, J. P. (2023). Velocity-aware geo-indistinguishability. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY)*. ACM. In press.

1.5 Outline of the Thesis

The remainder of this thesis is organized into five chapters. Chapter 2 contextualizes this work in the current state-of-the-art in privacy in mobile devices, focusing on automation, personalization, context-awareness and location privacy.

Chapter 3 details our field study, where we collect permission decisions, the surrounding context and respective user expectations. Using the data we explore the relation between context, user expectation and privacy decisions. Finally,

we develop prediction models that use these features to automate privacy decisions.

Chapter 4 presents an empirical analysis on the impact of the frequency of reports on the privacy level of location traces. The obtain results then motivate Chapter 5 which proposes a novel notion based on differential privacy towards preserving privacy under continuous location reports. A comparative analysis with other existing differentially private mechanisms is conducted and a generalization of the mechanism that facilitates wide deployment is evaluated.

Chapter 6 concludes this thesis with a summary of the work and respective contributions, and provides future work venues.

Chapter 2.

Privacy In Mobile Devices

Contents

The objective of this chapter is to provide a thorough literature review on privacy in mobile devices. Throughout this work, a stronger focus in these type of devices is given due to their pervasiveness, ubiquity and rich sensory capabilities.

The remainder of this chapter is organized as follows. Section 2.1 presents background on privacy, including its (lack of a common) definition. Section 2.2 reviews existing permission managers, where Section 2.2.1 focus on current industry solutions, Section 2.2.2 on research proposals for better automation and personalization, and Section 2.2.3 reports on proposals to enhance the manager by adding obfuscation techniques, that is, methods where the quality of the data is degraded in order to retain a certain degree of privacy. Section 2.3 expands the previous sections by describing context-aware techniques used for both automation and privacy enhancement. Section 2.4 focuses on location privacy and presents obfuscation techniques used for this type of data. Finally, Section 2.5 summarizes this chapter and presents open issues that are tackled in this thesis.

2.1 Privacy

Although everyone has an idea of what is privacy, there is no universally accepted standard definition [Langheinrich, 2009]. Nertheless, privacy has been recognized as a right in the *Universal Declaration of Human Rights* [United Nation General Assembly, 1948] in 1948, however to a limited scope: the right to privacy at home, with family, and in correspondence. The difficulty in defining privacy comes as a consequence of the broadness of areas to which privacy applies [Yu, 2016; Acquisti et al., 2015]. The scope of privacy can be divided into four categories [Banisar et al., 1999]: *information*, which concerns the handling and collection of personal data; *bodily*, which relates to physical harms from invasive procedures; *communications*, which refers to any form of communication; *territorial*, which concerns the invasion of physical boundaries.

In the information scope, Westin defined privacy as "*the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others*" [Westin, 1968], or in other words, as the right to control the handling of one's information. Bertino et al. gave a similar definition, in terms of the control of the data, but explicitly incorporate the risks of privacy violation. These authors define privacy as "*the right of an individual to be secure from unauthorised disclosure of information about oneself that is contained in an electronic repository*" [Bertino et al., 2008]. Other definitions were proposed based on similar ideas of control and security [Langheinrich, 2009].

The aforementioned definitions of information privacy allow one to conclude that having control over the collection and handling of one's personal data is having

control over their privacy. This is fundamentally different from the definition of security [Langheinrich, 2009]. Security deals with authenticity, confidentiality and integrity of the data. However, it does not imply on how, when and by whom such data is accessed.

Some benefits of the information technologies are only possible through the collection and analysis of (sometimes sensitive) data. However, this may result in unwanted privacy violations. To protect from information leakage, privacy preservation methods have been developed to protect owner's exposure or to grant control over their data [Mendes and Vilela, 2017]. However, preserving privacy incurs in challenging usability and data utility issues that can harm the adoption and must therefore be addressed by researchers [Cranor et al., 2015].

Numerous privacy-preserving mechanisms have been proposed and these can be categorized based on the phase of the data lifecycle that they are applied [Mendes and Vilela, 2017]. Specifically, privacy can be retained at *collection time* if the mechanism acts before sending the data to the collector; at *publishing time*, when entities publish or share the data while maintaining some privacy of the data subjects; at the *output of data mining models* the models can be degraded in order to avoid revealing sensitive results; and when joining distributed datasets to extract global insights without revealing local information to other entities, that is, at a *data distribution*. This work focus on mechanisms that act at collection time, which empower users with control over their privacy even against the collecting entity.

Since privacy has no single standard definition, quantifying privacy is quite challenging and unfortunately, no single metric is enough as multiple parameters may be evaluated [Mendes and Vilela, 2017]. Privacy-preserving techniques may be evaluated in three fundamental aspects: privacy level metrics measure how secure is the data from a disclosure point of view, data quality metrics quantify the loss of information/utility and complexity metrics, which measure efficiency and scalability of the different techniques.

Privacy level and data quality metrics can be further categorized into two subsets [Bertino et al., 2008]: *data metrics* and *result metrics*. Data metrics evaluate the privacy level/data quality by appraising the transformed data that resulted from applying a privacy-preserving method (e.g. randomization or a privacy model). Result metrics make a similar evaluation, but the assessment is done to the results of the data mining (e.g. classifiers) that were developed with the transformed data. Presenting a survey on privacy metrics is out of the scope of this literature review, but more detail can be found in [Mendes and Vilela, 2017].

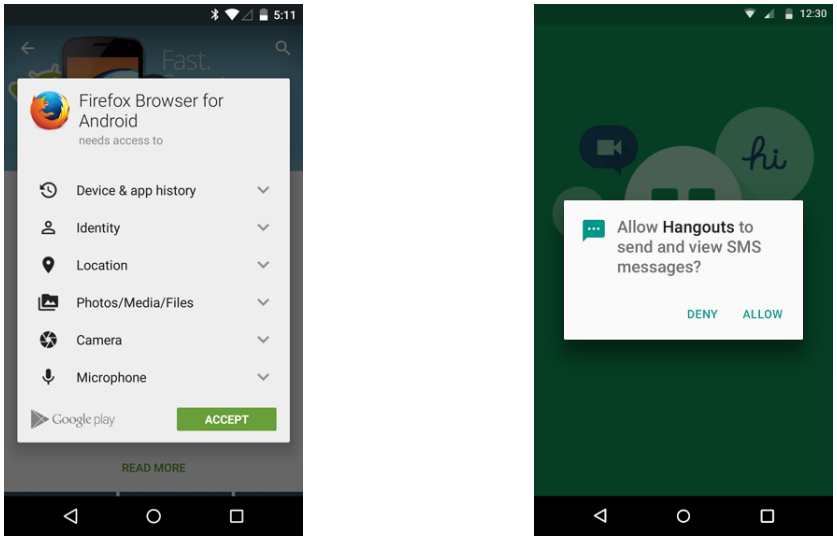
2.2 Privacy In Mobile Devices

Privacy in mobile devices is typically ensured by permission managers. Permission managers grant users the control to allow or deny application accesses to sensitive resources or information, such as the the camera or the contacts list.

In this section, we analyze permission managers of mobile devices, focusing on smartphones due to their market penetration, mobile data collection capabilities and relevance to this work. Within the smartphone industry, we mainly consider Android, due to being open source and for having over 80% of the market share [StatCounter, 2022]. Section 2.2.1 describes industry permission managers and the respective shortfalls. Section 2.2.2 then presents research proposals towards improving the status quo. Finally, Section 2.2.3 discusses obfuscation approaches towards providing a finer grained trade-off between privacy and utility that departs from the binary allow/deny that is currently employed in permission managers.

2.2.1 Current Permission Managers

Smartphones have implemented permission managers to give users control over the access to resources (data and sensors) and to warn about the associated risks. However, current industry privacy managers have been shown to be ineffective at protecting users' privacy.



(a) Install time permissions.

(b) Runtime permissions. Available since Android 6.0 (Marshmallow).

Figure 2.1.: Permission managers in Android.

In earlier versions of Android (previous to Android 6.0), applications would request access to all the needed resources at install time, as illustrated in Figure 2.1a. The user would either grant access to all permissions or cancel the installation. This ask-on-install approach has been shown to be ineffective as few users read the prompts and even fewer completely understand them [Felt et al., 2012; Kelley et al., 2012] leading users to make uninformed privacy and security choices.

To overcome the limitations of install time permissions, subsequent research proposed fine-grained runtime permissions [Nauman et al., 2010; Conti et al., 2010]. With this approach applications prompt the permission requests at runtime, allowing these prompts to be contextualized by the need for a certain functionality

and allowing users to accept or deny each of the permissions independently (fine-granularity). However, at the time, applications were not prepared to be denied access to permissions and consequently, application failure was frequent [Hornyack et al., 2011].

Fine-grained runtime permissions has since been adopted in the smartphones' industry and has been positively accepted by its users [Andriotis et al., 2018a; Reinfelder et al., 2018; Bonné et al., 2017]. Android first introduced runtime permissions with Android 6.0 (Marshmallow) in October 2015 [Rakowski, 2015], while iOS had it implemented earlier. However, the Android adoption was quite slow. In May 2018, almost 40% of current Android devices still ran earlier versions (c.f. Table 2.1), thus managing permissions at install time. At the time of writing (September 2021), there are still over 15% of devices using install-time permissions. Furthermore, even for devices with runtime permissions, if an Application (app) targets an earlier version of Android, permissions will still be prompt at install time. In this latter case, users can then go to device settings and disable access to certain permissions at the risk of breaking applications' execution [Hornyack et al., 2011]. However, it has been reported that users typically do no change their permission settings [Andriotis et al., 2018a]. Google has recently made a policy to their application store that requires that applications to be submitted or updated must target a recent Application Programming Interface (API) level [Google Developers, 2020], thus enforcing runtime permissions for recent apps.

Following recent research on the importance of app visibility in users' privacy decisions [Wijesekera et al., 2018], released in late 2018, Android 9 is able to restrict access to sensors, such as camera and microphone, when an app is idle or running in the background. Furthermore, in 2019 Android 10 has introduced a new permission which is required for apps to access location when the app is in the background, thus enabling the user to allow access to the location permission only while using the app. Finally, the latest Android version released in 2020, Android 11, implements: one-time permissions, which grant an apps the permission a single time; permissions auto-reset, in where granted permissions from apps are automatically set to the denied state when the app is unused for some time; and automatically blocked permissions, for permissions that are always denied by the user for specific apps.

While the enumerated improvements greatly enhance the privacy in mobile devices, the major drawback with current runtime fine-grained permissions and similarly to install time permissions is that, in general, after being accepted once, applications can access the resources at any time and for any purpose even without users noticing [Almuhimedi et al., 2015; Wijesekera et al., 2015]. This is often referred as a violation of the contextual integrity [Wijesekera et al., 2015; Tsai et al., 2017]. In fact, the comfort of users regarding apps requiring certain permissions is highly related to their expectancies [Lin et al., 2012], and therefore, users feel their personal space violated when confronted with apps' intrusive practices [Almuhimedi et al., 2015; Shklovski et al., 2014]. Consequently, industry permission managers found in Android (and iOS) still fail to convey the privacy risks that arise from allowing these permissions [Bonné et al., 2017;

Shen et al., 2021].

Towards improving privacy awareness, researchers have proposed using personal examples to better convey the permission risks [Harbach et al., 2014], crowdsourcing the feelings of uneasiness regarding apps' pratices [Lin et al., 2012], and designing better permission warnings [Shih et al., 2015] or indicators of resource usage [Feng et al., 2021]. These type of warnings can show information on how, how often and even for which purpose permissions are used in a non-intrusive approach [Schaub et al., 2015a; Gluck et al., 2016], thus leading users towards knowledgeable privacy decisions. However, the challenges here relate to the decision on which information is relevant and how to clearly present this information such that users understand and act upon these warnings [Schaub et al., 2015a; Shen et al., 2021]. Furthermore, it has been reported that these notices might "annoy" users when prompted at inconvenient times [Liu et al., 2016; Almuhimedi et al., 2015]. Solutions to this latter problem include configurable periodic nudges [Almuhimedi et al., 2015; Elbitar et al., 2021], and contextualized notices [Schaub et al., 2015a], where the user is presented with nudges which are relevant to the current context, such as upon occurrence of a specific data practice.

Existing permission tools such as Privacy Guard [Holly, 2015] and XPrivacy [Bokhorst, 2013] slightly improve default permission managers by giving finer grained permissions and by allowing an "always ask" option, similar to the one-time permissions from Android 11. Nevertheless, this latter option greatly increases the amount of user input required. In fact, applications can make hundreds of permission requests per day [Wijesekera et al., 2015], leading users towards the accept-once-use-everytime option. Indeed, constantly warning users provides limited success due to warning fatigue [Felt et al., 2012], a state where users become desensitize, and therefore solutions to set preferences automatically or with minimal user interaction are required.

Version	**Codename**	**API**	**Distribution**		**Permission Manager**
			2018-05-10	**2021-09-21**	
2.3.3 - 2.3.7	Gingerbread	10	0.3%	0.2%	Install Time
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%		Install Time
4.1 - 4.4	Jelly Bean	16-19	14.6%	5.7%	Install Time
5.0 - 5.1	Lollipop	21-22	22.4%	9.2%	Runtime
6.0	Marshmallow	23	25.5%	11.2%	Runtime
7.0 - 7.1	Nougat	24-25	31.1%	12.9%	Runtime
8.0 - 8.1	Oreo	26-27	5.7%	21.3%	Runtime
9.0	Pie	28	0%	31.3%	Runtime
10	Android 10	29	0%	8.2%	Runtime

Table 2.1.: Relative distribution of the number of devices running a given version of Android in two different timestamps: 10th of May 2018, as retrieved from [Android Developers, 2018] and 21st of September 2021 as retrieved from the Android Studio.

2.2.2 Automation And Personalization

In order to avoid warning fatigue [Felt et al., 2012] and improve usability, researchers have proposed solutions to define and/or enforce the permissions settings automatically. However, automating privacy choices can be challenging due to i) the always present trade-off between privacy and utility [Mendes and Vilela, 2017], which is often not obvious even for the users which are led to poor choices with respect to privacy [Acquisti et al., 2015]; ii) the subjectiveness of the choices which is influenced by personal preferences and cultural differences and beliefs [Acquisti et al., 2015]; iii) privacy's context-dependence [Acquisti et al., 2015]. This section focuses on the first two points, namely, automation and personalization, leaving privacy's context-dependence to Section 2.3.

An intuitive approach towards automated and personalized privacy protection is to allow for the definition of rules/policies that are enforced automatically. Such rules may be defined by either the end-users [Neisse et al., 2016] or by system administrators to enforce enterprise policies [Wang et al., 2015b]. The advantage of this approach is to allow for very specific and personalized rules as these are created by the users themselves. However, this requires a high amount of time and interaction to setup, and more importantly, expertise, which the average user does not have [Felt et al., 2012; Kelley et al., 2012; Shen et al., 2021].

The Android permission manager can be seen as an example of this rule creation approach, where the permissions are the rules. The drawback lies in the number of available permissions and the fact that each permission is set on a per-application and per-resource basis, and thus, customization/personalization comes at the expense of user interaction. Specifically, Android 12, the latest version at the time of writing, defines over 100 permissions, of which 34 require user intervention on a per app basis, while the majority of users have more than 50 installed apps [Andriotis et al., 2018a] that make hundreds of resource accesses daily [Almuhimedi et al., 2015; Wijesekera et al., 2015; Mendes et al., 2022a]. The framework proposed in [Neisse et al., 2016] improves this scenario by allowing users to define higher level policies that abstract users from lower level and more technical decisions. An example given by the authors is a user wanting to restrict an Android application from accessing location. Their framework then translates such rule to block all permissions that allow for the extraction of location, including less obvious permissions such as ACCESS_NETWORK_STATE, the Android permission which grants access to information about the mobile network, particularly the location as given by the cell towers. Enterprise policy enforcement frameworks mitigate the expertise and setup time drawback by having policies set by a system administrator [Wang et al., 2015b]. However, less personalization is achieved, which might not be critical for enterprise devices, and the effectiveness in protecting privacy and/or security relies on the precision of the rules.

To tackle the previous drawback, researchers proposed assigning privacy profiles, that is, a set of predefined rules that are defined according to the preferences of the users [Lin et al., 2014; Liu et al., 2014]. This line of work showed that while

people's mobile app privacy preferences are diverse, a small number of privacy profiles, which can be obtained through clustering techniques, can effectively capture the vast majority of users' preferences [Liu et al., 2014] and thus, minimize user interaction. A privacy manager implementing privacy profiles was proposed in [Liu et al., 2016], where the authors showed, using data from real users, that the generated privacy profiles can 1) effectively be assigned through a small number of questions, therefore reducing the amount of required input from users; 2) help users adopt the profile settings which better aligns with their personal preferences.

The previously mentioned approaches, i.e. policy frameworks and privacy profiles, present an essential weakness with respect to effective privacy protection, namely, the inability to evolve or adapt towards new situations. In fact, both approaches are static in the sense that after the setup, either through building the policies or adopting the privacy profile, no new settings are created without user interaction. Towards tackling this issue, researchers proposed an alternative approach based on privacy nudges [Almuhimedi et al., 2015], that is, informative warnings that are occasionally presented to the user in order to incentivize reviewing privacy settings. While still requiring interaction, nudges can present information on how, how often and even for which purpose permissions are used in a non-intrusive approach [Schaub et al., 2015a]. However, the challenges here relate to the decision on which information is relevant and how to clearly present this information such that users understand and act upon these warnings [Schaub et al., 2015a]. Furthermore, it has been reported that these notices might "annoy" users when prompted at inconvenient times [Liu et al., 2016; Almuhimedi et al., 2015]. Solutions to this latter problem include configurable nudges [Almuhimedi et al., 2015], to allow for user set notice periodicity, and contextualized notices [Schaub et al., 2015a], where the user is presented with nudges which are relevant for the current context, such as upon occurrence of a specific data practice.

Finally it should be pointed out that replicating user actions can also be implemented towards achieving automated and personalized privacy. This approach solves the problem of requiring expertise to set up, as a classifier can be trained through user actions, while still requiring some interaction for training. An illustrative example of this approach is found in [Olejnik et al., 2017], where the authors propose a privacy protection mechanism for smartphones that correctly automated 80% of user decisions (assessed in a field trial) using a Bayesian linear regression. The problem with this approach based on replicating user behavior is that users lack knowledge to make informed decisions [Felt et al., 2012; Kelley et al., 2012; Shen et al., 2021] and do not act accordingly to their preferences, and in fact, privacy choices have been shown to be malleable, that is, users often act against their preferences/beliefs in exchange for some benefit [Acquisti et al., 2015].

An under-looked aspect in the context of permission managers is the lack of fine grained control over privacy. Specifically, permission managers allow to either grant or deny a permission. This corresponds to either having maximum privacy and no utility, for the deny case, or maximum utility and zero privacy, for the

grant case. However, after being collected by a service provider, the data can be shared with third-parties, sold and even sometimes published publicly [Mendes and Vilela, 2017]. Therefore, simply allowing or denying access is a limited approach towards privacy protection. Techniques to preserve privacy at data collection such as obfuscation are required. The following section details this type of techniques.

2.2.3 Obfuscation

Several works on privacy in mobile devices focus on limiting the control to either allow or to deny exchange of information [Almuhimedi et al., 2015; Lin et al., 2014; Liu et al., 2016; Conti et al., 2010; Shebaro et al., 2015]. Nevertheless, when information is collected, data collectors have full access and control over such data and are often allowed to publish the data for either public access or for cloud distributed processing [Mendes and Vilela, 2017]. During this process, sensitive information can be sold, "leaked" to third-parties [Shklovski et al., 2014], or even exposed due to insufficiently anonymized datasets [Narayanan and Shmatikov, 2008; Tsoukaneri et al., 2016].

In order to empower users with control over the collected data, privacy protection mechanisms must act at data collection time, that is, before the data reaches the service providers. Obfuscation techniques, that is, techniques that purposely degrade the quality of the data, can be used within this context towards retaining a certain level of privacy. In Privacy-Preserving Data Mining (PPDM), obfuscation techniques are referred to as data sanitizing operations [Mendes and Vilela, 2017], and the most common techniques are:

- Generalization: replacement of a value for a more general one (parent). Numerical data may be specified by intervals (e.g. an age of 53 may be specified as an interval in the form of [50, 55]), whereas categorical attributes require the definition of a hierarchy. A good example of a hierarchy could be the generalization of the values "engineer" and "artist" from an occupation attribute to "professional". Another possibility would be to have the parent value of "student" to represent all types of student in the same occupation attribute;
- Suppression: removal of some attribute values to prevent information disclosure. This operation can also be performed column wise in a data-set (removes all values of an attribute) or row wise (removes an entry/record);
- Perturbation: replacement of the original data by synthetic values with identical statistical information. Perturbation includes additive and multiplicative noise, data swapping and synthetic data generation. In data swapping, sensitive attributes exchange between different entries of the dataset in order to prevent the linkage of records to identities, whereas in synthetic data generation, a statistical model is formed with the original data, and then synthetic values are obtained from the model.

Current permission managers allow users to either grant or deny access to the data, but fail to allow users to restrict the amount of information, or the quality

of the data that is collected. Furthermore, denying access to certain resources may render services inoperable. Obfuscation techniques such as perturbation, synthetic data generation and generalization can be applied at data collection to allow users to benefit from the services while restraining the quality of information that is collected, and thus, preserving a certain degree of privacy at the expense of the quality/utility of the data [Mendes and Vilela, 2017].

Since denying access to resources may often cause application failures, earlier work in obfuscation in permission managers focused in providing synthetic (false) data [Bokhorst, 2013; Hornyack et al., 2011; Beresford et al., 2011] instead of the real data. However, this approach still incurs in a great functionality loss and can still lead to failures [Hornyack et al., 2011]. A different approach was proposed more recently [Olejnik et al., 2017], where the authors trained a model to replicate the user in either granting, denying or obfuscating the data for permission requests. Their system was able to apply obfuscation to location, contacts, storage and camera data. While the authors suggest that better methods can be used for obfuscation, an open problem that arises in this context is the heterogeneity of types of data that can be accessed [Cunha et al., 2021], how the data is accessed by different types of applications, and how much data is "leaked" through obfuscation. In Section 2.4, a focus is given on obfuscation of location data, due to the relevance of this type of data in the context of mobile devices.

An essential advantage of the replication of user actions towards automated and personalized privacy is the trivial incorporation of new variables for the purpose of privacy enforcement. For instance, adding context-awareness would correspond to the addition of context features in a classifier and then training through user interaction as new contexts appear. By contrast, for policy frameworks this would require the creation of rules for uncountable new contexts/situations, whereas for privacy profiles would mean an increase in the number of possible profiles and interaction for profile assignment, as these would require to become context-aware. The following section focuses on this particular subject, namely, context-awareness, and reviews automated and context-aware privacy protection mechanisms.

2.3 Context-Awareness

In Ubiquitous Computing, context-awareness is a property of systems which improve and facilitate user interaction by taking into account the dynamic context, thus being useful for utility and possibly towards enhancing privacy. In this section, a review on context-aware systems is given as follows. Section 2.3.1 provides some background and definitions on the fundamental concepts of *Context* and *Context-Awareness* and Section 2.3.2 presents the state-of-the-art in context-aware systems for privacy in mobile devices.

2.3.1 Fundamental Concepts

The development of portable and always-connected devices has shifted computing systems from the static desktop to mobile devices that integrate seemingly with the environment. Mark Weiser predicted this paradigm and defined it as *ubiquitous computing* [Weiser, 1999]. Pervasive and Ubiquitous Computing has allowed for better interaction and immersion with the technology, leading to the development of emerging areas [Alegre et al., 2016] such as Intelligent Environments, Ambient Intelligence and even Internet of Things (IoT), in where the recognition of context plays a critical role. While somehow intuitive, the definition of *context* has seen no consensus [Alegre et al., 2016] and has consequently been subjective to specific concerns in different disciplinary areas [Sundmaeker et al., 2010].

An early definition of context dates back to 1994 [Schilit and Theimer, 1994] where the authors proposed a mobile application capable of reacting to changes in its environment. This definition was tightly related to location and objects within each location. Later in the same year, the authors highlighted three key aspects of context, namely: "where you are, who you are with, and what resources are nearby" [Schilit et al., 1994].

A broader and more recent definition was proposed in [Abowd et al., 1999] and has since became the most acknowledged one. These authors defined context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*". This definition allows for constraining what type of information is meaningful and thus define context in the scope of the application. Following this line of thought, the authors also define a *context-aware* system as a system that "*uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*".

The definitions of context and context-aware leave open the task of deciding which information is relevant to model the situation of the involved entities. Towards this end, the authors of [Abowd et al., 1999] referred to location, identity, time and activity as primary context types and every other context that could be inferred from the primary type as secondary context. A finer-grained approach was taken in [Perera et al., 2014] where primary context is used to refer to any data extracted directly from sensors, without performing any data fusion operations. By contrast, secondary context is any information that is extractable using the primary context. This includes not only data fusion operations but also data retrieval operations. In this sense, location data can be both primary context (e.g. the raw GPS data) and secondary context (e.g. semantic location data extracted from the GPS data).

Collecting and processing the data are essential steps of context-aware systems. In fact, the process of recognizing, handling and disseminate context information is known as context life cycle [Perera et al., 2014; Alegre et al., 2016]. Different context life cycles have been proposed, however [Perera et al., 2014] derived a

minimal form containing the four essential steps:

Context Acquisition In the first step, the primary context is sensed directly from the sensors. As multiple sensors are used and often distributed geographically, inaccuracies, missing data and jitter can pose a challenge [Bettini et al., 2010].

Context Modeling The sensed data from the first step is translated into readable modeling constructs. These models represent and relate the contexts with the different envolved entities, and must thus be simple, reusable, expandable and effective to allow to use the information at runtime [Bettini et al., 2010]. Many context modeling techniques have been proposed, being the most popular ones [Perera et al., 2014]: key-value, markup schemes, graphical, object based, logic based, and ontology based modeling. A detailed survey on these techniques is found in [Strang and Linnhoff-Popien, 2004].

Context Reasoning In the third phase, the modelled data is processed to derive higher level context (secondary context) from the sensor data. This includes pre-processing the primary context through data cleaning, sensor data fusion and finally context inference, where higher level context is derived.

Context Dissemination The final step concerns the method(s) for delivering the context to the consumer. Optimally the dissemination should occur in real-time.

The dissemination of context can have many forms. In fact, context-aware systems can be classified by the user interaction (or lack thereof). [Alegre et al., 2016] makes a distinction between two modalities: execution and configuration. The first one refers to systems that act automatically upon the arrival of specific situations, whereas the second one is related to the adjustment/settings of actions that a system will exhibit in the future. Both execution and configuration can further be classified as active, if the system can change its content autonomously, or passive, if user involvement is explicitly necessary in the actions taken by the system. Due to their relevance in this work, it is presented below a description of each of the four interaction types.

Active Execution In this modality, a system is able to respond autonomously to changes to the environment and to the system itself.

Passive Execution User action is required towards the response of the system to a change in the context. The system automatically presents options to the users, but requires user permission/choice to take action.

Active Configuration After deployment, the system is able to learn user preferences and autonomously evolve its rules for future behavior.

Passive Configuration Users manually configure their preferences and system behavior for future context changes.

Other taxonomies and important aspects of context and context-awareness have

been researched, such as the features and requirements of a context-aware system [Alegre et al., 2016] and context awareness management design principles [Perera et al., 2014]. However, doing a complete survey on context-awareness alone is outside the scope of this work. The next section focuses on context-aware privacy protection.

2.3.2 Context-Aware Privacy

As introduced in the beginning of Section 2.2.2, privacy has a strong dependence with context. In fact, [Acquisti et al., 2015] discusses on how individuals' feeling about privacy can range from complete apathy to extreme concern depending on the situation. This dependency has led researchers to explore context-awareness towards privacy [Schaub et al., 2015b].

Defining users' context in mobile devices can be challenging [Abowd et al., 1999] due to environment dynamism, missing information and inaccuracy of measurements [Schaub et al., 2015a], thus leading to simplified approaches. One of the earliest works on context-aware privacy [Conti et al., 2010] proposed user-set context and privacy policies, in which context was limited to a mix of time and location attributes, where location was defined statically by the user as a circular area (e.g. a work place). A more recent work [Shebaro et al., 2015] expanded this approach by allowing more precise locations (sub-areas), such as rooms. However, time and location are insufficient to precisely define context [Abowd et al., 1999], thus limiting the effectiveness of the privacy regulation. Moreover, user-defined context-policies may allow for better definition of context but are not effective for personal devices because it requires user intervention and expertise to setup [Shebaro et al., 2015], which the average user does not have [Felt et al., 2012; Kelley et al., 2012; Shen et al., 2021].

A better approach in defining context in mobile devices is taken in [Zavala et al., 2011], in where geo-location is translated into semantic locations using online services, which is then used to infer activities such as working, or walking. However, results show that as the number of activities grow, the accuracy of the inference drops greatly.

More recently, researchers have also been focusing in using the device context towards improving permission managers [Wijesekera et al., 2015, 2018; Das et al., 2016]. This type of context is defined by any piece of data that can define the current status of the phone, including which applications are running on the foreground and background, if there's an ongoing call, if the phone is locked, and others. Wijesekera et al. [Wijesekera et al., 2015] showed that permission decisions are closely related to what users are doing at the time of the prompt, showing that, for example, visibility of the application, that is, if the application is visible to the user, is of critical importance in the decision to either allow or deny a permission request.

The work in [Das et al., 2016] proposes a policy enforcement system, where permissions are either granted or denied depending on user defined policies that take into consideration both user and device context. As aforementioned, user

defined policies are not suitable for personal devices as they require user intervention and expertise to setup [Shebaro et al., 2015], which the average user does not have [Felt et al., 2012; Kelley et al., 2012; Shen et al., 2021].

A simple, yet generally more usable approach than defining policies, is to use Machine Learning (ML) models trained with contextual features. The work in [Wijesekera et al., 2018] is a field study to assess with real users and in real conditions a machine learning (ML) technique to automate permission decisions through contextual-awareness [Wijesekera et al., 2017] and their permission manager interface [Tsai et al., 2017]. This line of work is based on the notion of contextual integrity, where accesses to sensitive resources should be made as expected by the users. A previous work from these authors found that applications often violate contextual integrity as accesses to sensitive resources are often made when these apps are invisible (running in the background) to the user [Wijesekera et al., 2015]. In fact, the authors found visibility of the application to be one of the most if not the most important contextual feature for users' permission decisions [Wijesekera et al., 2015, 2017]. Thus, while their ML proposal used 20 device contextual features [Wijesekera et al., 2017], including both behavioral and runtime features, their follow up work focuses mostly on visibility of the requesting app [Wijesekera et al., 2018; Tsai et al., 2017]. Similarly, in [Olejnik et al., 2017], the authors propose a permission manager that learns from user behavior in order to predict users' permission decisions. This evolving system takes into consideration both user context, using time and a semantic location obtained through user input, and the device context as features for the predictor. While their approach is simplistic with respect to the user context, they were able to correctly predict 80% of users' decisions using a Bayesian linear regression model.

Following the notion of contextual integrity and building on the importance of visibility towards automation from [Wijesekera et al., 2018, 2015], the work in [Fu et al., 2019] proposes a new contextually-aware permission manager. The base idea is to extract contextual data from the UI presented to the users at the time of the request. The contextual data collected intends to answer the following three questions: 1) who initiated the request, which can give clues for the purpose of the request; 2) when did it happen, which indicates whether it resulted from user interaction; 3) what kind of environment, which encompasses the device context at the time of the permission access. Such data is crucial to model and detect violations of contextual integrity. To collect the data, the authors propose an hybrid approach composed of a static analysis to locate code corresponding to foreground permission requests, followed by dynamic rendering to extract the contextual data from the UI (layout and widget information).

Another type of information that has been attracting interest towards enhancing permission systems is the permission request purpose. Recall from Section 2.2.1 that run-time permissions have been proposed to allow for prompting the permission requests at the time the application requires access to the resource, thus allowing these requests to be contextualized by the need for access. However, after being accepted once applications can use the resource for any purpose and

in fact, it has been shown that applications often access resources for other purposes than their core functionality [Enck et al., 2014; Chitkara et al., 2017].

The authors of [Bonné et al., 2017] found that "one of the main reasons for granting or denying a permission request depends on users' expectation on whether or not an app should need a permission". This goes in line with a survey result from [Liu et al., 2016], where participants reported desire to be able to deny certain permissions for specific purposes but that doing so would break functionality. The authors of this latter study then conclude that purpose-centric controls as opposed to resource-centric could be used towards better privacy decision making.

With the current paradigm, there is no trivial way to obtain the purpose of permission requests. Techniques towards inferring the purpose do exist and can be categorized in three main approaches [Van Kleek et al., 2017]: (1) static analysis, where reverse engineering techniques are used towards analyzing the code to identify data collection activities, (2) Operating System (OS) instrumentation, where data flow is monitored through the system to identify potential misuses and unwanted transmissions of private information (e.g. using taint techniques [Enck et al., 2014]), and (3) network traffic monitoring, in where transmitted data is intercepted and analyzed.

The concept of privacy depends not only on the person and the context, but also on the type of data itself, and different types of data require distinct obfuscation techniques [Cunha et al., 2021]. This poses a challenge as mobile devices have rich and ever increasing sensory capabilities, thus throttling the adoption of these obfuscation techniques in permission managers. For instance, considering the case of location data, while obfuscation of a single location report can be straightforward, as the frequency of reports increases the geo-temporal correlation between points reduces the achievable privacy level (c.f. [Mendes and Vilela, 2018; Mendes et al., 2020]). In fact, location data is a prominent research area in mobile devices [Huang et al., 2018] specially considering the potential sensitivity of this information as not only it reveals whereabouts, but can also disclose identity, habits, health conditions and social connections [Primault et al., 2019; Krumm, 2009]. The following section delves into this subject through a detailed literature review on location privacy.

2.4 Location Privacy

The pervasiveness of smart devices and the always on and always connected paradigm has fostered applications that benefit from sensing the environment to provide contextualized services to its users. One category that has recently seen enormous growth in this space is the Location-Based Services (LBS) [Huang et al., 2018], in where mobile devices, such as smartphones, share their current position in order to obtain related information (e.g. finding the nearest restaurants). However, LBS providers may be incentivized to publish their data or share

with third-parties for financial or research purposes. However, poorly anonymized datasets [Sweeney, 2002], disclosure of aggregated mobility data [Xu et al., 2017] and even the publication of fully anonymized mobility datasets [Tsoukaneri et al., 2016] can be leveraged by informed adversaries leading to the deanonymization of individuals. This is due to the fact that human mobility traces are highly unique [De Montjoye et al., 2013; Song et al., 2014; Zang and Bolot, 2011], that PoIs act as quasi-identifiers [Bettini et al., 2005; Primault et al., 2014], that is, information that can be combined with data from other public sources to de-anonymize the owner, and that individual's traces are extremely predictable given past location history [Song et al., 2010].

The intrinsic nature of location data difficults effective privacy protection. In fact, location data can have different forms, depending on how the data is collected. Specifically, LBSs have been classified based on the frequency of location reports [Shokri et al., 2011; Liu et al., 2018a] as either sporadic, if the user makes use of the service irregularly and therefore the data corresponds to single points scattered in space and time, or continuous, if the user requires the service periodically and therefore full geo-temporal trajectories are generated. In turn, the amount of disclosed information shapes the possible attacks that can be carried by an adversary [Wernke et al., 2014]. Consequently, several Location Privacy-Preserving Mechanisms (LPPMs) have been proposed, fundamentally differing on the protection objective (identity and/or location), the type (sporadic or continuous) and amount (single user or multiple users) of available data [Liu et al., 2018a; Primault et al., 2019].

Privacy protection has originally and predominantly been employed by the service providers after the data has been collected. However, this scenario requires trust from the users that their data is handled properly, as after the data is collected, the user has no (or limited) control over it [Mendes and Vilela, 2017]. More recently, mechanisms that protect privacy at data collection, that is, in an online fashion before the data is sent to the provider, have been raising research interest due to empowering users with control over their privacy. This is specially true for LPPMs, where a great portion of the recent studies are mechanisms for online privacy protection [Primault et al., 2019; Liu et al., 2018a].

A typical framework to evaluate an LPPM consists of a (or multiple) user(s) reporting locations, an LPPM, an adversary, which is characterized by its attacks and background knowledge, and a (or multiple) metric(s) [Shokri et al., 2011]. To understand the state of the art of LPPMs, a precise definition of employed notation is required, as follows and summarized in Table 2.2. This notation is valuable for describing relevant LPPMs (Section 2.4.2) and respective attacks (Section 2.4.3) to evaluate LPPM efficacy.

2.4.1 Formalizing Location Privacy

As in previous relevant works [Chatzikokolakis et al., 2017; Mendes and Vilela, 2018; Shokri et al., 2011; Oya et al., 2019; Shokri, 2015], we shall consider a user of an Location-Based Service (LBS) which reports his location to the LBS provider to obtain information. We consider as adversary any entity with access

Table 2.2.: Summary of notation

Symbol	Description
x^r	r^{th} location from $\mathcal{X}$, with $r \in \{1, \ldots, \lvert\mathcal{X}\rvert\}$
x_i	Exact user location at timestamp i.
z_i	Obfuscated location at timestamp i.
$\hat{x}_i$	Adversary's estimated location at timestamp i.
t_i	Time at timestamp i.
$\mathbf{x}$, $\mathbf{z}$, $\hat{\mathbf{x}}$	Vector of all real, obfuscated or estimated locations, respectively.
$\mathbf{x}_i$, $\mathbf{z}_i$, $\hat{\mathbf{x}}_i$	Vector of real, obfuscated or estimated locations up to timestamp i.
$\mathcal{X}$, $\mathcal{Z}$, $\hat{\mathcal{X}}$	Set of all possible real/obfuscated/estimated locations.
Δ_t	Minimum interval between consecutive reports.
f, $p(z_i \lvert x_i)$	Location Privacy-Preserving Mechanism (LPPM).
ϵ	Geo-indistinguishability privacy parameter.
h, $p(\hat{x}_i \lvert z_i)$	Adversary's attack.
$P_{AE}(f, h, \mathbf{x}, \mathbf{z})$	Mean adversary error of $\hat{\mathbf{x}}$ given $\mathbf{z}$ and h.
$Q(f, \mathbf{x}, \mathbf{z})$	Mean quality loss given the LPPM f and locations $\mathbf{x}$.
$d(\cdot)$	Euclidean distance metric.
$g(\cdot)$	Great-circle distance.
o_i	Noisy GPS reading at timestamp i.
$s_{i,k}$	k^{th} candidate location for o_i at timestamp i.
$p(o_i \lvert s_{i,k})$	Map-matching emission probability.
$p(s_{i,k} \lvert s_{i-1,j})$	Map-matching transition probability
σ	Standard deviation of the (GPS) measurement error.
λ_y	Parameter for the exponential of the measure of circuitousness.
λ_z	Parameter for the exponential of the measure of temporal plausibility.

to the location reports attempting to infer private information [Gambs et al., 2010; Krumm, 2007], including the LBS provider or any passive eavesdropper. Furthermore, the adversary can have arbitrary background information (prior) and computational power. In order to protect his privacy, the user uses an LPPM to report an obfuscated version of his exact location, consequently trading the quality of the LBS response for privacy.

Formally, let $x_i \in \mathcal{X}$ denote the exact user's location at the report with timestamp $i \in \{1, 2, \ldots, T\}$ and $z_i \in \mathcal{Z}$ the reported obfuscated location at the same i computed using the LPPM f. For convenience, we use t_i to express the real time of timestamp i. The adversary has access to z_i and it is assumed to know f and possibly have some a priori knowledge and thus computes $\hat{x}_i \in \hat{\mathcal{X}}$, an estimation of x_i at each timestamp i, using an attack h. We shall denote $\mathbf{x}_i$ and $\mathbf{z}_i$ the vectors of real and obfuscated locations up to timestamp i, respectively, that is, $\mathbf{x}_i = \{x_1, \ldots, x_i\}$ and $\mathbf{z}_i = \{z_1, \ldots, z_i\}$. Unless otherwise stated, let $\mathcal{X}$, $\mathcal{Z}$ and $\hat{\mathcal{X}}$ to be in $\mathbb{R}^2$. In the context of frequency of reports, we define Δ_t in seconds as the minimum interval between any two consecutive location reports. Formally, $\Delta_t = \text{argmin}_i\ (t_{i+1} - t_i)$.

Generically [Oya et al., 2019], online user-centric obfuscation mechanisms can

be described as a probability distribution in the form of equation (2.1).

$$p(z_i|\mathbf{z}_{i-1}, \mathbf{x}_i) \tag{2.1}$$

Intuitively, an LPPM maps the real location $x_i \in \mathcal{X}$ with the knowledge of past locations $\mathbf{x}_{i-1}$ and past reports $\mathbf{z}_{i-1}$ to a new report $z_i \in \mathcal{Z}$. In the context of sporadic location privacy, existing LPPMs consider location reports to be independent, and consequently, each obfuscated report z_i is made only with respect to the exact position x_i at the same timestamp i. Therefore, equation (2.1) is reduced to the form:

$$p(z_i|\mathbf{z}_{i-1}, \mathbf{x}_i) = p(z_i|x_i) \tag{2.2}$$

LPPMs of this form are referred to as memoryless [Oya et al., 2019].

In the context of localization attacks, the primary privacy metric is the correctness of an adversary measured by the expected estimation error [Shokri et al., 2012; Oya et al., 2017a] and modeled through a distance metric between the exact locations and the adversary's estimations. Given an LPPM f, an attack h and observations $\mathbf{z}$, the expected adversary estimation error (AE) is defined by the following equation:

$$P_{AE}(f, h, \mathbf{x}, \mathbf{z}) = E\{d(x_i, \hat{x}_i)\} \tag{2.3}$$

where the expected value is taken over x_i and $\hat{x}_i$, and $d(\cdot)$ is a distance metric which is typically the Euclidean distance [Shokri et al., 2012].

From the user perspective, the LPPM f introduces a quality loss due to reporting the obfuscated location instead of the exact position [Shokri et al., 2011; Oya et al., 2017a]. The average quality loss is therefore given by:

$$Q(f, \mathbf{x}, \mathbf{z}) = E\{d(x_i, z_i)\} \tag{2.4}$$

The following sections provides an overview on existing LPPMs and location attacks and detail the ones relevant for this thesis.

2.4.2 Location Privacy-Preserving Mechanisms (LPPMs)

To empower users with control over their privacy against untrustworthy providers, it is required to employ mechanisms that protect privacy at data collection, that is, in an online fashion before the data is sent to the provider. This is specially true for LPPMs, where a great portion of the recent studies are mechanisms for online privacy protection [Primault et al., 2019]. However, due to the aforementioned characteristics of location data, namely, its uniqueness, identifiability and predictability, properly preserving privacy of individuals at collection time while allowing for the use of this type of services is challenging.

Location privacy can be achieved through anonymity, to protect the ownership of the data, application-specific queries, which can protect the identity or location data, or through data obfuscation [Liu et al., 2018a], which protects the spatial and temporal data. Anonymization requires either the use of a trusted third-

party server (anonymizer) [Beresford and Stajano, 2003; Gruteser and Grunwald, 2003], which is arguably only a shift from trusting an LBS provider [Shokri et al., 2014], or a Peer-to-Peer (P2P) collaboration between devices in the vicinity [Chow et al., 2011], where this latter approach has other challenges such as colluding malicious peers and non-ubiquitous connectivity [Hoh et al., 2010]. For application-specific queries, there are cryptography-based approaches [Ghinita et al., 2008], which can incur in high computational costs, making it unfeasible for the majority of applications, and caching approaches [Meyerowitz and Roy Choudhury, 2009], which for effectiveness require the use of an anonymizer or P2P communications. Finally, obfuscation approaches degrade the quality of the released information as to reduce the risk of disclosure [Krumm, 2009; Liu et al., 2018a] and can therefore be generally applied. It should be noted that these types of LPPM are not mutually exclusive and can therefore be combined to protect both the identity and geo-spatial data [Liu et al., 2018a]. In this work we focus on online obfuscation due to their general applicability and to the fact that in the context of smartphones, LBSs typically require an account, rendering anonymization approaches unusable.

Differential privacy has become the standard for privacy preservation, giving rigorous and provable guarantees [Dwork, 2008]. This notion was first proposed in the context of statistical databases towards protecting aggregated statistics. Specifically, it bounds the information gain of an adversary regarding whether a single user (record) is present or absent from that database, thus preserving individual privacy. This concept has been generalized to location data under the concept of geo-indistinguishability [Andrés et al., 2013], a notion to design online LPPMs with the privacy guarantees inherited from differential privacy. This thesis follows the lines of geo-indistinguishability and therefore the following sections present a literature review on these type of mechanisms.

2.4.2.1 Geo-Indistinguishability And The Planar Laplace

Geo-indistinguishability [Andrés et al., 2013] has been proposed as a formal notion based on differential privacy [Dwork, 2008] to design user-centric LPPMs, that is, LPPMs that obfuscate the user data independently of other users [Shokri, 2015]. Geo-Indistinguishability (Geo-Ind) guarantees that the user location is indistinguishable to any other location *close to the user* based on the observed (obfuscated) report independently of an attacker's background information. Or in other words, the obfuscated report could have been generated with (almost) the same probability from any location around the exact user location.

Geo-Ind is formally defined as follows [Mendes and Vilela, 2018]. Consider a location privacy mechanism as a probabilistic function $K(\cdot)$ that assigns to each location $x \in \mathcal{X}$ a probability distribution on $\mathcal{Z}$, the set of all possible obfuscated locations, where $\mathcal{X}$ and $\mathcal{Z}$ are assumed to be discrete to simplify notation. A mechanism K satisfies ϵ-Geo-Indistinguishability iff:

$$d_{\mathcal{P}}\left(K(x), K(x')\right) \leq \epsilon d_{\mathrm{x}}(x, x') \quad \forall x, x' \in \mathcal{X} \tag{2.5}$$

where $d_{\mathrm{x}}(\cdot)$ is any distance function and $d_{\mathcal{P}}(\cdot)$ is the multiplicative distance

between two distributions, defined as $d_{\mathcal{P}}(\sigma_1, \sigma_2) = \sup_{S \in \mathcal{S}} \left|\log \frac{\sigma_1(S)}{\sigma_2(S)}\right|$, where σ_1 and σ_2 are two distributions on some set S, with the convention that $\mathcal{L} = \left|\log \frac{\sigma_1(S)}{\sigma_2(S)}\right| = 0$ if $\sigma_1(S) = \sigma_2(S) = 0$ and $\mathcal{L} = \infty$ if one of the two is 0.

Intuitively, equation (2.5) states that the probability of reporting location z while standing in location x is similar to that of standing in any location x'. In fact, both probabilities differ at most by the distance between x and x' factored by a small constant ϵ, where ϵ may be used to tune Geo-Indistinguishability. Commonly, and as specified in the seminal work [Andrés et al., 2013], this constant is set to $\epsilon = l/r$, such that for any x, x' s.t. $d_{\mathrm{x}}(x, x') \leq r$, $d_{\mathcal{P}}\left(K(x), K(x')\right) \leq l$, where d_{x} is an arbitrary metric and l is a user defined parameter termed *privacy loss*. This enforces that any x' within distance r of x discloses at most l information. Consequently, the true location x is better concealed for closer x' locations, while allowing higher dissimilarity for distant locations, thus preserving some degree of utility.

The Planar Laplace (PL) mechanism was the first proposed mechanism to achieve the notion of Geo-Indistinguishability [Andrés et al., 2013] and consists of adding 2-dimensional Laplacian noise centered at the exact user location x and with PDF [Andrés et al., 2013]:

$$p(z|x) = \frac{\epsilon^2}{2\pi} e^{-\epsilon d_{\mathrm{x}}(x,z)} \tag{2.6}$$

Obtaining z from x using equation (2.6) can be efficiently done by adding a randomly drawn vector expressed as a radius r and angle Θ. Θ is uniformly chosen from $[0, 2\pi)$ and r is computed by drawing p uniformly from $[0, 1)$ and feeding it to the inverse planar Laplacian cumulative distribution function defined as $C^{-1}(p) = -\frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{e}\right) + 1\right)$, where W_{-1} is the negative branch of the Lambert W function. Finally, $z = x + \langle r\cos\Theta, r\sin\Theta\rangle$.

While Geo-Ind is promising due to giving strong and quantifiable privacy guarantees, problems arise from continuous disclosure of information, as each report is treated independently of past locations. Namely, such approaches disregard the potential threat that arises from exploring the correlation between reports, which in turn can be used by an adversary to track users over time and even predict future locations [Krumm, 2009; Liu et al., 2018a; Xiao and Xiong, 2015]. Furthermore, the authors show that the privacy degradation is linear with the number of queries [Andrés et al., 2013], that is, a user performing n queries through a ϵ-geo-indistinguishable mechanism enjoys $n\epsilon$-Geo-Indistinguishability, which is only acceptable for a small n.

A later work from the original authors of Geo-Ind proposed a natural extension to the case of location traces [Chatzikokolakis et al., 2014]. The base idea is to use the metric $d_\infty(\mathbf{x}, \mathbf{x}')$, where $\mathbf{x}$ and $\mathbf{x}'$ are two traces (instead of locations), in equation (2.5). However, such approach is not ideal for an online LPPM at data collection as it considers a report to contain a full trajectory as opposed to a single point, that is, the secret is the full trajectory.

Depending on the LBS, location data can be reported either continuously or

rather sporadically [Shokri et al., 2011; Shokri et al., 2011]. This frequency of reports directly impacts the temporal correlation between subsequent reports which in turn can be used by an adversary to track users over time and even predict future locations [Liu et al., 2018a; Krumm, 2009; Xiao and Xiong, 2015]. While geo-indistinguishability bounds the amount of disclosure, it considers reports to be independent between each other. In fact, in the context of sporadic release of data this consideration has been assumed when designing LPPMs [Shokri et al., 2011; Oya et al., 2019]. Tackling this drawback, two recent adaptations of geo-indistinguishability for online privacy protection were proposed: Clustering Geo-Indistinguishability [Cunha et al., 2019] and Adaptive Geo-Indistinguishability [Al-Dhubhani and Cazalas, 2018]. The base of Clustering Geo-Indistinguishability is to reduce the amount of privacy loss by reusing previous obfuscation reports as a function of the past and current user locations, such that if the user at the current timestamp is still close to its previous location, then the same obfuscated location is reported instead of generating a newer one. By reporting a previously generated obfuscated location, the privacy budget is preserved and, therefore, the privacy no longer degrades linear, but as a function of the mobility of the user. A different approach is taken in the Adaptive Geo-Indistinguishability, where a dynamic adjustment of the privacy budget is made in order to increase privacy or utility according to the correlation between past and current location. The following sections detail each of these approaches.

2.4.2.2 Clustering Geo-Indistinguishability

The composability property of differential privacy states that the privacy loss is linear with the number of reports. Specifically, reporting n locations under Geo-Indistinguishability results in a privacy loss of $n.\epsilon$ [Andrés et al., 2013; Chatzikokolakis et al., 2014]. Therefore, Geo-Indistinguishability is only effective for sporadic reports [Mendes et al., 2020]. Under continuous reports however, the privacy loss becomes prohibitive and correlations between subsequent reports can be used to improve the efficiency of potential attacks [Liu et al., 2018a; Xiao and Xiong, 2015]. Clustering Geo-Indistinguishability [Cunha et al., 2019] tackles this problem by reducing the number of reports by taking into consideration the traveled distance. Namely, if the distance between the current position and the previous position is smaller than some radius, then instead of reporting a new obfuscation, the previous obfuscated report is used instead.

Formally, let x_c and r be the center and radius of an area, denoted cluster, x_i the user position at timestamp i and z_i, the obfuscated report at timestamp i. Then:

$$z_i = \begin{cases} z_{i-1} & \text{if } d_2(x_c, x_i) \leq r \\ planarLaplace(x_i, \epsilon) & \text{otherwise} \end{cases} \tag{2.7}$$

Essentially, if the distance between the center of the cluster x_c and the current user position x_i is higher than a radius r then a new obfuscation z_i is generated using the Planar Laplace as defined in equation (2.6). When this happens, a

new cluster is created by setting the center of the cluster to the current user position, that is, $x_c = x_i$.

In Clustering Geo-Indistinguishability, the privacy and utility level can be tunned by the radius r. Increasing the radius results in an increased privacy at the expense of the utility, and vice-versa for a decrease of the radius. To avoid increasing the number of required parameters, the radius r can be set following the approach from Geo-Indistinguishability $\epsilon = l/r \leftrightarrow r = l \cdot \epsilon$. Therefore, only two parameters are required, the privacy loss l and privacy budget ϵ.

The downside of Clustering Geo-Indistinguishability is the lack of adaptability to the context, such as varying frequency of reports. After setting the radius r, by tuning the privacy loss l and the budget ϵ, the obfuscated reports are generated with the same Laplacian PDF from equation (2.6). This can result in cases where an inefficient privacy-utility trade-off is achieved. The following section describes a more dynamic geo-indistinguishable approach where the privacy and utility are automatically adjusted as a function of the correlation between reports.

2.4.2.3 Adaptive Geo-Indistinguishability

Geo-Indistinguishability is only effective for the sporadic use of an LBS as the privacy degrades linearly with the number of queries (c.f. [Andrés et al., 2013; Chatzikokolakis et al., 2014]) and due to the fact that continuous location reports are highly correlated [Chatzikokolakis et al., 2014; Wang et al., 2015c]. Towards tackling this disadvantage, the Adaptive Geo-Indistinguishability was proposed [Al-Dhubhani and Cazalas, 2018], in where the privacy and utility is adjusted depending on the correlation between past and current locations at each report. Specifically, this notion dynamically increases privacy if the correlation is high, signaling that the current location is highly predictable, or increases utility if the correlation is low, signaling that sufficient privacy is already achieved.

The Adaptive Geo-Indistinguishability uses the Planar Laplace mechanism described in equation (2.6) as baseline, while providing the aforementioned adaptability by dynamically adjusting the privacy budget ϵ according to the correlation. For measuring the correlation, a linear regression is used to produce an estimation $\hat{x}_i$ of the real user location x_i at each timestamp i using past locations up to i. Depending on the Euclidean distance between the estimation and real location $d(x_i, \hat{x}_i)$, the mechanism increases either privacy or utility by adjusting the privacy budget as follows:

$$\epsilon_i = \begin{cases} \alpha \cdot \epsilon, & \text{for } d(x_i, \hat{x}_i) < \Delta_1 \\ \epsilon, & \text{for } \Delta_1 \leq d(x_i, \hat{x}_i) < \Delta_2 \\ \beta \cdot \epsilon, & \text{for } d(x_i, \hat{x}_i) \geq \Delta_2 \end{cases} \tag{2.8}$$

where Δ_1 and Δ_2 are thresholds and α and β two constants with the following constraints: $\Delta_2 > \Delta_1$, $0 < \alpha < 1$ and $\beta > 1$. Fundamentally, if the the distance between the estimation and the user location is lower than a threshold Δ_1, then the correlation between past and current locations is high. Therefore,

the mechanism decreases the privacy budget ϵ_i to increase privacy. If instead the correlation is low, signaled by a distance between the real and estimated locations higher than a threshold Δ_2, then the mechanism adjust for increasing utility.

As defined in equation (2.8) and in contrast with the Planar Laplace and Clustering Geo-Indistinguishability, Adaptive Geo-Indistinguishability provides a dynamic adjustment of the privacy and utility by taking into account previous reports. Note however, that this adjustment comes at the expense of usability. Namely, in addition to setting ϵ, the user must also define four extra parameters: Δ_1, Δ_2, α and β. This is a crucial drawback on the usability of the LPPM, as a misconfiguration may lead to an ineffective privacy/utility adjustment or even no effective privacy as we demonstrate in Chapter 4.

2.4.3 Attacks on Location Data

Location privacy attacks are diverse with respect to both the objective and the applied methods [Wernke et al., 2014; Liu et al., 2018a]. In this thesis we focused on the objective of locating the user at each timestamp. This objective is general in the sense that it allows for the reconstruction of the true mobility of the user and consequently, for posterior inference attacks, that is, attacks which produce additional knowledge from the geolocation data [Gambs et al., 2010] (e.g. extraction of user's PoI). However, different LBSs require different frequency of location reports. For example, finding the nearest PoI (e.g. restaurant) applications only require the location at the time of the query, so the release is sporadic, whereas for navigation services the reports are continuous. In this context, one must consider both tracking techniques, which consist in following a user over time and space, and localization techniques, which have as objective to localize the user at certain points in time [Shokri et al., 2011].

For localization attacks, we focus on the state-of-the-art by considering the optimal attack given a mobility profile [Shokri et al., 2012] and an heuristic which learns the mobility profile as locations are shared [Oya et al., 2019]. Section 2.4.3.1 and 2.4.3.2 detail these attacks, respectively.

In tracking attacks, one can consider regression analysis, Kalman filtering, particle filters and map-matching [Wernke et al., 2014; Krumm, 2009]. In a previous work [Mendes and Vilela, 2018], regression analysis has been used to produce simple estimators (such as linear and polynomial) as a tracking attack. However, results showed that such solution generates a non-negligible amount of outliers due to time-gaps in reports, which occur due to failures in the GPS or in communications. Kalman filters have been used effectively in navigation to reduce uncertainties arising from the noisy measurements. Particle filters can be used for the same purpose incurring in higher computational complexity. However, these two techniques are oblivious of the underlying map and consequently generate positions that are not physically possible (e.g. inside a building if the user is driving). A knowledgeable adversary can make use of the map to reduce this kind of uncertainties and thus locate the user with higher precision [Wernke et al., 2014]. This process is known as map-matching and it is typically used to

locate vehicles on road-networks [Kubicka et al., 2018].

In Section 2.4.3.3 we detail a map-matching technique that is robust to noise and to varying frequencies of location reports, that was used in this work for the purpose of tracking users in Chapter 5. Even though map-matching has been used as an attack, for instance, against area obfuscation [Wernke et al., 2014], to the best of our knowledge, we are the first to consider road-network map-matching as a tracking attack. We also note that this choice was further supported by the fact that hidden Markov chains, which are used in map-matching, have been shown effective in modeling the temporal correlations of location traces [Xiao and Xiong, 2015; Murakami, 2017]. Finally, it should be noted that the considered optimal localization attacks can also be used for trajectories [Shokri et al., 2017]. However, these attacks require the discretization of the space (and possibly time), which becomes computationally infeasible for finer resolutions.

2.4.3.1 Optimal Localization Attack

As formalized, the adversary observes $\mathbf{z}$, knows the used LPPM f and has some priory knowledge in the form $p(\mathbf{x})$. Consequently, it computes $\hat{\mathbf{x}}$ by means of an attack h. We focus on the case that the adversary estimates x_i using only observed reports up to i, that is, $\mathbf{z}_i$. This case can be generalized to the estimation of x_i using z_k with $i \leq k$ [Oya et al., 2019], however this is rarely the case in tracking approaches. Following [Oya et al., 2019; Shokri et al., 2012], the optimal localization attack minimizes the estimation error defined by equation (2.3). Formally:

$$\hat{x}_i = \operatorname*{argmin}_{\hat{x}_i} \sum_{x_i \in \mathcal{X}} p(x_i|\mathbf{z}_i) \cdot d_P(x_i, \hat{x}_i) \tag{2.9}$$

where $p(x_i|\mathbf{z}_i)$ is the posterior probability of x_i given all reports up to i:

$$p(x_i|\mathbf{z}_i) = \frac{p(\mathbf{z}_i|x_i) \cdot p(x_i)}{p(\mathbf{z}_i)} = \frac{\prod_{l=1}^{i} p(z_l|\mathbf{z}_{l-1}, x_i) \cdot p(x_i)}{p(\mathbf{z}_i)} \tag{2.10}$$

Note that since z_l is conditionally independent of x_i for $l \neq i$ and since we are considering only memoryless LPPMs, we have:

$$\begin{cases} p(z_l|\mathbf{z}_{l-1}, x_i) = p(z_l|\mathbf{z}_{l-1}) & \text{if } l \neq i \\ p(z_l|\mathbf{z}_{l-1}, x_i) = p(z_i|\mathbf{z}_{i-1}, x_i) = p(z_i|x_i) & \text{if } l = i \end{cases}$$

Furthermore, since equation (2.9) is a minimization, we can ignore the denominator and thus reach the attackers objective function as:

$$\hat{x}_i = \operatorname*{argmin}_{\hat{x}_i} \sum_{x_i \in \mathcal{X}} p(z_i|x_i) \cdot p(x_i) \cdot d_P(x_i, \hat{x}_i) \tag{2.11}$$

The final consideration of an attacker is the characterization of $p(\mathbf{x})$. Tradi-

tionally [Shokri et al., 2011; Shokri et al., 2012], $p(\mathbf{x})$ is described by a mobility profile π which is a probabilistic representation of the user mobility, where each user location is considered an i.i.d. sample of π. Formally, let $\pi(x)$ denote the probability that the user is at $x \in \mathcal{X}$ given the mobility profile π, then $p(\mathbf{x}) = \prod_i \pi(x_i)$. Therefore, and in practice [Chatzikokolakis et al., 2017], a realistic adversary would use a mobility profile built with training data, π^{train}. An omniscient adversary is sometimes considered as one who has access to the test data and thus, builds the mobility profile from this data, π^{test}. This latter adversarial consideration gives a lower bound for the expected privacy. We refer to the optimal attack using π^{train} as **optHW** and using π^{test} as **omniHW**.

Recently, Oya et al. [Oya et al., 2019] observed that building the mobility model a priori with the training data might fail to capture the true mobility of the users. The closer the model is to the real mobility, the better performant is the attack[1]. Consequently, the authors propose a new approach towards building mobility profiles which considers the true mobility to be unknown, and therefore learned based on real user behavior in an a posteriori fashion. An attack using this approach was proposed in [Oya et al., 2019] and results showed to have better performance than the optimal attack using the a priori model. The attack is denominated Profile-Estimation Based Attack (PEBA) and described in the following section.

2.4.3.2 Profile-Estimation Based Attack (PEBA)

PEBA [Oya et al., 2019] is based on the idea that the real mobility profile is unknown and consequently has to be learned/adapted after each query. Formally, let $p(\pi)$ be the probability of being assigned a profile $\pi \in \mathcal{F}_\pi$, then the real locations are i.i.d samples of the distribution given by π, such that:

$$p(\mathbf{x}) = \sum_{\pi \in \mathcal{F}_\pi} p(\pi) p(\mathbf{x}|\pi) = \sum_{\pi \in \mathcal{F}_\pi} p(\pi) \prod_i \pi(x_i) \tag{2.12}$$

This consideration creates a dependency between exact locations due to the fact that a previous location gives information on the unknown profile π which in turn affects the probability of the following locations. Therefore, the real locations and obfuscated locations will also be dependent as a location at x_j affects distribution of a location at x_i with $i > j$ which in turn affects z_i. Consequently, it becomes mathematically intractable to find the optimal attack considering equation (2.12) [Oya et al., 2019]. Thus, PEBA is a sub-optimal attack.

Following [Oya et al., 2019], PEBA is decomposed in two sequential steps: 1) estimation of the mobility profile using the observed obfuscated reports $\mathbf{z}_i$ up to the current timestamp, i. In the original proposal the Maximum Likelihood (ML) estimator is used and thus, this mobility profile is denoted by $\hat{\pi}_i^{ML}$; 2) estimate the real location $\hat{x}_i$ using $\mathbf{z}_i$ and assuming that x_i follows the estimated mobility profile $\hat{\pi}_i^{ML}$. We skip the foundational details of the method and focus

[1]Note that the mobility profile might not only be used by an adversary in the attack but also by the user in the LPPM [Shokri et al., 2017].

on the implementation steps. The interested reader should refer to [Oya et al., 2019].

The procedure of the steps is as follows. From the training data an initial average mobility profile π^{avg} is built from all the users. Then, this initial profile is used to estimate $\hat{\pi}_i^{ML}$, through an iterative Expectation-Maximization method:

$$\pi^{r,t+1} = \frac{1}{i}\sum_{l=1}^{i} p(x_l^r|\mathbf{z}_l, \pi^t) == \frac{1}{i}\sum_{l=1}^{i} \frac{\pi^{r,t} \cdot f(z_l|\mathbf{z}_{l-1}, x_l^r)}{\sum_{k=1}^{|\mathcal{X}|} \pi^{k,t} \cdot f(z_l|\mathbf{z}_{l-1}, x_l^k)}$$

where t is an iteration counter and $\pi^r \equiv p(x = x^r)$ with $x^r \in \mathcal{X}$ and $r \in \{1, \ldots, |\mathcal{X}|\}$ denotes the probability mass function defined by π. Furthermore, $\pi^0 = \pi^{avg}$. This step is repeated while the change from π^t to π^{t+1} is significant. Then, a normalization of the profile is made following equation (2.13). This latter equation holds that for the initial queries, the initial mobility profile π^{avg} is dominant, and then fading out as the number of queries increase in favor of the ML estimator.

$$\hat{\pi}_i = \frac{1}{i^{0.5}} \cdot \pi^{avg} + \left(1 - \frac{1}{i^{0.5}}\right) \cdot \hat{\pi}_i^{ML} \tag{2.13}$$

The posterior is then computed as:

$$p(x_i|\mathbf{z}_i, \hat{\pi}_i) = p(\mathbf{z}_i|x_i, \hat{\pi}_i) \cdot \hat{\pi}_i(x_i)/p(\mathbf{z}_i) = \prod_{l=1}^{i} p(z_l|\mathbf{z}_{l-1}, x_i, \hat{\pi}_i)\hat{\pi}_i(x_i)/p(\mathbf{z}_i) \tag{2.14}$$

And finally, using the posterior, the PEBA estimation of the exact location of the user is calculated as:

$$\hat{x}_i = \underset{\hat{x}_i}{\operatorname{argmin}} \sum_{x_i \in \mathcal{X}} p(x_i|\mathbf{z}_i, \hat{\pi}_i) \cdot d_P(x_i, \hat{x}_i) \tag{2.15}$$

2.4.3.3 Map-Matching

The previous sections described attacks against sporadic reports, referred to as localization attacks. This section focus on a tracking problem, known as map-matching. Map-matching (MM) is the process of continuously identifying the position of a vehicle on the road network given noisy location readings [Kubicka et al., 2018]. However, map-matching can also be used as an adversary tracking/locating a user as detailed in this section.

In the context of MM, it is typically considered high frequency of reports when reports are made up to every 1 minute. Any value above this interval is considered low frequency of reports, and commonly, low frequency MM techniques are evaluated up to a maximum of 5-6 minutes [Hashemi and Karimi, 2014]. In the context of LBSs however, 5 to 6 minutes is still considered continuous reports. Nevertheless, using a MM technique allows to evaluate the impact of frequency in highly continuous reports and consequently, assess the privacy level under the full range of frequencies.

Several map-matching techniques have been proposed following different approaches [Kubicka et al., 2018]. In this thesis we focus on MM techniques that are robust to noise, as an LPPM can apply additive noise, and effective on low frequency of reports, which results in sparse data. A seminal work fulfilling these criteria is found in [Newson and Krumm, 2009], where their method is evaluated over frequency of reports (referred to as sampling period) varying from 1 second to 600 seconds and over the addition random Gaussian noise to the GPS readings with multiple standard deviation values. A follow up on this work was made by Jagadeesh and Srikanthan [Jagadeesh and Srikanthan, 2017], where locations were measured with cellular network positioning instead of the GPS. The measurement error from the former positioning system is higher by almost 2 orders of magnitude and therefore the MM technique was adapted to be more robust against noise.

Comparative results between [Jagadeesh and Srikanthan, 2017] and the seminal work from [Newson and Krumm, 2009] showed the former technique to be more robust to both low frequency of reports and noisy measurements. Consequently, we have implemented the MM technique from [Jagadeesh and Srikanthan, 2017], which we describe next. We refer the reader to [Jagadeesh and Srikanthan, 2017] for a more detailed explanation of the original method.

Let us denote $o_i \in \mathbb{R}^2$ as the location report (referred to as observation in [Jagadeesh and Srikanthan, 2017]) at timestamp i. This report is not obfuscated but it is assumed to be noisy due to measurement imprecision. The road network is a direct graph $G = (V, E)$, where V is a set of nodes representing intersections and endpoints of road segments and E is the set of these segments. A path p between nodes u and v is a sequence of edges $e_1, \ldots, e_n$ such that u is the tail of e_1 and v is the head of e_n. The objective of a MM algorithm is to find a path p that corresponds to a sequence of T locations given noisy observations $o_1, \ldots, o_T$. Towards this goal, an Hidden Markov Model (HMM) is used in [Jagadeesh and Srikanthan, 2017].

At each noisy observation o_i, the HMM's hidden states at time step i correspond to potential locations on the road where the user can be. We denote the k^{th} potential location at time step i by $s_{i,k}$ and the hidden true state by $s_i^* = x_i$. Given that the location measurement error can be assumed effectively to follow a Gaussian distribution with zero mean [Newson and Krumm, 2009; Jagadeesh and Srikanthan, 2017], the probability that the observation o_i was generated from state $s_{i,k}$, referred to as emission probability, is given by:

$$p(o_i|s_{i,k}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{g(o_i,s_{i,k})^2}{2\sigma^2}} \tag{2.16}$$

where σ is the standard deviation of the measurement error and $g(o_i, s_{i,k})$ is the great-circle distance, that is, the shortest distance along the surface of the earth, between the observation o_i and the state $s_{i,k}$. Note that from equation (2.16), it is clear that closer states to the observation will have a higher probability than farther states, as the denominator increases exponentially with the increase of the distance $g(\cdot)$.

The transition probability, that is, the probability that the vehicle moved from state $s_{i-1,j}$ to $s_{i,k}$ depends on both the circuitousness of the path and on the temporal plausibility, that is, if the travelled distance is plausible given the time interval between timestamps $(t_i - t_{i-1})$. To measure the circuitousness of the path, the authors of [Jagadeesh and Srikanthan, 2017] defined the following equation:

$$y(s_{i-1,j}, s_{i,k}) = \frac{d(s_{i-1,j}, s_{i,k}) - g(s_{i-1,j}, s_{i,k})}{(t_i - t_{i-1})} \tag{2.17}$$

where $g(s_{i-1,j}, s_{i,k})$ is the great circle distance between the states and $d(s_{i-1,j}, s_{i,k})$ the driving distance, calculated using Dijkstra's shortest path algorithm [Dijkstra, 1959]. For the temporal plausibility, the equation is given as:

$$z(s_{i-1,j}, s_{i,k}) = \frac{\max\left(f(s_{i-1,j}, s_{i,k}) - (t_i - t_{i-1}), 0\right)}{(t_i - t_{i-1})} \tag{2.18}$$

where $f(s_{i-1,j}, s_{i,k})$ is the free-flow travel time, in seconds, of the optimal path between the states $s_{i-1,j}$ and $s_{i,k}$. Finally, the transition probability comes in the form:

$$p(s_{i,k}|s_{i-1,j}) = \lambda_y e^{-\lambda_y y(s_{i-1,j}, s_{i,k})} \lambda_z e^{-\lambda_z z(s_{i-1,j}, s_{i,k})} \tag{2.19}$$

where λ_y and λ_z are empirically determined parameters from equations (2.17) and (2.18), respectively.

To compute the most likely path from the HMM, a Viterbi algorithm is used as follows:

$$\begin{aligned} V_{1,k} &= p(o_1|s_{1,k}) \\ V_{i,k} &= p(o_i|s_{i,k}) \max_j \left(V_{i-1,j} p(s_{i,k}|s_{i-1,j})\right) \end{aligned} \tag{2.20}$$

where $V_{i,k}$ is the joint probability of the most likely state sequence ending at state $s_{i,k}$ based on the observations $o_1, \ldots, o_i$. The index j that maximizes $V_{i,k}$ is stored for each potential location k as it points to the predecessor state $s_{i-1,j}$ that most likely lead to $s_{i,k}$. Consequently, the most likely sequence for observations $o_1, \ldots, o_T$ is obtained by saving the indices j at each timestamp that maximize $V_{i,k}$, starting in $\max_w V_{T,w}$. The path p is then obtained by concatenating the optimal (shortest) paths between successive states in the most likely sequence.

Using the shortest segments to connect the states might not be the optimal solution. Therefore, in [Jagadeesh and Srikanthan, 2017] is also presented an heuristic that uses features to take into consideration drivers' preferences and thus increase the likelihood of getting the right segment between states. However, this additional heuristic achieves only marginal improvements (c.f. [Jagadeesh and Srikanthan, 2017]) at the expense of computational power. Since we will be computing map-matching under several configurations (see Section 4.1.2), we did not implement the heuristic as to decrease execution time.

Returning to the problem defined in Section 2.4.1, MM is typically used as a preprocessing phase of an LBS service in which the noisy locations are mapped to the most likely position for x_i. Therefore, in our problem the user is considered

to already have the real location x_i, $\forall i$. Nevertheless, an adversary can use MM to track/locate users in a road given obfuscated location/versions of x_i. In this latter scenario, the location readings (observations) are the obfuscated locations.

As for measuring privacy, we can use the adversary error from equation (2.3) using z_i. However, a point-by-point metric would fail to assess the effectiveness of the tracking, as the Adversary Error (AE) could be 0 and the estimated trajectory be different from the true trajectory. This can occur for instance when the true location is at a cross-road and the true path crosses the matched path. In such case, the true position matches the MM estimation, but the paths only overlap on that single point. Thus, we further consider a trajectory metric from the original authors of the MM technique [Jagadeesh and Srikanthan, 2017], the F_1 score computed as:

$$\begin{gathered} precision = \frac{L_{correct}}{L_{matched}} \qquad recall = \frac{L_{correct}}{L_{truth}} \\ F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \end{gathered} \tag{2.21}$$

where $L_{matched}$ is the length of the output path, L_{truth} is the length of the corresponding ground truth and $L_{correct}$ is the length of the portions of the output path that overlap with the ground truth path. Intuitively, the precision and recall measure the length of the segments that were correctly matched as a fraction of the map-matching output and the true path, respectively. The F_1 score is then the harmonic mean between both metrics.

2.5 Discussion

Privacy in mobile devices sees several challenges arising from the inherent sensory and connectivity capabilities. Permission managers arm the users of these devices with control over the access to their resources. However, current permission managers fail at both protecting and warning users about the risks thus leading towards uninformed decisions [Bonné et al., 2017; Shen et al., 2021]. In fact, users feel their personal space violated when confronted with apps' intrusive practices [Almuhimedi et al., 2015; Shklovski et al., 2014].

Despite being positively adopted by users [Bonné et al., 2017; Andriotis et al., 2018b], runtime permissions have a major flaw in the number of permissions that are allowed without user consent or even awareness [Almuhimedi et al., 2015; Wijesekera et al., 2015; Calciati et al., 2020], due to the fact that after being accepted once, subsequent requests are in general automatically granted. Therefore, while prompting at runtime contextualizes the permission request by the need of the application and therefore helps to make an informed decision [Bonné et al., 2017; Andriotis et al., 2018a], automatically granted requests violate this contextual integrity [Wijesekera et al., 2015].

Expectation is important: if an app fully behaves as expected by the user then fewer privacy problems would arise [Lin et al., 2012]. However, the users' expectations and apps practices often diverge due to the lack of knowledge by the user [Lin et al., 2012; Bonné et al., 2017] or by apps' intrusive practices [Almuhimedi et al., 2015; Wijesekera et al., 2015]. No work has captured the expectancy of users under runtime permissions and therefore the analysis of the importance of the expectancy in privacy decisions and the respective dynamism with changing contexts is yet to be done, that is, whether the expectancy changes with changes in the context. To address this issue, in Chapter 3 we perform a field study to collect privacy decisions, the surrounding context, and respective expectations. With the collected data we identify which contextual factors impact both privacy decisions and user expectations, and subjectivity of such impact for each individual.

Towards improving privacy in mobile devices, researchers proposed context-aware protection to account for privacy's context-dependence [Acquisti et al., 2015]. One of the main challenges here arises from the lack of precise definition of context. While mobile devices have naturally rich sensory capabilities, the device context is highly dynamic making its extraction and modeling a difficult task. It is thus realistic that for dynamic devices an automatic extraction of context is required as system designers will not be able to predict every possible scenario. Moreover, these inference methods have to take into consideration imprecision, noise and missing data [Abowd et al., 1999].

In context-aware privacy, the base idea is to leverage user and device context towards privacy protection [Schaub et al., 2015a]. In this regard, the identification of which and how much data is relevant not only for the context inference, but also towards privacy preferences. In mobile devices, some works focus on user context [Zavala et al., 2011], others on device context [Tsai et al., 2017], and others on the combination of both [Olejnik et al., 2017]. However, a qualitative evaluation on which context data is relevant towards privacy is yet to be made. Such assessment is itself a relevant research problem as it is closely related to how to measure the privacy level, as measuring privacy is an integral step of the process. Furthermore, personalization approaches that resort to privacy profiles lack contextual features, that is, the profiles are built without taking into consideration different contexts. The open issue here is to evaluate whether privacy profiles with contextual features could improve automated privacy decisions. To tackle the evaluation of which contextual data is actually relevant towards privacy decisions, we performed an exploratory data analysis on our collected data in Chapter 3. Leveraging on such findings, in the same chapter we develop personalized and contextually-aware predictive models, which we then compare to the default Android permission manager.

Finally, the last identified issue in existing permission managers lies in the limited control over the trade-off between privacy and utility. Specifically, permission managers allow to have maximum privacy and no utility, by denying a permission, or zero privacy and maximum utility, by allowing the permission. This poses a risk due to potentially non-trustworthy providers that can share the data with other entities [Mendes and Vilela, 2017]. Olejnik et al. [Olejnik et al.,

2017] proposed a simplistic approach that allowed for the obfuscation of four data types: camera, location, microphone and storage. The authors reported that users found obfuscation useful and noted that more complex obfuscation techniques could be integrated. The challenge here however, is that apps can access resources for different purposes, thus generating different types of data, which in turn require distinct obfuscation techniques [Cunha et al., 2021]. For example, finding the nearest restaurant requires sharing a single location, while a navigation app would require continuous location reports. The use of the same LPPM in this example, would result in ineffective privacy protection in at least one of the apps, due to the correlations between reported locations. Due to the relevance of this type of data in mobile devices, we focused our attention to location privacy towards potentially obfuscating location reports before sending the data to providers.

In Section 2.4 we have identified the privacy degradation that advents from continuous location reports. In fact, when designing LPPMs for the sporadic release of data, reports have been assumed to be independent [Shokri et al., 2011; Oya et al., 2019]. However, there is no formal nor quantitative distinction between sporadic and continuous reports and thus, the distinction is often based on the type of LBS application [Shokri et al., 2011]. In Chapter 4 we empirically evaluate the effect of the frequency of updates in the privacy level of location traces. The findings of this analysis motivate the development of online LPPMs that consider the correlation between reports and that adapt to varying frequency of reports. Unfortunately, few online Geo-Indinstinguishable LPPMs have been proposed that consider the correlation between reports. An illustrative example is the Adaptive Geo-Indistinguishability from Section 2.4.2.3 which measures the correlation through simple linear regressions and increases (decreases) privacy if the correlation is high (low). However, this automatic adjustment comes at the expense of usability. Namely, in addition to setting the privacy budget ϵ, the user must also define four extra parameters: Δ_1, Δ_2, α and β. This is a crucial drawback on the usability of the LPPM, specially since fine-tunning the privacy budget ϵ can be challenging on its own [Kaaniche et al., 2020; Clifton and Tassa, 2013; Lee and Clifton, 2011; Hsu et al., 2014] and can even mislead with respect to the privacy guarantees [Oya et al., 2017b]. For practical applicability and wide deployment of LPPMs, the automatic adaptability to the varying correlation between reports must be considered, while minimizing the required configuration. Such adaptability must additionally be robust to varying frequency of reports. To address challenge, in Chapter 5 we propose a novel notion to design LPPMs that automatically adjust to varying frequency of updates and user velocity, while requiring minimal configuration. Such proposal can further be personalized to a single driver or for specific regions, and can be generalized for wide deployment. In the referred chapter we provide an empirical comparison with existing geo-indistinguishable LPPMs.

In summary, this thesis intents to improve privacy in mobile devices by focusing on automation, personalization and context-awareness. Additionally, to empower users with control over the trade-off between privacy and utility, we considered obfuscation techniques, particularly in the context of location pri-

vacy, a prevalent type of data in ubiquitous computing. Towards this end, this section identified open issues which are tackled in the following chapters.

Chapter 3.

Automated Privacy Protection through Prediction of Privacy Preferences

Contents

In the current age of information, the rich and pervasive data collection sparks new applications that foster advances in our society. In this context, smart and mobile devices are of paramount importance due to their inherent sensory capacity, allowing for user-tailored and context-aware services. However, this data exchange often weights on the privacy of each individual, whose practiced trade-off is not often perceived or even understood.

To empower users with control over their privacy, smartphones have implemented permission managers that control, with user oversight, which resources, such as sensors and data, can be accessed by each application. Under the runtime permission system, the current mechanism employed in both Android and iOS, applications must require user permission the first time they require access to a sensitive resource. When presented with the prompt request, the user may either deny or allow the request for this single time, which will enforce the app to request the next time it needs the same access, or allow indefinitely, an option that can then be changed, but seldomly is [Andriotis et al., 2018a], in the settings of the phone.

The runtime permission system has replaced the dated install-time permission model, in where users either allowed all permissions or refuse to install an application. This upgrade has allowed for fine-grained management and for permission requests to be contextualized by the necessity of an app to access a functionality; it has therefore been positively received by users [Bonné et al., 2017; Andriotis et al., 2018a]. The problem with this model however, is that, after being allowed once, the permission is typically automatically allowed on all subsequent occasions, including when the user is unaware that the app is running [Almuhimedi et al., 2015; Wijesekera et al., 2015], thus violating privacy's contextual integrity [Nissenbaum, 2004], or in other words, defying users' expectations.

Privacy as contextual integrity is a model that binds privacy to the appropriateness of gathering and disseminating data at each specific context [Nissenbaum, 2004]. In this model, context is not limited to time and location, but is instead an abstract sphere that describes a situation and thus can encompass the activity being performed, the roles and norms binding each involved entity, the cultural and political ecosystem, and any other information that characterizes the current status. In this regard, any given data practice might be both appropriate or a violation of privacy depending on the context and on the expectations of the user within that context [Nissenbaum, 2004]. In mobile devices, the expectation of a user is their mental model that describes the functionality of an app [Lin et al., 2012], i.e., what the app does and how it works.

Expectation is important: if an app fully behaves as expected by the user then fewer privacy problems would arise [Lin et al., 2012]. However, users' expectations and app practices often diverge due to the lack of knowledge by the user [Lin et al., 2012; Bonné et al., 2017] or by apps' intrusive practices [Almuhimedi et al., 2015; Wijesekera et al., 2015]. Expectations should guide app design and support privacy-aware decisions [Lin et al., 2012; Bonné

et al., 2017].

To summarize, automation is paramount as applications make hundreds of permission checks per day [Mendes et al., 2022a; Almuhimedi et al., 2015], rendering approaches that ask on every use unfeasible. However, for effective privacy protection, the automation must take into account user preferences at each context surrounding the privacy decision [Acquisti et al., 2015], and ideally in accordance with user expectation, such that contextual integrity is preserved. This chapter presents our contribution to this subject, whose main goal is to enhance automated privacy protection through context-aware personalization and by taking into account user expectation. Towards this objective, we started by collecting data through a succession of campaigns with volunteers under real world conditions as described in Section 3.1. In Section 3.2 we provide an exploratory data analysis on the collected dataset, with a focus on the relation between privacy decisions, their surrounding context and the user expectation. In Section 3.3, we leverage such relations to propose automated, personalized and context-aware privacy protection mechanisms. Finally, Section 3.4 presents the limitations and future work remarks, and Section 3.5 concludes this chapter.

This chapter makes the following contributions:

- To the best of our knowledge, this is the first field study to capture the expectation of users regarding runtime permissions at scale and in-situ, thus avoiding potentially aspirational responses that might not align with behavior [Acquisti et al., 2015]. We make this dataset available to interested researchers.
- We uncover a strong misalignment between app practices and the expectation of users. Specifically from the collected data, almost half of requests are unexpected by users, a ratio that mostly varies with the requested permission, category of the requesting app, the visibility of the requesting app and, more importantly, the user.
- We empirically unveil an intrinsic relation between the pair category of the requesting app – requested permission, and the context of the user. This relation advents from the fact that different applications are used under different contexts, therefore conditioning the permission requests that are prompted to the user.
- Privacy decisions see the strongest correlation with expectation, mainly due to the fact that 90% of expected requests are allowed by users. However, expectation greatly varies with each individual. Thus we conclude that not only is expectation personal but so is the importance of it in privacy decisions.
- We develop a personalized automated permission manager for prediction of privacy decisions by taking into consideration the expectation and the context of the user and the context of the phone, thus achieving a ROC AUC of 0.96 and an F1 score of 0.92. Without user expectation, which is the strongest correlated feature with privacy decisions but requires user input which we seek to minimize, we achieve a ROC AUC of 0.9 and an F1 score

of 0.88.

- Finally, our data shows that Android 9 default permission manager based on runtime permissions would have resulted in 15% privacy violations, i.e. allowed permission requests that were explicitly denied by our participants. Our automated solution is able to reduce the number of privacy violations by 60% when compared to a standard Android handset. Without using the expectation as input feature for the prediction, these violations can still be reduced by 28%.

3.1 Data Collection

Towards building personalized and context-aware prediction models, data about privacy decisions and their surrounding context must be captured to train such models. Privacy preferences have previously been collected from smartphones in different studies. However, existing datasets present limitations with respect to either the lack of contextual data [Liu et al., 2016] or runtime permissions [Wagner et al., 2013] or were simply not made available due to privacy concerns [Wijesekera et al., 2017]. The SmarPer dataset [Olejnik et al., 2017] contains both runtime permissions and contextual information collected by instrumenting Android, to intercept apps' accesses to sensitive data and therefore prompt the user for permission. In addition to collecting the permission response, SmarPer also collects contextual information regarding device status, foreground running application and semantic location of the user. This work served as baseline for our data collection tool and therefore our dataset strongly relates with the SmarPer dataset [Olejnik et al., 2017] with the following relevant differences. The data collected in [Olejnik et al., 2017] might be sufficient to support building machine learning models, however it is limited in the number of data points. Furthermore, in order to reduce the dimensionality of the features, the authors focused on a few popular applications and only in 4 permissions: location, contacts, storage and camera. Our dataset makes no restrictions on the apps and permissions, more than doubles the number of participants (93), and contains over 7 times more permission requests answered by participants (65261). Additionally, we collect more contextual features, such as whether the user is in an event, and the expectation of the user for each permission request that is prompted. Privacy as expectation has been previously framed in the context of privacy decisions in mobile devices and even shown that the expectations strongly vary with the conception (or misconception) about apps' functionality [Lin et al., 2012]. However, to the best of our knowledge, our dataset is the only that captures in-situ user expectations regarding app permissions. Finally, while the work in [Olejnik et al., 2017] contains static permission decisions, these were collected through exit surveys. Instead, we collect static permissions from the settings of the personal phones, thus translating real behavior instead of potential aspirational responses [Norberg et al., 2007].

Our dataset was collected in a set of campaigns spawning from the 27th of July 2020 up to the 12th of May 2021, as illustrated in Figure 3.2 with a total of 93

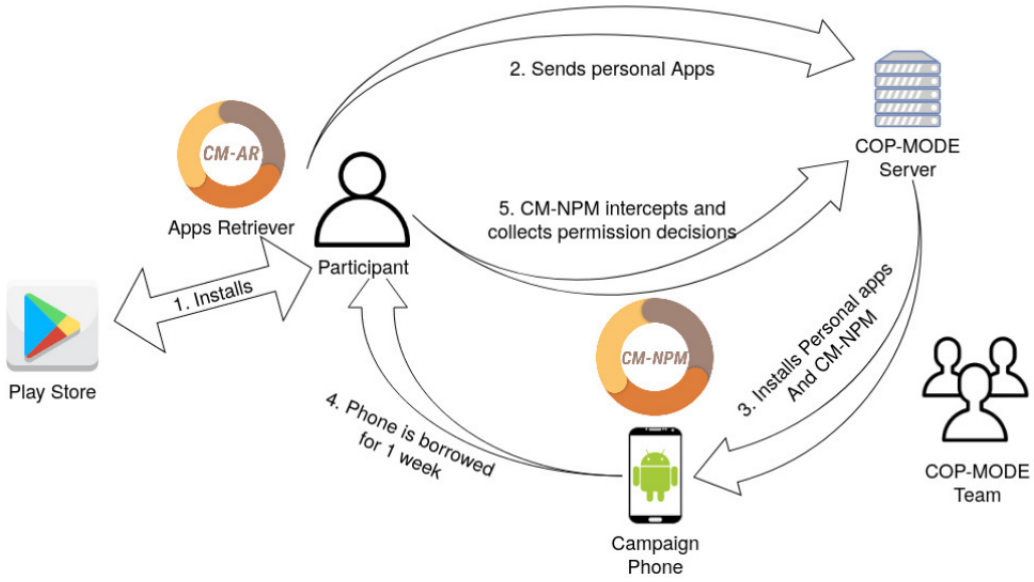


Figure 3.1.: Summarized diagram of a data collection campaign.

volunteers from Portugal. Participants were recruited through word-of-mouth, university mailing lists and from oral presentations. This resulted in the participation of 60 (64.5%) students, 11 (11.8%) researchers and the remaining 19 (20.4%) with diverse backgrounds. Some 66 (71%) participants were between 18-24 years old, 25 (26.9%) between 25 and 39 and 2 (2%) over 40 years old. While most participants were students, the professional areas of occupation diverged: 53 (57%) participants were from informatics engineering or computer science, 12 (12.9%) from other engineer fields, 8 (8.6%) from exact sciences other than engineering and the remainder spread through other occupations, retired or did not answer the question. Therefore, the dataset is skewed towards young adults and slightly more than half with an IT background.

The data was collected in two phases that can be summarized in 5 steps, as illustrated in Figure 3.1. In the first phase, the volunteer would sign-in to a campaign by installing our app COP-MODE Apps Retriever (CM-AR)[1]. By running CM-AR and consenting to the data collection, this app would collect the list of installed applications and respective permissions, including their current status (allowed or denied). This first set of data corresponds to our **static data**.

The second phase corresponds to a campaign of at least 1 week, where the participants used borrowed smartphones that came pre-installed with their personal apps, collected in the previous phase, along with Naive Permission Manager (NPM), our data collection tool and permission manager. NPM intercepts apps' permission checks and escalates them to permission dialogs that are prompted to the user, to collect the user decision and the surrounding context at the time of the prompt. Therefore, this second set of data is referred to as **permission requests data**. NPM and the collected data types are detailed in

[1] `https://play.google.com/store/apps/details?id=pt.uc.dei.copmode.appsretriever`

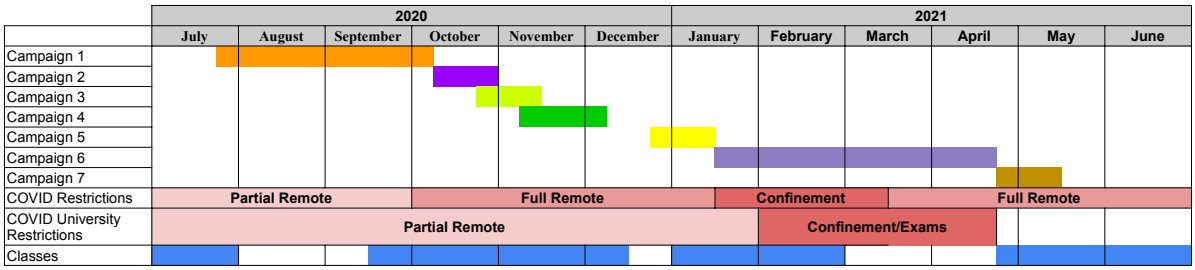


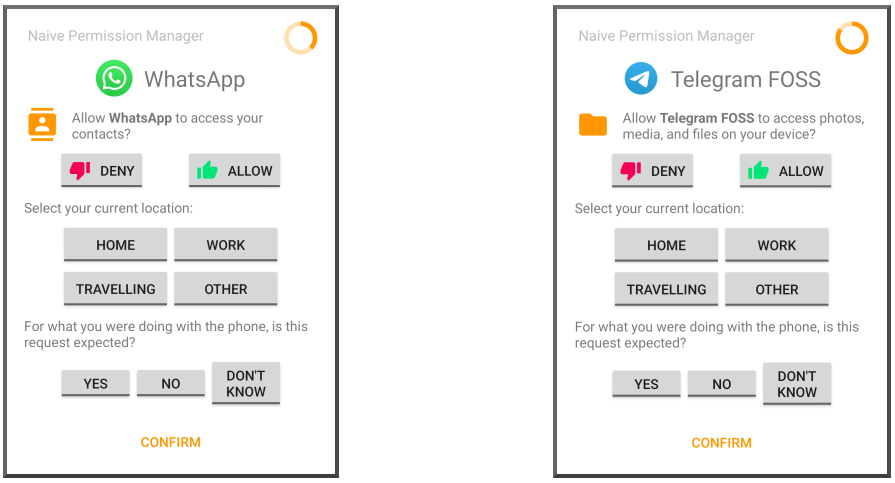
Figure 3.2.: Timeline of the COP-MODE's campaigns and the COVID-19 confinement periods. Partially remote periods correspond to when companies could have a limited number of workers in the office, thus scheduling remote or face-to-face work for teams in a phased fashion. In full remote, work from home was mandatory, unless otherwise unfeasible.

Section 3.1.2.

Participants that showed interested in continuing the experiments were sometimes allowed to continue using the phone for longer. Participation was rewarded with a gift card with the requirement of using the campaign phone as the main smartphone for the duration of the campaign. While this requirement was explicitly announced, we gave the voucher to every participant as a minimum acceptable number of requests (50) was answered by everyone – verified before handing the voucher. However, we did implement a metric and feedback mechanism to evaluate the phone activity. Simply, if the participant had not answered at least 5 permissions in the last 24 hours, he would be notified of inactivity to encourage engagement. After handing the voucher, we would send an email to the participant asking to fill an optional, brief and anonymous questionnaire to obtain some feedback. This questionnaire is presented in Appendix B.

3.1.1 Impact of the COVID19 on the Data Collection Campaigns

COP-MODE's campaigns were impacted by the COVID-19 in two major ways. First, from a participation point-of-view, as the amount of participants per campaign was lower than anticipated due to the difficulty in exchanging the campaign smartphones and concerns related to the potential of COVID-19 infection. As a result, 7 campaigns were necessary to reach the goal of 90 participants, instead of the 3 projected campaigns. Second, and more importantly, the contextual environment of the participants was predominantly their home due to partial or full quarantines, thus limiting the richness of the contextual data. Section 3.2 empirically evaluates this latter limitation. Figure 3.2 presents the timeline of the campaigns alongside with COVID-19 confinement periods. Because some campaigns were directed towards college students, we also depict COVID restrictions and teaching periods for context. From the timeline it is clear that most campaigns operated during a *partially remote* period where companies could have a limited number of workers in the office, thus scheduling remote or face-to-face work for teams in a phased fashion. A *fully remote* period



(a) WhatsApp requesting access to contacts.

(b) Telegram FOSS requesting access to storage.

Figure 3.3.: Examples of translated (from Portuguese) permission prompts issued by Naive Permission Manager.

refers to a setting where workers could only go to the office in cases where remote work was not possible.

3.1.2 Naive Permission Manager

To collect the data throughout the campaigns, the borrowed smartphone has our Naive Permission Manager (NPM) pre-installed alongside the participant's personal apps. NPM is both a permission manager and our data collection tool. Specifically, NPM intercepts *permission checks* performed by any app and prompts the user to either accept or deny the permission. At the time of the prompt, NPM further collects contextual data and additional information from the user as follows [Mendes, 2021a]:

- Requesting Application: name, package name, version code, UID, flags and app category from the Play Store.
- Permission: the name and group of the permission and the user response.
- Phone state: geolocation, plug, dock, call, screen and keyguard states, network connection type, list of apps running in the foreground and in the background. An application is in the foreground if it either has an activity in the foreground (visible to the user) or a service with a foreground notification. Apps running in the foreground and background have the same fields as the requesting application.
- User context: current time, semantic location, Bluetooth and WiFi devices in vicinity and whether the user is or is not in an event, as returned by their calendar. The semantic location was collected from user input, whose possibilities were "home", "work", "travelling" or "other" as illustrated in Figure 3.3.

- Expectation: the participant has to answer the question (translated from Portuguese) "For what you were doing with the phone, is this request expected?" with: yes, no or do not know. See Figure 3.3 for an illustration on how this data was asked to participants.

The permission dialog and context data are collected, stored locally and finally sent opportunistically to our project server.

3.1.2.1 Implementation Details

To intercept permission requests we could have either changed the operating system, use the accessibility features to read permission requests [Fu et al., 2019] or to require root [Olejnik et al., 2017]. We chose the latter as it is easier to setup and maintain and supports multiple operating system versions.

To intercept the operating system API calls, we implemented NPM as an EdXposed module [ElderDrivers, 2020]. EdXposed is a fork of the Xposed framework [rovo89, 2012b] to support newer Android versions[2]. Xposed allows modules to change the behavior of other apps or the system itself without changing any APKs [rovo89, 2012b], by adding an additional library (Xposed) to *app_process*, the process that spawns every application. Xposed then allows modules to hook (intercept) API calls, including the OS, and therefore modify the functionally before and after API execution [rovo89, 2012a].

The work in [Olejnik et al., 2017] focused on hooking API calls that collected data. This approach worked well at the time as it targeted Android API versions that were before Android supported runtime permissions. With runtime permissions, apps need to check if they have a permission before executing the API call that would collect the data. If the permission is denied (either previously denied by the user, or never requested), the app may or may not request the permission [Developers, 2022b], as it can check whether it has the permission without explicitly asking the user. This conditional execution led us to primarily hook *permission checks*, escalating them to permission requests in the form of the prompts illustrated in Figure 3.3, while intercepting *permission requests* only to override the result with the participant input given to the prompt created by NPM at the respective permission check. This latter interception allow us to bypass the Android default permission manager. Note that while we collect the data at permission checks, we refer to this data as *permission requests*.

In Android, permissions can have different protection levels and only *dangerous permissions* require an explicit request by applications [Developers, 2022b]. These permissions are considered "dangerous" because they allow apps to access sensitive data or resources that can affect the system and/or the privacy of the user. Therefore, NPM only handles, and therefore collects, permission requests related to dangerous permissions. Furthermore, because the default Android permission manager manages the permissions at a group level, a controversial

[2]The Xposed framework supports from Android 4.0.3 (API level 15) up to Android 8.1 (API level 27) [rovo89, 2018], while EdXposed supports from Android 8.0 (API level 26) up to Android 11 (API level 30) [ElderDrivers, 2020].

implementation decision [Calciati et al., 2020], so does NPM. That is, by default, if, for instance, a permission requires the read calendar permission and the user grants it, the app will automatically be allowed the write calendar permission on request. While one can argue about this feature from a privacy perspective, NPM follows this behavior in order to replicate Android's permission manager from a data collection point of view.

When a dangerous permission check call is made by an app, NPM prompts the user as illustrated in Figure 3.3a and collects the contextual data aforementioned. Similarly to the work in [Olejnik et al., 2017], we cache the answer for 30 minutes, thus returning the same answer for the given app and permission group for this duration, in order to avoid warning fatigue [Felt et al., 2012]. The permission icon and permission description are obtained directly from the Android operating system, so as to not bias the response. To avoid breaking functionality, NPM does not handle permission requests from system apps, letting the Android native permission manager handle those.

3.1.3 Dataset Sharing and Ethics

Due to the limitations in existing datasets, we make an anonymized version of our dataset available to interested researchers [Mendes, 2021a]. All shareable data is stripped of identifiable information. Specifically, application names are removed and package names are one-way hashed with random salt, calendar events are reduced to flags that indicate whether the user is at an event, geographic location was removed (semantic location given by the user in the prompt is kept), and information about devices in the neighborhood (wi-fi and Bluetooth) were removed. Our data collection tool, Naive Permission Manager, is open-sourced and freely available [Mendes, 2021b].

This research was approved by the Ethics Committee, Department of Computer Science and Technology, University of Cambridge, and by the Ethics Commission of the Faculty of Sciences and Technology of the University of Porto.

3.2 Exploratory Data Analysis

As described, our data falls in three different sets: **static data**, which contains the application list of each participant, and the corresponding permission settings; **permission requests data**, which is the runtime permission data and respective context collected from the permission prompts; and the **questionnaire data**, the anonymous responses to the survey described in Appendix B. The following sections provide a preliminary characterization of the dataset by exploring each of these sets. Specifically, we start by analyzing the responses to the questionnaire in Section 3.2.1, which serves as motivation for the development of better automated and ideally context-aware approaches to privacy enforcement. Section 3.2.2 describes the static data set. Finally, Section 3.2.3 provides an in-depth discussion on privacy decisions using the permission requests data set, with a particular focus on the grant result, that is, whether the user allows or denies permissions and user expectations.

3.2.1 Questionnaire Data

As aforementioned, at the end of a campaign, and after handing the reward, an email would be sent with a link to an optional and anonymous questionnaire to collect some feedback on the experience of the campaign. Appendix B presents these questions, and below we examine the answers.

From the 48 participants that answered the questionnaire, 11 (22.9%) considered the default Android permission manager enough to manager permissions, 14 (29.2%) were on the fence and 23 (47.9%) said it was insufficient. Additionally, 38 (79.2%) were highly surprised (at least 4 on a scale from 0 to 5) with the number of requests made from apps that were intercepted by our collection tool, the NPM. Some of these participants (41 out of 48) expressed concerns about specific apps requiring permissions such as access to the microphone and location, potentially due to the misalignment between app functionality and the requested permissions. Other participants thought it was strange that apps were requesting permissions when they were not being used. However, apps requiring permissions while running on the background can be a legitimate use case. For instance, WhatsApp retrieves messages from the server in the background while requiring the contacts permission to identify the senders of the messages based on the local contacts. Unfortunately, such actions can be unexpected by users and could therefore be improved with visual cues in the form of notifications, for example. This is an example on the dichotomy between expectation and legitimacy that sees roots in user's lack of knowledge [Lin et al., 2012; Felt et al., 2012].

Of these 41 participants that expressed concerns about permission requests, 16 (30%) mentioned the permission PHONE being requested by apps where this need is not clear for these participants. Examples of these apps are YouTube, WhatsApp, Instagram, Twitter, and other less known apps. We hypothesize that most of these apps require the PHONE permission to have access to unique identifiers, such as hardware identifiers (e.g. the International Mobile Equipment Identity (IMEI)), to allow for cross-app tracking. This functionality is not obvious to the users as the text in the request only mentions doing and managing calls, but it is made possible by the access to this permission. Confirming this hypothesis is out of the scope of this work as it would require code analysis. Nevertheless, this result might be a good indication that Android should create a new permission related to the access to unique identifiers. In this context it should be noted that the Android API is making efforts to limit the access to unique unchangeable identifiers [Developers, 2022a].

When asked about which contextual data was most important towards permission decisions the answers were varied. Some participants mention the requesting app as the most important information, hinting the relevance of trust in the developers as discussed in [Bonné et al., 2017], others focused on particular permissions, some mentioned the semantic location or the activity that they are performing and one person mentioned the expectation. A common occurrence however, was the purpose of the access to the resource, that is, the motive why the app requires the access to the data/resource. This piece of information has

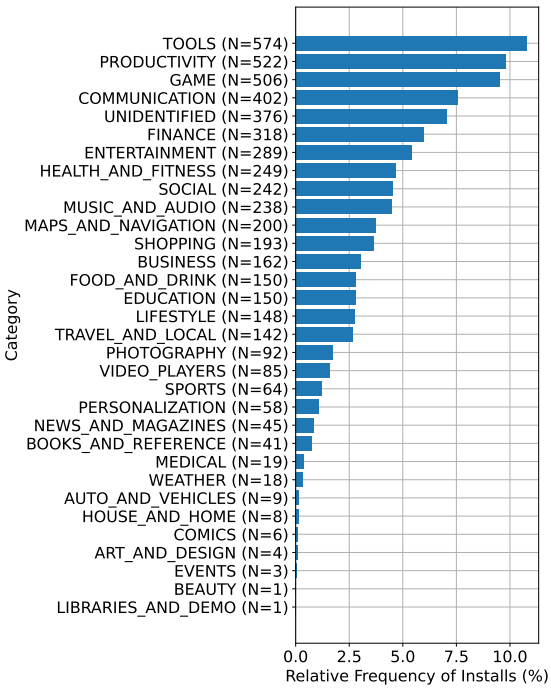


Figure 3.4.: Relative histogram of the installed non-system apps per category. The number of installed apps per category is denoted by N.

been shown to be critical in the context of privacy decisions [Shih et al., 2015; Lin et al., 2014; Liu et al., 2016; Smullen et al., 2020] and for the expectation of the user [Lin et al., 2012; Wijesekera et al., 2015]. Unfortunately, mobile devices lack purpose-specific permission control and inferring the purpose is a challenge in itself as it requires static analysis [Lin et al., 2014] and/or network traffic analysis [Enck et al., 2014].

3.2.2 Static Data

The static data was collected directly from the participants' personal phones and consists in the list of installed apps and respective permissions. From the 93 participants, a total of 30768 applications were installed (3926 distinct), of which only 5315 (17.27%) (1737 (44.24 %) distinct) are non-system apps. Participants had in average 57.2 non-system applications installed, with a standard deviation (*std*) of 29.5, and a maximum of 162. Figure 3.4 presents the number of installed non-system applications per app category, as retrieved from the play store, and Table A.1 presents the top installed apps per each category. It is observable from the histogram that TOOLS, PRODUCTIVITY and GAME are the most predominant type of installed apps in our static data. This result can a be a consequence of the biased sample of volunteers, in where the majority of participants were young adult students.

From the non-system apps, an average of 18.23 permissions ($std \approx 16.83$) are declared per application, up to a maximum of 202. However, of the declared permissions per app, only an average of ≈ 5.12 ($std \approx 3.76$) are dangerous

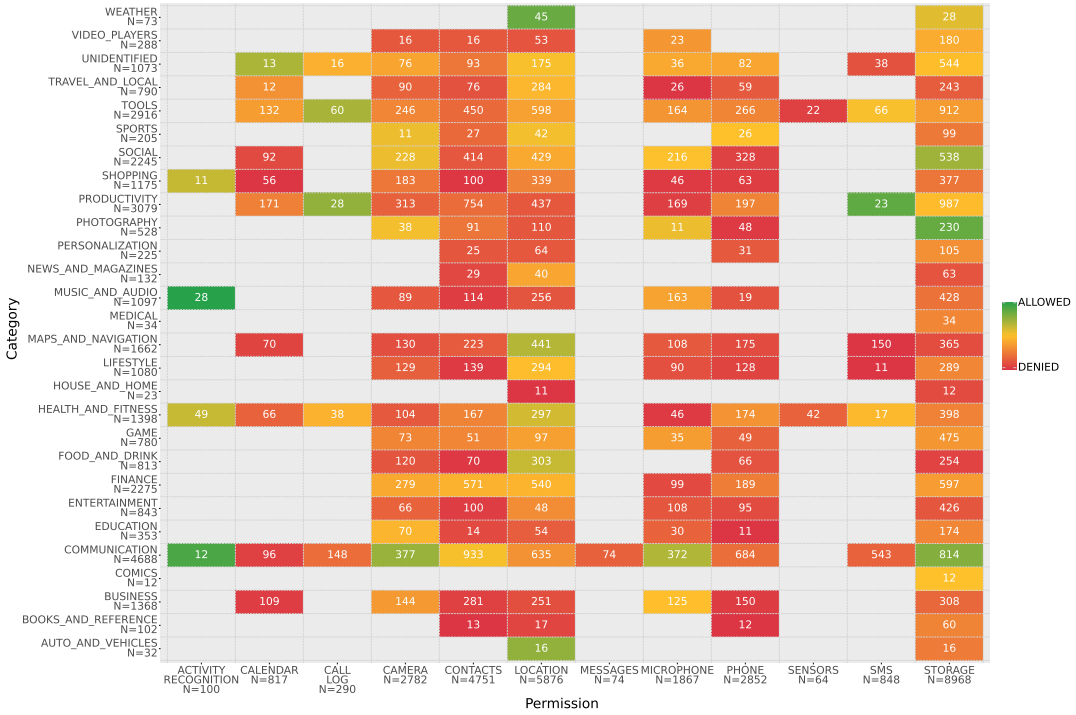


Figure 3.5.: Average permission status per dangerous permission group and app category for the static data, showing only non-system applications and permissions with at least 10 apps. The "N" in the axis labels and the number in each cell are the number of permissions in the dataset for the given permission group or app category and for the pair permission–category, respectively.

permissions, as defined by Android [Developers, 2022b]. Figure 3.5 presents the average permission status for each dangerous permission group and each app category of the personal non-system applications. From this plot, it is clear that most permission groups tend to be in a denied state, and the ones showing mostly ALLOWED, have a low number of occurrences in the dataset. This is potentially a consequence of the current Android permission system, in where a dangerous permission is denied if either the user has explicitly denied it, or if the permission was never asked [Developers, 2022b]. Therefore, from the static dataset we cannot know with certainty whether the denies in the static data are defaults or explicitly set. Additionally, these settings might misrepresent privacy preferences as users are unaware of intrusive data collection practices [Felt et al., 2012; Almuhimedi et al., 2015]. Consequently, this data has low utility with respect to privacy preferences.

3.2.3 Permission Requests Data

From the 93 participants, we collected 2180302 permission requests at an average of 836.85 requests per day and per participant with a standard deviation (std) of 19.15, or 34.87 ($std = 0.8$) per hour. These numbers prove that an ask-on-every-time approach, the ideal privacy choice, is infeasible in practice. Note however, that this number varies with general phone usage, including the type of installed

apps. Of the total requests, 65261 (2.99%) were answered by participants, corresponding to an average of 25 ($std = 0.42$) answers per day, per participant. Permissions not answered by the participant were either answered by the cache, timeouts or dismissed. Section 3.2.3.1 and Section 3.2.3.2 thoroughly analyze the grant rate and user expectancy, respectively.

3.2.3.1 Analysis of Grant Rate and Privacy Violations

From the 65261 answered requests, participants allowed 43263 (66%), while denying the remaining 21998 (33%); that is, users grant 2 out of every 3 permission requests. These results strongly contrast with the grant rate reported in [Bonné et al., 2017], where participants allowed 86% of requests. This disparity occurs due to the fact that the data in [Bonné et al., 2017] was collected from Android's runtime permission prompts, which only occur when apps have their permissions denied and are running a foreground activity. However, after being allowed once, applications can access the resource any time even without the user being aware, until it is explicitly denied through the phone settings. Our permission dialog, on the other hand, prompts users on every *permission check*, unless the same permission has been answered in the last 30 minutes as previously explained, including from background apps, regardless of whether they previously had the permission allowed.

Figure 3.6 presents the distribution of the grant result per user. From the plot we can already observe that there are widely distinct users regarding privacy preferences. Namely, there are users that allow all permissions, which correspond to users with only a green bar, and users that deny (almost) all requests, thus having only a red bar. Finally, there are users in between, corresponding to users with more selective privacy choices. This selectiveness can depend on the category of the requesting app, the requested permission and even the circumstances (context) of the request. Below we explore the grant result under these different aspects.

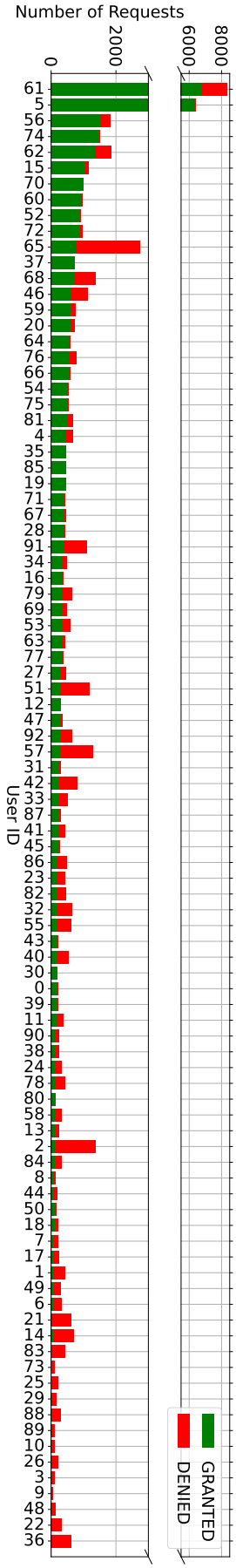


Figure 3.6.: Grant result distribution per user as a stacked histogram.

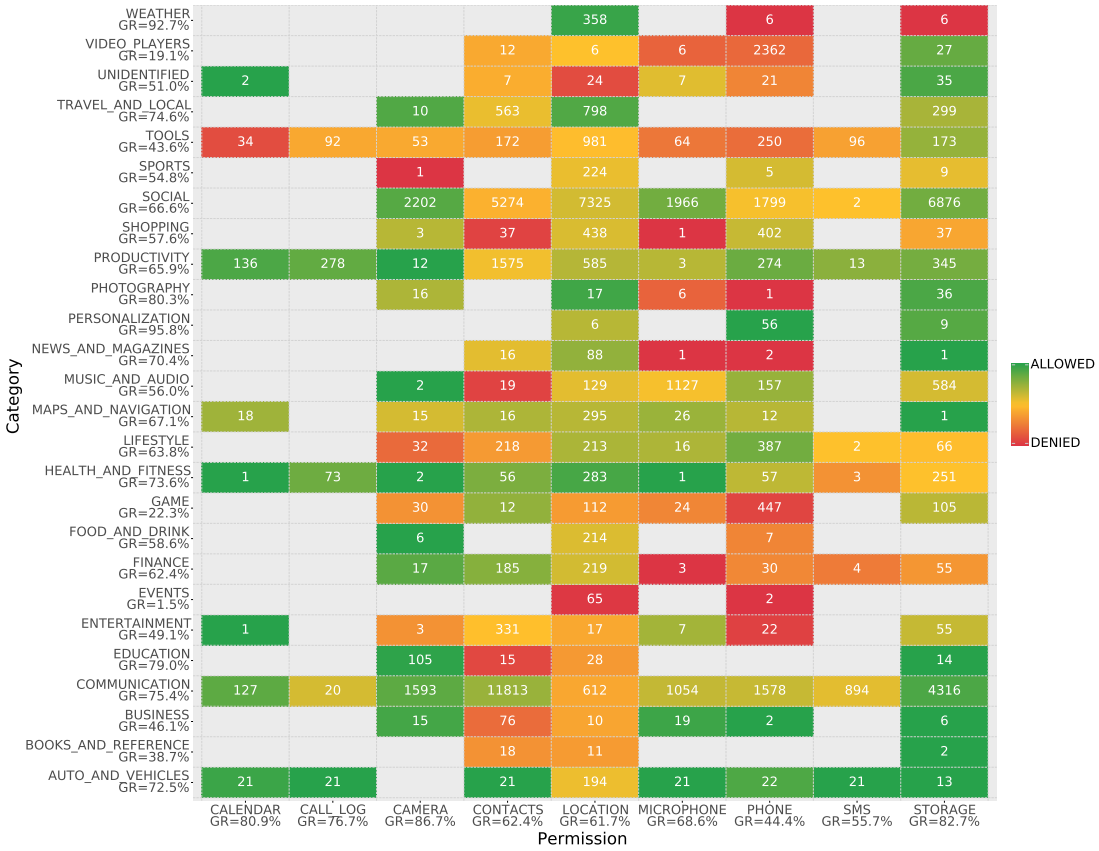


Figure 3.7.: Average grant result for each pair of category-permission. The number in each cell is the number of requests for the respective pair category-permission group, and GR is the grant rate for the respective category or permission. Categories and permissions with less than 10 requests were removed.

To have a holistic view on which permissions are allowed Figure 3.7 presents the average grant rate, i.e. the percentage of allowed permissions, per category (y axis) and per permission group (x axis), where dark green corresponds to all permissions allowed and dark red to all permissions denied. From the plot we can observe that the majority of categories have grant rates in the interval of $[45, 75]\%$. However some categories present grant rates of over 80% or closer to 0%, but the number of requests from these type of apps is rather small. The exceptions to this observation with a considerable number of requests are the WEATHER category, where 93% of the 370 requests were allowed, and GAME (730) and VIDEO_PLAYERS (2413) where almost 80% of requests were denied. It is possible that these latter categories see most of their requests denied because the permissions are not necessary for their primary functionality, which typically leads users towards denying [Bonné et al., 2017]. For instance, some of the requested permissions from apps in the GAME category, such as PHONE, MICROPHONE and CONTACTS are not intuitive with respect to the functionality of this type of apps. This is also true for VIDEO_PLAYERS that request access to the LOCATION or CONTACTS. As for the grant rate per permission group, the rate is near the interval of $[45, 85]\%$. CAMERA, STORAGE and CALENDAR permissions are allowed over 80% of the time, which might

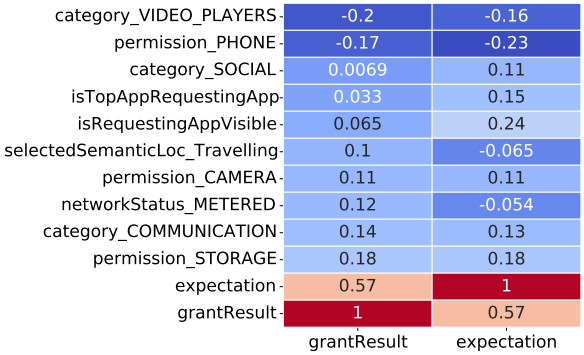


Figure 3.8.: Pearson correlation coefficient for the grant result and expectation with all other features, where categorical features are one-hot encoded, requests with UNKNOWN expectation value removed and coefficients in the interval of $]-0.1, 0.1[$ are omitted.

indicate that when apps request these permissions, there are contextual cues or a clear necessity that lead users to allow.

Changes in the context can also influence privacy decisions, due to privacy's contextual dependency [Acquisti et al., 2015]. To assess the importance of each collected feature in the grant rate, we present the relevant Pearson correlation coefficients in Figure 3.8 and the mutual information gain in Table C.1b between the grant result and all other features. Categorical features were one-hot encoded, for this purpose. Both the correlation coefficient and information gain appoint user expectation as having the strongest relation with privacy decisions. Due to this relevance, the analysis of user expectation is given in Section 3.2.3.2 and in the reminder of this section we focus on the relation between the grant result with the other features. After the expectation, the most importante features according to the information gain and Person correlation coefficient are some permissions and app categories, the visibility of the requesting app (isRequestingAppVisible), the location of the user (selectedSemanticLoc) and the network status. The following subsections analyze the grant result with respect to each of these latter three contextual features, and finally compare our users' privacy decisions with the default Android 9 automated permissions.

Visibility of the Requesting Application Previous work [Wijesekera et al., 2015] has identified the visibility of the requesting app has one of the most important contextual feature guiding users towards allowing or denying a permission request. Follow up work from the same authors [Wijesekera et al., 2017; Tsai et al., 2017] focused on this feature towards predicting the grant result. However, contrary to their conclusions, their feature analysis revealed that the visibility of the application was the feature with the lowest information gain, as can be seen in Appendix A and Appendix B of [Wijesekera et al., 2017]. In our dataset the information gain is almost 8 times higher (c.f. Table C.1b). However, from the 65261 answered requests, users allowed 68% of requests coming from visible apps and 62% of requests from background apps. This discrepancy is lower than anticipated, which signals that the visibility of the requesting app as a single feature has in fact a low impact in the grant result.

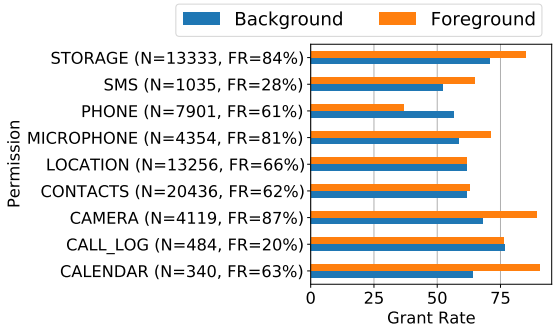


Figure 3.9.: Grant rate for each permission and whether the requesting app was foreground (visible) or background. The "N" is the number of requests per permission and "FR" the foreground ratio, that is, the percentage of requests that came from apps that were visible to the user at the time of the request.

While the overall grant rate between foreground and background requests varies little, this rate can strongly depend on the pairs visibility-category of the requesting app and visibility-requested permission. Figure 3.9 presents the grant rate for each permission and each visibility of the requesting app. From this plot we observe that CONTACTS, CALL_LOG and LOCATION requests are allowed equally regardless of the visibility of the requesting app. STORAGE, SMS, MICROPHONE, CAMERA and CALENDAR are more often allowed when requested from the foreground than from the background. Finally, the PHONE permission is the only permission that is more often allowed from the background. We have have no justification for this latter result as a limitation of the dataset is not collecting the reasoning for some privacy choices [Mendes et al., 2022a]. Similarly, figure 3.10 shows the grant rate per app category and per visibility requesting app, were one can observe the disparity depending the different visibility values in most categories. Thus we conclude that while the visibility of the requesting app alone has low impact on the privacy decision, which contrasts previous findings [Wijesekera et al., 2015], the combination with other features such as the permission and category might improve prediction performance. We further examine this correlation in Section 3.3.1 when evaluating the relevancy of each feature in the performance of a predictor of the grant result.

User Location and Network Status Contrary to the visibility of the requesting app, which describes the context of the phone, the location and network status relates to the context of the user. We saw a strong variation in the grant rate depending on these two latter contextual features. Below, we start by analyzing each of the features separately and then join them as they are also correlated.

According to the information gain in Table C.1b, user location has some impact on the grant result. Looking at the grant rate, users allowed 65% of requests while at home, 85% while traveling, 74% while at work and 57% in other locations. This variance is relevant, specially for when the user is traveling, where they accept almost 9 out of 10 requests. There are two main reasons for the ob-

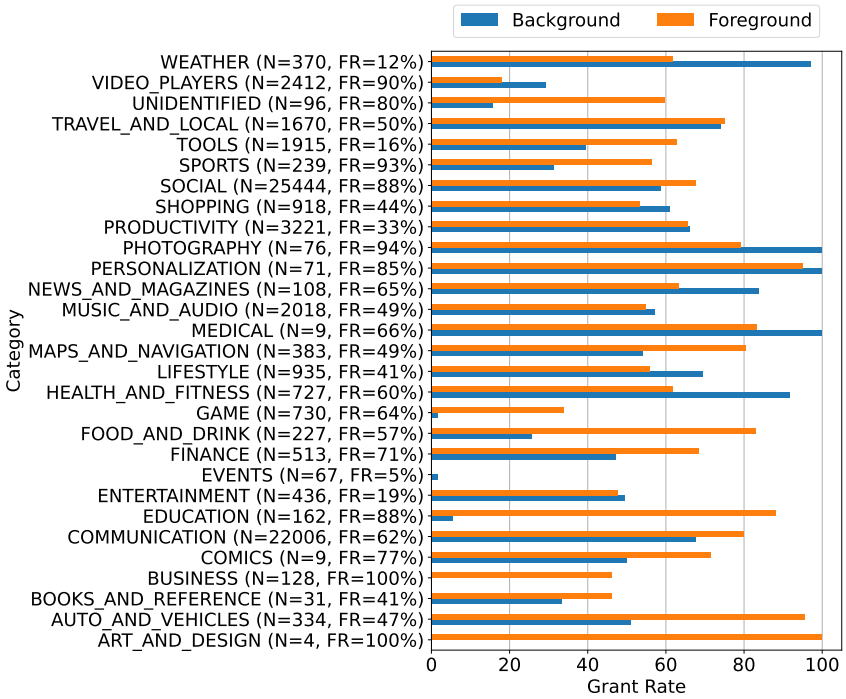
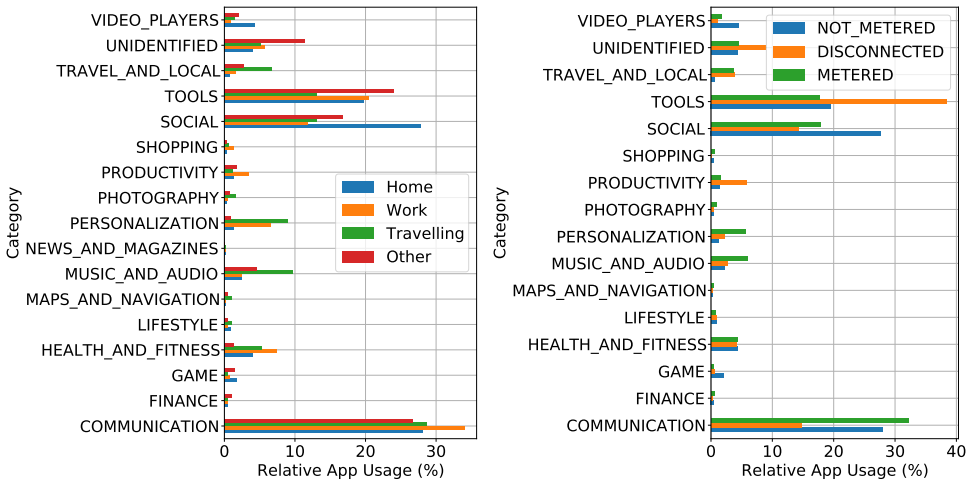


Figure 3.10.: Grant rate for each category and visibility of the requesting app. The "N" is the number of requests per permission and "FR" the foreground ratio, that is, the percentage of requests that came from apps that were visible to the user at the time of the request.

served variances in the grant rate for each location: privacy preferences vary with the user location; and the application usage also varies with each location.

As shown in Figure 3.7, different app categories have varied grant rates. Therefore, if the user uses different applications in different locations, it is expected that the grant rate also varies implicitly. Figure 3.11a presents the relative application usage in percentage given by the applications in the foreground, per semantic location, for the full dataset. The relative usage is made per location, such that a fair comparison between locations is achieved, as the dataset is strongly skewed towards the home location. From the plot we can observe that COMMUNICATION, SOCIAL and TOOLS are the most used apps regardless of the location. Additionally, we can clearly see that there are some trends in the type of application usage and the location of the user. Specifically, SOCIAL and VIDEO_PLAYERS apps seem to be predominantly more used at home than in other locations. TRAVEL_AND_LOCAL, PHOTOGRAPHY, PERSONALIZATION and MUSIC_AND_AUDIO are more used when travelling, which is expected except for the PERSONALIZATION category, while TOOLS are less used when travelling when compared to the other locations, which is also intuitive. Finally, both MAPS_AND_NAVIGATION and LIFESTYLE see a stronger usage when travelling. However, the use of this type of apps was strongly impacted by the COVID19 mobility restrictions, thus presenting a small overall usage. To conclude, the correlation between the location and the grant result can be explained not only because of personal preferences in each



(a) Per user location. (b) Per network status.

Figure 3.11.: Relative app usage as measured by the relative number of requests in where app from each category were in the foreground. Values inferior to 0.1% were removed from the plot to simplify visualization.

location but also due to the types of apps that are used in each context, which, as we have seen in Figure 3.7, can have diverging overall grant rates.

Similar conclusions can be made for the network status. From the answered permission requests, 1856 (2%) were captured while the phone was disconnected, 20591 (20%) while connected to a metered network and 80084 (78%) while connected to a non-metered network. These numbers indicate that most people are continuously connected to the Internet, although some impact of COVID19 travel restricts can influence this result. The user allows 77% of permission when using a metered network, 64% when using a non-metered network and only 47% when offline. Again, this discrepancy is relevant, as also highlighted by the information gain from Table C.1b. However, and similarly to user location, the network status is an indication of the context of the user, which in turn influences the apps that are used.

Figure 3.11b presents the relative app usage per category given by apps in the foreground for each of the network status. We can observe that TOOLS and PRODUCTIVITY apps are mostly used while offline, while COMMUNICATION and SOCIAL are mostly used online, which is expected. PHOTOGRAPHY, PERSONALIZATION, MUSIC_AND_AUDIO and MAPS_AND_NAVIGATION are mostly used in a metered connection, which as we have seen from Figure 3.11a, are typically used when travelling. Similarly, TRAVEL_AND_LOCAL sees most use in this location, nevertheless it also presents significant use when disconnected. From these observations we conclude that user context, which is partially described by their location and the network status, influences the type/category of applications that are used and therefore the apps that request permissions at these times. In other words, the category of the requested app and the required permission encapsulate con-

Location	Network Status	Count	Location Count (%)	Grant Rate (%)
Home	DISCONNECTED	923	1.69	41.93
	METERED	6600	12.11	74.71
	NOT_METERED	46997	86.20	63.61
Other	DISCONNECTED	129	5.81	51.16
	METERED	1273	57.34	59.15
	NOT_METERED	818	36.85	55.87
Travelling	DISCONNECTED	128	3.12	68.75
	METERED	3423	83.39	85.83
	NOT_METERED	554	13.50	83.57
Work	DISCONNECTED	126	2.85	59.52
	METERED	2433	55.10	81.38
	NOT_METERED	1857	42.05	66.34

Table 3.1.: Number, relative count and grant rate of permission requests per semantic location and network status. The grant rate is the percentage of permissions allowed for each pair of location–network status.

textual information that, while potentially insufficient to describe user context, give clues about the state of the user.

As aforementioned, either location, network status or even both are insufficient to effectively describe the variance in the grant rate. For instance, within a single location, the grant rate varies for each network status and vice-versa. Table 3.1 presents these values for each pair of location-network status. The first observable result from this table is that the location of the user and the network status are strongly correlated. Looking at the "Location Count (%)", both home and traveling locations have predominant network status. Specifically, when the user is at home, unmetered connections are used over 86% of times, while when traveling, metered connections are used 83% of times. At work and other locations, the connection status is more balanced between metered and unmetered connections. However, these latter ratios might vary greatly with each individual. Finally, while some previously mentioned trends endure, the grant rate strongly varies for each pair of location-network status. For instance, the highest grant rate in any location is when the user is using metered connections and the lowest is when the user is disconnected. However, under metered networks for instance, if the user is travelling, over 86% of requests are allowed, but if the user is at a location other than the specified three, the grant rate lowers to 59%. These observations allow us to conclude that while location and network status are related, both give contextual cues, even if in the form of the apps that are used in such contexts. In turn, these cues can be leveraged by a predictive model towards automating privacy decisions. Section 3.3 details this endeavour.

Comparison with Android 9 Permission Manager It is possible to assess the number of privacy violations of the default Android Permission system. Specifically, we can measure the number of requests that would have been allowed by Android's permission manager but instead were denied by the user, as after

being accepted once the permission is generally allowed. We do so by counting the number of requests for each pair of app–permission that were denied after being accepted once. *From the 65261 user answered permissions, 9950 (15%) were denied by participants that would have otherwise been allowed by Android 9 permission manager.* Note that while this number seems rather small, this corresponds to privacy violations 15% of time. Additionally, due to the fact that the default Android permission manager would prompt a request until the user allows a permission for each specific app, achieving this violation rate with an Android system would require users answering an average of 129.5 permission prompts (median of 64). The privacy violation ratio is identical to the results from [Wijesekera et al., 2017] (from their table III, 15.39% would be wrongly allowed). However, the number of prompts is significantly different, as we saw a median of 64 prompts per user, whereas [Wijesekera et al., 2017] reported 12.34. This disparity is justified by the fact that their work came before the introduction of runtime permissions in Android, and therefore the set of dangerous permissions that the authors considered differs from our set, the default Android dangerous permissions.

3.2.3.2 Analysis of the Effect of User Expectation

The strongest correlation with the grant result, that is, the privacy decision to either accept or deny the permission, is user expectation with a coefficient value of 0.57, as evidenced in Figure 3.8. Similarly, the strongest information gain to the grant result is the expectancy, as illustrated in Table C.1b. This relation can be further analyzed by looking at the distribution of the grant result for each expectation value, which can be EXPECTED, UNEXPECTED or UNKNOWN, where this latter value corresponds to when the user was unsure whether the request was expected or unexpected. Specifically, when users expect a request, they allow it 92% of the time, while allowing only 38% of the requests that are unexpected. When in doubt, the user accepts 2 out of 3 ($\approx 67\%$) requests, which is inline with the global grant rate. These results indicate that developers should explain the rationale behind permission requests, a possibility implemented since Android 6.0 and iOS6, that has been shown to help with privacy decisions [Tan et al., 2014], yet it is still largely unused in practice [Liu et al., 2018b].

From the 65261 user answered requests, 52% were EXPECTED, 46% were UNEXPECTED and the remaining 2% were UNKNOWN, that is, the participant was unsure whether the request was expected or unexpected. In other words, almost half of requests are unexpected. This result reveals a strong misalignment between app practices and the expectation of users, and therefore calls for more transparency from app developers, as an informed user is a comfortable user [Lin et al., 2012], and endorses the use of the minimum required permissions for the functionality of the app.

Similarly to the grant result, user expectation varies for each category and each permission. Figure 3.12 presents the average expectation for each pair of category-permission and the grant and expected ratios, where the latter is the percentage of EXPECTED requests, for each category and permission. It is

Figure 3.12.: Average expectation for each pair of category-permission, with requests with UNKNOWN expectation removed. The number in each cell is the number of requests for the respective pair category-permission group and ER and GR are respectively the expected ratio (percentage of requests that were expected) and grant rate for the respective category/permission. Categories and permissions with less than 10 requests were removed.

clear from the plot that some pairs of category-permission are often expected, such COMMUNICATION-STORAGE, while others are often unexpected, such as GAME-PHONE. In fact, the PHONE permission sees the lowest expected ratio from all permissions at only 22.3%, closely followed by the CALL_LOG permission with 23.6% of expected requests. Our reasoning for these low values lies on their lack of understanding surrounding these two permission groups. The CALL_LOG permission group was created in Android 9 by moving some of the PHONE permission to the former group. At the time of the study it is possible that some users did not have Android 9 in the personal phones and were therefore first exposed to this permission group during the campaign. Furthermore, the PHONE permission allows not only checks on the phone state, but also to make and manage calls and even access unique identifiers[3], functionalities that might not be evident to the user. In fact, previous work [Felt et al., 2012] showed that less than 5% of 85 respondents correctly identified the functionality of READ_PHONE_STATE, a permission within the PHONE group. Almost as low in the expectation ratio as the PHONE and CALL_LOG, comes

[3]Access to unique identifiers is now restricted since Android 10 [Android Developers, 2019].

the SMS permission with less than 30% of requests expected, as illustrated in Figure 3.12. Contrary to the other two permissions, the functionalities allowed by the SMS group are arguably clearer [Felt et al., 2012]. A possible reasoning for this low expectation lies in the number of SMS requests that originate from the background. Only 28% of these requests (c.f. in Figure 3.13) originate from apps that are visible to the user, making the user unsure on the need for these requests. It is possible that the functionality provided by this permission group is not worth for the user. Confirming this would require a survey. However, the sensitivity of the SMS and CALL_LOG groups has led Google to restrict their usage to the default SMS/Phone/Assistant handler or as core app features [Google, 2018].

Finally, we should note that permission prompts from the Android system do not allow the user to distinguish the different permissions within a given group. That is, if an application requests a permission to read a resource such as the SMSs, contacts or calendar or requests a permission to write the resource, the same permission request would be prompted to the user. The read permission can be more sensitive from a privacy perspective as it can allow for an app to access all messages or contacts, but the write permission can for instance, incur in costs, such as sending messages. Regardless of whether the user is aware of these implications, a finer grained management over the permissions, instead of only at the permission group level, would potentially improve the perception for the requirement of the app and increase privacy controls. We leave for future work the implementation and evaluation of such system.

In contrast with the limited influence of the visibility of the requesting app in the grant result as discussed in Section 3.2.3.1, our data reveals that the expectation is influenced by it, as showcased by a correlation coefficient value of 0.24 in Figure 3.8. Particularly, approximately 60% of requests originating from a foreground app are expected, whereas only $\approx$ 34% are expected from background apps, that is, 2 out of every 3 requests originating from background apps are unexpected. This ratio greatly varies between the different categories and permissions, were the values for the latter are illustrated in Figure 3.13. From this plot we observe that for any permission, it is more likely to be expected when requested from a foreground app than from a background app. Additionally, most requests for both SMS and CALL_LOG permissions come from the background, where in these situations, less than 20% are expected, while STORAGE, MICROPHONE and CAMERA were requested from the foreground over 80% of times, where the expectation ratio in these cases was over 60%. It should be noted that while visibility is important in user expectation, which in turn is strongly related to the grant result, the effect of the visibility in the grant result is low, as previously highlighted. The reason for this is that visibility has a 0.24 correlation coefficient with the expectation and the expectation has a correlation of 0.57 with the grant result, of a maximum of 1. Therefore, these values justify that the correlation of the visibility in user expectation is not transitive to the grant result.

Privacy expectations can be highly subjective due to the imperfect mental model (knowledge) that each individual has about the functionalities of apps [Lin et al.,

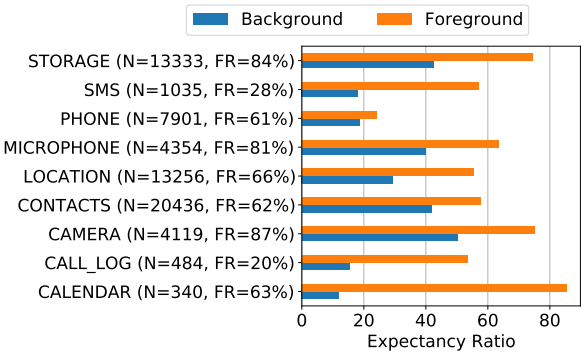


Figure 3.13.: Expected ratio per permission group and visibility of the requesting app. The "N" is the number of requests per group and "FR" the foreground ratio, that is, the percentage of requests that came from apps that were visible to the user at the time of the request.

2012]. An open question however, is how much influence does expectation have on privacy decisions. Our results showed than in general expected requests are allowed and unexpected requests are mostly denied. However, this can differ for each participant. Figure 3.14 presents the average grant result per expectation value, for each user. In this plot, users are represented by their ID on the x axis, while the respective average grant result is presented as a colored bar for each expectation value on the y axis. The color ranges from dark red, if the user denies all requests, to dark green if the user allows all. For example, the user with ID 22 mostly rejects requests independently of whether they are expected or not, while the user with ID 61 allows all expected and unknown requests, while denying all unexpected ones. From this plot we observe that the importance of expectation greatly varies with each individual. There are users whose privacy decisions are uncorrelated with their expectations. In the plot, these are the users with similar color bars for any expectation values, and we see examples of users that allow all (all green), deny all (all red) or allow or deny selectively (all orange/yellow). Then, there are people that deny most or all unexpected requests but allow most or all expected, as can be seen by the green bar in expected requests and red/orange in unexpected. These are individuals whose privacy decisions closely follow their expectations and correspond to the majority. Finally, there are participants in between the previous two extremes, which take into consideration the expectation as well as other variables, such as the visibility, the category or requested permission. In summary, while the importance of expectation in privacy decisions varies for each user, the majority acts in accordance with their expectations, as highlighted from the strong correlation in Figure 3.8 and now confirmed in Figure 3.14.

Figure 3.14.: Average grant result per expectation value and per user. Each user is represented by an id on the x label and the average grant result is represented as a color ranging from dark green, if the user allows all requests, to dark red, if the user denies all requests, for each of the three expectation values in the y axis.

3.3 Automated, Personalized and Context-Aware Privacy

The exploratory data analysis in the previous section has confirmed existing problems with current permission managers, and uncovered new findings. To summarize the main insights, the number of hourly resource accesses made by installed apps (an average of 35) proves the unfeasibility of asking the user on every use, and thus motivates the need for automation. In fact, almost 80% of our participants were surprised by the number of requests and only 23% of our participants expressed that the Android 9 permission manager was sufficient to effectively manage permissions. Moreover almost 50% of requests were unexpected by users, thus showing a strong misalignment between apps practices and user expectations. Our analysis further showed that the Android permission manager would incur in a violation of privacy in 15% of times. This is due to the fact that privacy decisions vary with changes in context and with personal preferences within each context, which the current permission manager is oblivious. These findings incite further research on smart automation of privacy preferences, particularly through personalization and context-awareness.

In this section we leverage the previous analysis and conclusions towards developing a personalized and context-aware permission manager. Specifically, we build on previous approaches on automated privacy enforcement by developing machine learning models that are able to automate privacy decisions [Olejnik et al., 2017; Liu et al., 2016]. However, we differentiate ourselves by considering and evaluating the impact of contextual features and user expectation both on the development of the privacy profiles, i.e., we develop context-aware profiles, and on the prediction of privacy decisions. Section 3.3.1 details and evaluates this endeavor. Finally, contrary to contextual features, such as the visibility of the requesting app or the network status, user expectation is not available in the prediction phase without asking for it to the user. As discussed, we seek to minimize user input. Therefore, in Section 3.3.4 we propose a two step approach, where we first estimate the expectation of the user, and then use such estimate to predict the privacy decision.

3.3.1 Predicting Privacy Decisions

The methodology for training a classifier to predict the grant result is as follows. From the collected permission decisions and respective contextual information, we normalize all and one-hot encode the categorical features, such as the request permission and category of the requesting app. We then start by analyzing the performance of a global predictor in Section 3.3.2, that is, a predictor that uses the input features to output the decision to allow or deny a request, while treating each user equally, i.e. without personalization. This evaluation is performed by first selecting the best predictor (model) and respective parameters through a cross-validated grid-search, followed by an evaluation of the best feature set to use in the prediction. We resort to the F1 score metric to compare the performance with previous works and to the Area Under the Receiving Operation

Curve (ROC AUC) as performance indicator, as the F1 score presented some misleading results as detailed in the referred section. The global predictor is then used as baseline comparison to the personalized predictors in Section 3.3.3, in where for the personalization we resort to the use of the privacy profiles as an additional feature in the prediction. However, one can consider different feature sets (tuples) for the creation of profiles and for predicting of the grant result with the profiles. Therefore, to evaluate the combination that leads to the best performance, Section 3.3.3 presents the results for all considered combinations of feature sets for the creation of the profiles and all considered feature sets for the prediction. All considered feature sets were based on their importance in the grant result as analyzed in the previous section. Finally, Section 3.3.3 further presents the privacy violations incurred by the best predictors, while contrasting them with the violation rate achieved by the Android permission manager with our dataset.

3.3.2 Global Prediction

Since there is no a priori best classifier to predict privacy decisions, we experimented using a grid-search with models from the literature. Specifically, Support Vector Machines (SVM) with linear [Liu et al., 2016; Olejnik et al., 2017; Wijesekera et al., 2018] and Radial Basis Function (RBF) kernels, decision trees [Olejnik et al., 2017], bagging, ada boosting, random forest and a neural network. Appendix D presents the parameters that were explored in the grid-search. The best results from the grid-search were similar between the different classifiers, therefore the choice of classifier is not important. Nevertheless, we picked the best performance: ada boost with a ROC AUC of 0.827 and a F1 score of 0.808, approximately. These results were achieved using 100 decision trees with a max depth of 1 as base classifiers and with a learning rate of 0.5. The following results use ada boost as classifier with the specified parameters. We also focus on the ROC AUC, as the F1 score was misleading. Specifically, using the mode as output resulted in an F1 score of 0.8 (close to the best performance) but in a ROC AUC of 0.5, which is the same value as a random classifier would achieve.

A 5-fold cross-validated feature forward selection by the ROC AUC with one-hot encoding of the features selects the expectation as the most important feature, followed by some permissions and categories. The performance plot as a function of the selected features is displayed in Figure 3.15. The visibility of the requesting app is selected as the seventh most important feature. However, the visibility is highly correlated with the expectation, as previously discussed, and thus, this cumulative forward approach fails to account for individual feature importance.

To better evaluate the importance of features, we have considered some feature set variants based on the analysis provided in Section 3.2.3.1 and cross-validated the performance of the classifier with each variant. Figure 3.16 presents the obtained performances, in where it is clear that the expectation is the most relevant feature. In fact, just using the expectation results in an F1 score and

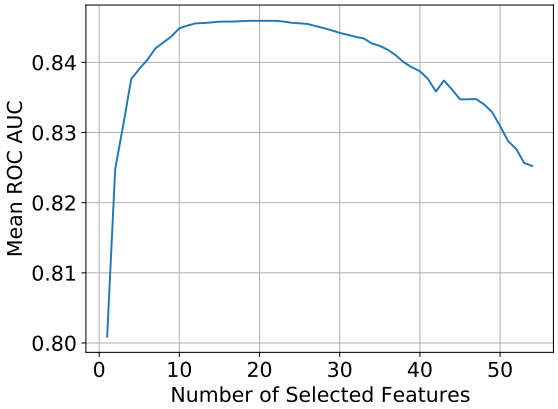


Figure 3.15.: 5-fold cross-validated feature forward selection, with all features in the dataset, after one-hot encoding.

ROC AUC of over 0.8. Adding the category and permission to the expectation, leads to the best ROC AUC (≈ 0.831), even slightly better than when using all features. Contextual features such as the [V]isibility, [L]ocation and [N]etwork status added very little or nothing to the category and permission (CP), as can be seen from the similarity of scores between using CP or any combination of V, L and N with CP. These results indicate a general lack of importance of the considered contextual features in the performance of the classifier. However, we believe that at least in part, this is due to the fact that the category of the requesting app and requested permission already encode part of the context as discussed in Section 3.2.3.1. Therefore, the additional information gain added by the contextual features is either not sufficient, or the classifier fails to account for them. Regardless, a ROC AUC of over 0.8 is already a good performance for a classifier that treats all users equally, that is, it fails to account for privacy's personal preferences. The next section enhances this approach by providing context-aware personalization.

3.3.3 Personalized Prediction

The previous prediction performance had no personalization, in the sense that a single classifier was trained with no feature that indicates personal preferences, thus treating each user equally. However, personalization has shown to increase performance [Liu et al., 2016; Wijesekera et al., 2017]. Therefore, in this section we build privacy profiles following a similar methodology from [Liu et al., 2016] in order to create personalized and automatic privacy decisions.

Traditionally, privacy profiles are build by applying hierarchical clustering to each user [Lin et al., 2014], where each user is represented as a tensor where each cell is the tendency to allow or deny requests for a particular pair of category-permission. However, our dataset contains additional features that capture the similarity between user behavior in a more fine-grained way. Specifically, instead of just using the pairs of category-permission (CP), we can additionally consider the expectation (E), to form expectation-aware profiles, or other contextual features such as the location (L) of the user, the visibility (V) of the requesting

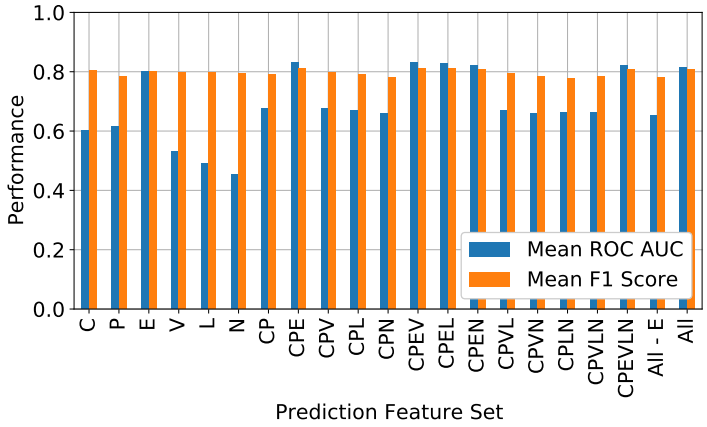


Figure 3.16.: 5-fold cross-validated performance of the ada boost classifier on the different considered dataset variants. Each variant is a combination of the following features, which are identified by their first letter: [E]xpectation, [C]ategory of the requesting app, [P]ermission requested, [V]isibility of the requesting app, [L]ocation, and [N]etwork status. "All" corresponds to using all features available in the dataset and "All - E" is all features except the expectation.

app and the network status (N) to form context-aware privacy profiles. Towards this end we consider the following feature variants for clustering: CP, CPV, CPE, CPL, CPN, CPVLN and CPEVLN, where each letter corresponds to a feature as previously described. Furthermore, regardless of how the profiles are formed, we can use any combination of features in the prediction alongside the profiles. Therefore, we performed all combinations of clustering with the feature variants displayed above, with the same feature variants in the predictions plus all features ("All") and all features except the expectation ("All-E"). For each combination of profiling and prediction, the number of profiles was varied from 1 to 9 and only the best results are displayed.

Figure 3.17 presents the obtained results, where the first observable result is that any profiling with any prediction approach outperforms not using profiles, thus confirming previous findings that personalization improves performance [Wijesekera et al., 2017; Liu et al., 2016]. Secondly, the best overall results are achieved by profiling only with the category and requested permission (CP). This is partially due to the fact that using more features in the profiling increases the amount of missing data that needs to be inputed, therefore potentially biasing the data. Nevertheless, profiling with CPE, that is, the tuple <category,permission,expectation>, followed by prediction with all features achieves a ROC AUC of 0.956 or prediction with CPE achieves a ROC AUC of 0.957, where this latter is the best performance. Similar results are achieved by profiling with CP and predicting only with CPE, a ROC AUC of 0.955, approximately. The advantage of this second best result is that less data is required, specially for assigning the privacy profiles, a step that requires asking questions to the user and therefore, should be minimized [Liu et al., 2016]. Finally, without the expectation, the best performance is achieved by clustering and predicting with CP, a ROC AUC of approximately 0.9.

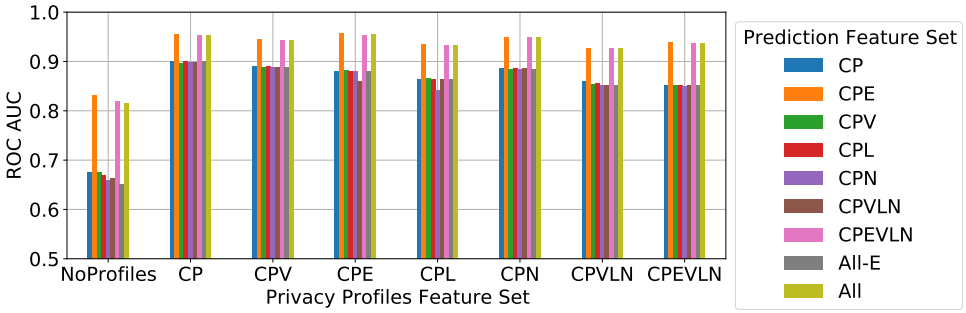


Figure 3.17.: 5-fold cross-validated performance with privacy profiles built with different feature sets, or no privacy profiles ("NoProfiles"), followed by prediction with several other feature sets. The number of profiles was varied from 1 to 9 and only the best result is displayed for each combination of inputs. Each feature set is identified by the combination of the following features identified by their first capitalized letter: [C]ategory, [P]ermission, [E]xpectation, [V]isibility, [L]ocation, and [N]etwork status. "All" and "All - E" corresponds respectively to using all features and all features except user expectation.

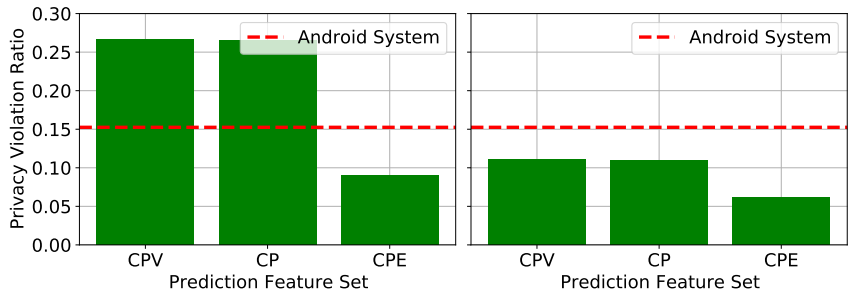
The previous results are comparable to the state of the art [Liu et al., 2016], whose reported F1 score was 0.9 with profiles built with the tuples <category,permission,purpose>. Our best F1 score is approximately 0.924, achieved through profiling and predicting with CPE, that is, with the expectation instead of the purpose. Without the expectation, our best F1 score is approximately 0.88, with profiles using only the pair category-permission and predicting with the category, permission, visibility, semantic location and network status (CPVLN). However, because the datasets are different, we cannot say that taking into consideration the expectation results in a better performance than using the purpose. A natural departure from this work is to combine both features.

An interesting, yet unexpected result that is also observable from Figure 3.17 is the rather low impact of the contextual features in the prediction. Specifically, if the expectation is not considered, using just the category and permission often results in the best performance, or very close to this value. This is partially explained by the correlation between the context of the user and the pair category-permission, as discussed in Section 3.2.3.1. However, we were expecting a stronger influence, particularly the visibility of the requesting app, which has been found to have a strong influence in privacy decisions [Wijesekera et al., 2015]. The reason for the low impact of the visibility of the requesting app is that users allow 68% of visible requests and 62% of background requests, as aforementioned. This difference might be irrelevant to the classifier. A potential reason for the low impact of the location is the fact that 84% of requests were with users at home, owed to COVID19 travel restrictions that were in place at the time of the campaigns. Due to this skewness, the importance of the location might be mis-measured. Therefore, we repeated the previous methodology while

subsampling the home requests to equal the number of work requests. The results with the subsampled data showed that without profiling, the location feature slightly increased the performance, but with profiling the results were similar to the ones obtained in Figure 3.17 and we therefore omit the plots. It is possible that these contextual features, specially the visibility, have a varying importance depending on the user as some users allow/deny everything regardless of any feature, while others are more selective. However, profiling with these features either failed to capture these personal preferences or the increase in the missing data deteriorated the results, as increasing the number of features in the profiling exponentially increases the amount of data that needs to be inputted for the hierarchical clustering. Towards validating the potential bias introduced by the inputted data, we build privacy profiles using the K-means clustering algorithm [Sanchez et al., 2020; Ravichandran et al., 2009] instead of hierarchical clustering, thus not requiring missing data inputation. The performances were worse in all cases, and thus, we omit such results.

Finally, we can compare the number of privacy violations that these approaches incur. Privacy violations are defined as permission requests that the user explicitly denied, but would otherwise be allowed. As previously mentioned, for the collected dataset, the Android 9 default permission manager based on runtime permissions would have violated the privacy in 15.25% of requests and would have incurred a median of 64 prompts to the user in a period of approximately a week. A personalized and automated prediction following the methodology above would require only a few questions to assign the profile [Liu et al., 2016] and it would result in 6.18% of privacy violations, a 59.5% reduction on Android permission manager, as displayed in Figure 3.18b, where the green bars present the violation ratio for the best personalized predictors and the dashed red line is the Android system violation ratio. Without the expectation, the lowest privacy violation ratio achieved is 11% when predicting with CP, which is still a reduction of 27.9% when compared to the standard Android permission manager. Looking at Figure 3.18a, it is interesting to note that automated solutions without privacy profiles, which correspond to the global predictors from Section 3.3.2, and without expectation, result in a higher amount of privacy violations than the Android system.

In summary, it is possible to automate privacy decisions with high performance, specially when taking into consideration user expectation. Contextual features seem to have a low impact in the performance of the prediction, which we mostly attribute to the fact that the pair category-permission already partially encode the context. Furthermore, the achieved prediction model can reduce the privacy violations in over 50% when compared to the current Android permission system based on runtime permissions. However, such system requires knowing the expectation of the user regarding every request, which we were unable to predict with sufficient accuracy and would therefore require user input, that should optimally be minimized. Without the expectation, it is possible to automate privacy decisions, while reducing the privacy violations by 27.9%. These results indicate that permission systems can still be enhanced, specially by taking the expectation of users into account.



(a) Global Prediction (no profiles). (b) Profiles using CP.

Figure 3.18.: 5-fold cross-validated privacy violation ratio of the best performant predictors for the global predictors 3.18a and the personalized predictors 3.18b. Each feature is a letter, where C is category, P is the requested permission, V the visibility of the requesting app and E is the expectation. The ratio of privacy violations that the Android permission manager would have incurred is presented as the red dashed horizontal line.

3.3.4 Predicting User Expectation

A natural improvement to the previous approach would be to predict user expectation such that we could either use it in the prediction without requiring user input or to devise an approach that only requests permissions when these are unexpected by the user. Towards this goal, we followed a similar approach to the one taken to predict privacy decisions in Section 3.3.1. Specifically, we experimented with the same combinations of data sets to first build the profiles, which in this case we referred to them as expectation profiles, and then train a classifier to predict the expectation with the profiles and every data set.

The best global classifier, that is, the non-personalized approach, for predicting user expectation was the Random Forest with a ROC AUC of 0.746. With the expectation profiles, the best achieved ROC AUC was 0.764, obtained with profiles built with the pair category-permission and predicting with the same pair of features plus the visibility of the requesting app. This result further confirms the importance of the visibility of the requesting app in the expectation of the user. Unfortunately, using this model's output as an estimation of the user expectation to then predict privacy decisions, resulted in a worse performance than predicting the decisions using only the pair category-permission.

One hypothesis for the low performance in predicting expectations is that it can be quite subjective and dynamically tied with the context, so it is difficult to capture even when using profiles. Additionally, it is possible that the expectancy of users varied throughout the campaign. To account for the potential variation of user expectation, we attempted to predict the expectation using adaptive learning, where the learning algorithms detect drifts in the classification error as to retrain the classifier with newest incoming data in response. Towards this end, we treated the permission request data as streaming data and fed it iteratively to an incremental classifier, that is a classifier that trains in an online fashion as more data is fed to it. To detect drifts in the classification performance, we used

two classical algorithms: the Drift Detection Method (DDM) [Gama et al., 2004] and the ADaptive WINdowing (ADWIN) [Bifet and Gavalda, 2007]. These two algorithms are similar as both compute statistics over the stream using dynamic windows to detect a drift. The major difference lies in ADWIN computing the averages between every pair of windows up until the current value, while DDM uses the average and standard deviation between an initial "training" window and all values of the stream until the current value. The threshold to detect the drift is statically defined in both cases.

Our results using adaptive learning with the DDM and ADWIN drift detection methods were slightly inferior than the global prediction counterpart: an average of 0.723 and 0.724 ROC AUC with the DDM and ADWIN, respectively, where the global predictor achieved a ROC AUC of 0.746. Therefore, we were unable to estimate user expectation with enough accuracy for it to be useful when predicting privacy decisions.

3.4 Limitations and Future Work

The limitations and future work remarks regarding this chapter can be fundamentally divided in two distinct topics: the data collection field study, and the mechanisms towards personalized and context-aware privacy enforcement. The following two sections detail each of these topics respectively.

3.4.1 Field Study

The data was collected in a set of campaigns spawning from July 2020 up to May 2021 in Portugal, as displayed in Figure 3.2. This period included periods of mandatory confinement and recommended remote work, thus limiting the data collected at each (semantic) location. In fact, both the collected context and app usage might differ from *normal* conditions, as both of these aspects are intertwined, as analyzed in Section 3.2.3. For instance, over 80% of requests were prompted with the participant at home. We should note that we balanced the dataset for each location and verified that the insights obtained in this work hold. We leave as future work any analysis of the impact of the COVID19 in the data.

Borrowing a campaign phone has the disadvantage of having to configure participant applications on the phone. To ease transition and favor using the campaign phone, we installed the participant's personal apps on the campaign phone before lending. The advantage, however, is that any person can participate in our experiments. Using personal phones would be possible, but CM-NPM requires administrator permissions (rooted Android device), which would reduce the experiment population and bias the dataset towards more tech-savvy participants. Unfortunately, due to the COVID19 related difficulties in recruiting participants, our dataset is still biased towards young adults with technical backgrounds. Therefore, collecting data from a more diverse population would improve the data quality. Furthermore, participants were still required to configure their accounts in each app. Due to the short duration of the campaigns,

some participants might have not configured all apps, potentially limiting the amount of data collected. Additionally, some sensitive apps, such as financial apps, might detect that the phone is rooted and refuse to run due to perceived security issues. We have analyzed the use of financial apps and noticed that 50% of the users that had financial apps on the personal phone did not use financial apps in the campaign phone through the campaign. In these cases, participants might have used both phones, thus reducing the amount of collected data.

To enhance the overall quality of the dataset and ecological validity of the findings we could have collected app usage from the personal phone, although doing so would require rooted personal phones due to Android's restrictions. We could have additionally implement opportunistic surveys to further analyze the reasoning behind the expectation and respective privacy choices. We leave these remarks as learnt lessons for future works.

Finally, a possible addition to our data collection would be to infer the purpose of the permission request, that is, the functionality of the app that requires the information that will be collected, should the permission be allowed. As mentioned in Section 2.3.2, inferring the purpose is challenging, specially in runtime [Van Kleek et al., 2017; Smullen et al., 2020]. However, the purpose has been highlighted as a key feature in privacy decisions [Liu et al., 2016] and could have therefore, complemented our expectation analysis. In fact, it has previously showed how diverging personal expectations can be depending on the knowledge about app functionality [Lin et al., 2012].

3.4.2 Personalized and Context-Aware Privacy

In Section 3.3.4 we report our attempts to predicting user expectation, whose results were insufficient towards improving the prediction of privacy decisions without expectations. This is an indicator that the expectation can be more personal and dynamic than the respective privacy decisions. An interesting unexplored venue is to use more complex learning paradigms, such as deep learning, that can potentially find more convoluted and/or more subtile relationships between the features.

An additional enhancement could be to develop or integrate a context modeling/inference mechanism. With such approach, instead of using raw contextual features, we could have a more rich and semantic description of the user context that could potentially improve the context-awareness of the privacy mechanism. Some existing approaches and respective limitations have been discussed in Section 2.3.2. A potential arising challenge in this integration is the evaluation of which contextual data type is actually relevant towards privacy decisions. For instance, it might be irrelevant whether the user is cycling or running, but it is relevant to know that the user is traveling, as identified in Section 3.2.

Collecting data regarding permission decisions and application usage towards building automated privacy mechanisms raises privacy issues. Our approach was centralized, in the sense that a single entity (in this case us) collected all data, including sensitive values such as the user location or devices in vicinity.

This requires users to trust the entity that builds these models. A different approach is to use privacy-preserving distributed learning, in where the models can be trained without revealing sensitive data even to the entity training the models. In a master thesis supervised supervised in the scope of this work, we have proposed a method to use privacy-preserving K-means clustering to build the profiles, followed by federated learning towards training the prediction models. Our proposal can be found in [Brandão et al., 2022]. In that line of work, a natural departure is to develop a monitoring framework towards iterative and online adaptation of the models and profiles as to continually improve and further personalize the prediction of privacy decisions. The challenge however is that such framework should also preserve the privacy of the users.

Finally, privacy decisions in permission managers are generally limited to either allowing or denying access. This binary decision provides limited control over the privacy and utility trade-off. The incorporation of obfuscation in permission managers is an interesting, yet challenging venue, due to the heterogeneity of data types and corresponding specific obfuscation techniques [Cunha et al., 2021]. However, these techniques could be integrated in permission managers for each of the sensitive data types to allow for finer-grained control over the referred trade-off. An additional challenge that advents from this integration is to have privacy-preserving mechanisms that adapt to varying contexts. The following chapters delve into this subject focusing on location data, a particularly relevant data type in the context of mobile devices.

3.5 Chapter Summary

The ever increasing complexity of smart and mobile devices require automation for effective privacy protection. Replacing the install-time permission model, in where all permissions were allowed at all times, the runtime permission brought fine-grained control and in-context permission requests, thus being positively received by users. However, after being allowed once, requests are in general, successively allowed without user interaction or even awareness. This automation steams from the necessity to reduce the user input as to avoid warning fatigue and or desensitization. Regrettably, the context at which successive requests occur can vastly differ from the context at the time of the first allowed request. These later accesses thus violate contextual integrity, i.e., they violate the privacy preferences and expectations of the user under the involving context.

Automated privacy mechanisms should ideally take into consideration personal privacy preferences and expectations under each surrounding context, such that contextual integrity is upheld. This chapter reports our efforts towards such main goal. We started by performing a field study with 93 participants to collect permission decisions, the surrounding context and the expectation of the user regarding each request at runtime. This is the first dataset to collect user expectations in-situ, and, therefore, we made it available to the research community.

From the campaigns questionnaire, less than 23% of our participants reported the default Android permission manager as being sufficient for managing permissions, while almost 80% were highly surprised by the number of requests issued by apps. In fact, the collected data reveals a strong misalignment between apps practices and user expectations, as almost half of requests are unexpected. Furthermore, the default Android system would have violated user privacy in 15% of requests, that is, it would have allowed requests that our participants explicitly denied. These results serve as motivation towards the development of better privacy mechanisms, and therefore motivate this work.

Ours analysis on the collected data reveals that the visibility of the requesting app, the location of the user and the network status are important contextual cues that partially explain the variability of the grant result, i.e., the user decision to allow or deny a permission. In addition, we find that the category of the requesting app and the requested permission moderately encode user context, as different apps are used under different contexts. Notwithstanding, privacy decisions see the strongest correlation with user expectations. In particular, 9 out of 10 expected requests are allowed, while less than 40% of unexpected requests are allowed. These ratios highlight the importance of explaining app requirements to the user. However, both the expectation and the importance of the expectation in the decision are highly personal and context dependent.

Leveraging on the relations between privacy decisions, user expectations and the context, we develop automated, personalized and context-aware permission mechanisms for prediction of the grant result. Our results show that by taking into account the expectation of the user, one can reduce the number of privacy violations by over 50% when compared to the Android 9 permission manager based on runtime permissions. Without user expectation, it is still possible to reduce the privacy violations by approximately 28%.

The automated solution presented in this chapter works with permission requests for any type of data and at data collection (i.e. before the data leaves the mobile device), thus preserving privacy even against untrustworthy providers. However, it gives limited control over the trade-off between privacy and utility. Particularly, allowing a permission request corresponds to having full utility, but zero privacy. In contrast, denying a request corresponds to having maximum privacy, but no utility – assuming that the application does require the requested permission for a given functionally, which should (but unfortunately is not) true in all cases. This model prohibits users from using any functionality without losing the maximum privacy and is therefore sub-optimal. Thus, a natural departure from this work, as highlighted in Section 3.4, is to incorporate privacy-preserving mechanisms that retain a certain level of privacy, while allowing the use of a (degraded) service. Regrettably, this type of mechanisms are data type dependent [Cunha et al., 2021].

The following chapters focus on location data, a data type that is particularly relevant in the context of mobile devices. Chapter 4 evaluates the privacy impact of the frequency of location reports sent to a provider. In the use-case of the smartphone, multiple apps can access the location at different times/contexts, or

the same app might have different frequency of reports – for example, retrieving the closest restaurant might require a single position, while navigating to said restaurant requires continuously sharing the location. The privacy loss in each of these cases varies and therefore, the privacy mechanism must adapt to each situation/context, ideally automatically. Therefore, in Chapter 5 we propose a novel Location Privacy-Preserving Mechanism (LPPM) that automatically adjusts privacy and utility as a function of the frequency of location reports and the user velocity, thus adapting for example, to varying application usage (device context), methods of transport and road typologies (user context). This adaptability is paramount for the integration of such solution in a permission manager, an endeavor that, due to time constraints, is left as future work.

Chapter 4.

Impact of the Frequency of Reports on the Privacy Level of Location Traces

Contents

Current permission managers implement binary access policies: the user either allows or denies apps' access to the resources. Allowing a permission corresponds to having no privacy and maximum utility, as the original data is sent to a potential untrustworthy service provider, while denying corresponds to having maximum privacy and no utility, as the functionality is lost. Towards improving the status quo, obfuscation techniques can be used in permission managers to allow for fine-grained balance between privacy and utility in an online fashion, that is, before the data is sent to the collecting entity. However, and as identified in Section 2.2.3, obfuscation is data type dependent. Therefore, this and the following chapter focus on location data, a prominent type of data in the context of mobile devices [Huang et al., 2018].

Mobile devices and ubiquitous connectivity fostered services that take into consideration users' contextual information. One emergent category of these services is the Location-Based Services (LBSs), in which users share their location to obtain geographically and temporally related information (e.g. finding the nearest open restaurant). While beneficial to the user, sharing location data poses a threat to privacy that goes beyond physical safety. In fact, visited locations can reveal users' identity, habits, addictions, health conditions and even social connections [Krumm, 2009; Gambs et al., 2010].

Untrustworthy LBS providers, that may share or publish the data, passive eavesdroppers and security breaches can cause disclosure of location data thus putting at risk the privacy of its users. Preserving privacy against this range of attack vectors requires Location Privacy-Preserving Mechanisms (LPPMs) at collection time, i.e. mechanisms that run in-device in an online scenario [Mendes and Vilela, 2017]. LPPMs report an obfuscated version of the exact user location as to preserve a certain level of privacy at the expense of a degraded quality of service.

Geo-indistinguishability [Andrés et al., 2013], a formal notion based on differential privacy [Dwork, 2008] has seen increasing research interest due to its simplicity of implementation, efficiency and effectiveness [Liu et al., 2018a; Chatzikokolakis et al., 2017; Hsu et al., 2014]. Geo-indistinguishability guarantees that any two points within a given radius around the user are statistically indistinguishable independently of an adversary's background information. Specifically, the reported (obfuscated) point is generated with (almost) the same probability for any point within this circle, consequently concealing the exact location of the user.

Depending on the LBS, location data can be reported either continuously or rather sporadically [Shokri et al., 2011; Shokri et al., 2011]. This frequency of reports directly impacts the temporal correlation between subsequent reports which in turn can be used by an adversary to track users over time and even predict future locations [Liu et al., 2018a; Krumm, 2009; Xiao and Xiong, 2015]. While geo-indistinguishability bounds the amount of disclosure, it considers reports to be independent between each other. In fact, in the context

of sporadic release of data this consideration has been assumed when designing LPPMs [Shokri et al., 2011; Oya et al., 2019]. However, there is no formal nor quantitative distinction between sporadic and continuous reports and thus, the distinction is often based on the type of LBS application [Shokri et al., 2011]. In this chapter we argue that the consideration of independence depends on the frequency of reports, even in the context of sporadic reports. Therefore and to evaluate our premise, we quantitatively study the impact of the frequency of reports on the achieved privacy level through geo-indistinguishability. The contributions of this chapter are as follows.

- We evaluate the effect of the frequency of reports in the privacy level of the Planar Laplace [Chatzikokolakis et al., 2017], a geo-indistinguishable LPPM, using state-of-the-art localization attacks and a tracking attack on real datasets. The variation of the frequency of reports is made such that typical values for both continuous and sporadic are considered as well as values in between both ends. Results showed that the privacy level when considering localization attacks is roughly constant over the range of tested frequencies of reports, while the effectiveness of tracking attacks decays as the frequencies of reports lowers. These results suggest that the consideration of independence between reports can be effectively assumed in the sporadic scenario.
- We evaluate the effectiveness of several values of ϵ, the privacy budget, in the privacy level of the Planar Laplace against the state-of-the-art localization attacks. The choice of a privacy budget in differential privacy, and consequently based approaches such as geo-indistinguishability, is still an open problem as it strongly depends on the application [Hsu et al., 2014]. In fact, it has been discussed that the definition of ϵ in geo-indistinguishability may be misleading in terms of the privacy level [Oya et al., 2017b]. In contrast with [Oya et al., 2019], our results showed that the relation between the average quality loss and average adversary error is only linear after a non-negligible threshold. That is, there exists an upper bound on the value of the privacy budget necessary to guarantee relevant privacy protection, which in our setting was $\epsilon = 4$ km^{-1}.
- We assess the effects of the grid resolution, i.e., the size of cells in the space discretization required by the optimal attacks, in the effectiveness of the implemented localization attacks. These results show a linear correlation between the cell width and the average adversary error, and thus suggest that a powerful adversary (with infinite computational power) could potentially defeat obfuscation. However, increasing the obfuscation (by decreasing ϵ) decreases the slope of the linear correlation. Consequently, by increasing the obfuscation, a higher decrease in cell width, and consequently an increase in computational complexity, is required for the same reduction in the average adversary error.

A previous work [Mendes and Vilela, 2018] has shown that the correlation between subsequent reports can be explored by an adversary using simple regression models as estimators. From such results it was concluded that not only does

the frequency of reports greatly impacts the temporal correlation but also that the estimation function affects the results significantly. However, the privacy level evaluation in that work was limited due to the use of simple regressions as attacks. This work greatly expands those results by providing a quantitative privacy evaluation with state-of-the-art attacks under both continuous and sporadic release of location data. While map-aware LPPMs have been proposed in the literature (e.g. [Liu et al., 2017]) and map knowledge has been used to reduce obfuscation areas (e.g. [Krumm, 2007]), to the best of our knowledge, we are the first to consider road network map-matching as a tracking attack.

The remainder of this chapter is structured as follows. Section 4.1 describes the empirical methodology whose results are displayed and discussed in Section 4.2. Section 4.3 presents the limitations and future work remarks, and Section 4.4 concludes this chapter.

4.1 Methodology

The main objective of this work is to evaluate the impact of the frequency of location reports on the privacy level of a Geo-indistinguishable LPPM, namely the Planar Laplace (PL) [Chatzikokolakis et al., 2017] described in Section 2.4.2.1. Towards this goal, we obfuscate the location reports using the PL mechanism to several sub-samples of real datasets, where each sub-sample corresponds to a different frequency of reports. Subsequently, we apply state-of-the-art localization attacks as to measure the privacy level obtained through the PL mechanism against possible adversaries. The following sections will describe the datasets used in this work and detail the carried out methodology.

4.1.1 Datasets Characterization

To evaluate the impact of frequency one must consider both continuous and sporadic release of data. As previously mentioned there is no formal nor quantitative boundary for the frequency of reports that defines what intervals belong to the continuous or sporadic scenarios. In fact, this distinction is made based on the type of LBS application [Shokri et al., 2011]. Therefore, and to allow for tuning the frequency of reports from highly frequent to "sporadic" reports, we selected three highly continuous datasets: the **Cabspotting** [Piorkowski et al., 2009] and **Portocabs** [Moreira-Matias et al., 2013] datasets, which are composed of taxi trajectories from the city of San Fracisco, USA, and Porto, Portugal, respectively; and the **Geolife** dataset [Zheng et al., 2009], a dataset of GPS data captured by handheld devices.

The Cabspotting dataset [Piorkowski et al., 2009] contains trajectories from over 500 taxis navigating in San Francisco Bay Area in a period of 30 days. It contains not only geo-location collected through a GPS at an average rate of 10 seconds, but also whether the cab is occupied or not. The Portocabs dataset is composed of trajectories belonging to 441 taxis in the city of Porto, Portugal, collected over a full year (from 2013/07/01 to 2014/06/30) with a sampling rate

of 15 seconds [Moreira-Matias et al., 2013]. The Geolife dataset [Zheng et al., 2009] is a well known repository of GPS traces collected from 182 worldwide users in the period from April 2007 to August 2012. It contains a total of 18670 trajectories reflecting the movements under a variety of transportation means, where 91% of these have a sampling rate of 1 to 5 seconds or 5 to 10 meters per point. The majority of the trajectories lie in Beijing.

While the datasets of taxi mobility are highly continuous, these movements often have a limited timespan. In fact, most of these trajectories present a timespan under 1 hour. On the other hand, [Mendes and Vilela, 2018] shows that the Geolife dataset contains a significant amount of time-gaps between reports, that is, discontinuities in the frequency of reports. Furthermore, since our tracking attack is a road network map-matching technique [Jagadeesh and Srikanthan, 2017], only vehicular trajectories can be considered. Consequently, we use the Cabspotting and Portocabs datasets to evaluate highly continuous reports and the Geolife dataset in a more sporadic scenario. It should be noted that while the Geolife is not a sporadic dataset, the continuity of reports allows to fine-tune the frequency of reports by periodically suppressing points to cover the full spectrum. Intuitively, this subsampling can be perceived as users in their quotidian trajectories making sporadic accesses to a LBS.

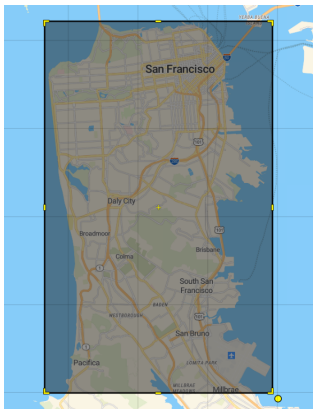
Our pre-processing for each dataset is as follows:

Geolife since in a sporadic scenario there are no trajectories, we first append all traces of each user as a single array of locations and subsequently sort by date. We then filter out locations that fall outside a bounding box containing the 5^{th} ring road of Beijing as illustrated in Figure 4.1a. This filtering reduces the space of possible user locations ($\mathcal{X}$), which in turn allows for a finer grid for the localization attacks. A total of 65.4% of points belonging to 179 of the 182 initial users remained after this pre-processing.

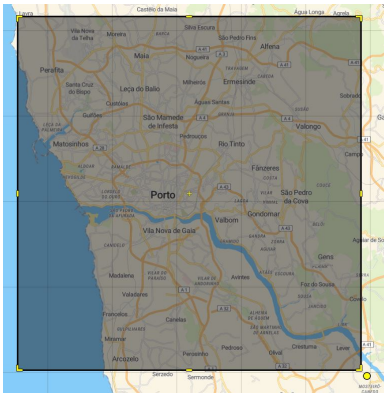
Cabspotting we first limit the trajectories to a bounding box within the San Francisco peninsula as specified in Figure 4.1b. Then we consider only trajectories with passenger as to remove cases where the taxi is stopped waiting for a client. Finally, we select trajectories with a duration of at least one hour, with intervals between reports of at most (approximately) 2 minutes as to avoid temporal discontinuities between reports. After this pre-processing, 85 trajectories remained.

After manual inspection of some of these trajectories in the map we were able to observe that the dataset contains noisy readings. For example, some GPS locations are reported in the ocean instead of in a bridge. Thus, to improve the original (noisy) data so as to build our ground-truth, we apply the MM technique described in section 2.4.3.3 to the original dataset. This way, we obtain a set of locations in the road network that serves as our ground-truth to compare against the locations after obfuscation and being subject to adversary attacks, as illustrated in the diagram of Figure 4.2. For that, we use the parameters from [Goh et al., 2012], which uses GPS data and is the work that served as baseline to the development

(a) Bounding box over 5^{th} ring road of Beijing used for the Geolife dataset. Approximately defined from South and North by the latitudes 39.753, 40.026, and from West and East by longitudes 116.199, 116.547.



(b) Bounding box over the peninsula of San Francisco used for the Cabspotting dataset. Approximately defined from South and North by the latitudes 37.600, 37.811, and from West and East by longitudes −122.517, −122.354.



(c) Bounding box over the city of Porto, Portugal, used for the Portocabs dataset. Approximately defined from South and North by the latitudes 41.0524, 41.257, and from West and East by longitudes −8.727, −8.456.

Figure 4.1.: Bounding boxes used in this work for each of the three datasets.

of [Jagadeesh and Srikanthan, 2017]. In [Goh et al., 2012] the estimated standard deviation was $\sigma = 6.86$m and they limited the potential locations $s_{i,k}$ to a circular radius of 50m from o_i. This discards candidate locations with low emission probability (c.f. equation (2.16)) and speeds up the map-matching process. For the remaining parameters we used the original values from [Jagadeesh and Srikanthan, 2017]: $\lambda_y = 0.69$ and $\lambda_z = 13.35$. The restriction of the 50m radius around o_i produced observations without candidate points in some trajectories due to both the considered road network (explained in the following section) and to the noisy dataset. For these observations, we considered the nearest road network node as candidate. Furthermore, after manual inspection of the 85 trajectories, we observed that in some the taxi stays roughly in the same place to which we attribute to heavy traffic. Consequently, we removed those trajectories

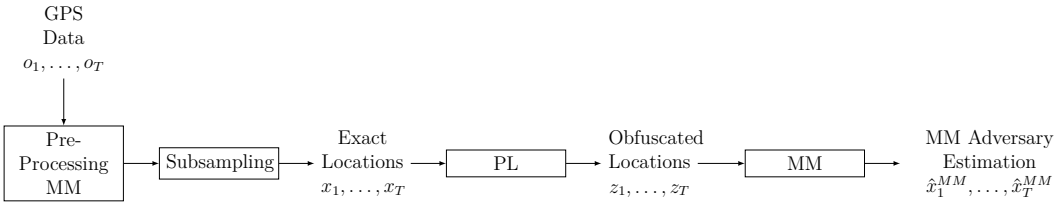


Figure 4.2.: Diagram of the methodology conducted for the Map-Matching attack.

and ran our tests for the 63 remaining trajectories.

Portocabs following a similar procedure to the one taken for the Cabspotting dataset, we limit the selected trajectories to a bounding-box containing the city of Porto, as illustrated in Figure 4.1c. From these trajectories, we select only the ones that present no missing data, that is, there is a location report every 15 seconds. Finally, we select the trajectories with a duration of 1 hour and 1 hour and 15 seconds, as to increase the number of trajectories. This resulted in 134 trajectories, which after some manual inspection as performed for the Cabspotting dataset, reduced to 123. To these final trajectories, we perform the same procedure as in the Cabspotting to obtain the ground-truth data.

The Geolife, Cabspotting and Portocabs datasets can be found in [Zheng et al., 2012], [Piorkowski et al., 2009] and [Moreira-Matias et al., 2015], respectively.

4.1.2 Experimental Setup

The methodology for the experiments consists in sub-sampling the datasets, applying the Planar Laplace mechanism described in Section 2.4.2.1 and subsequently apply the localization and tracking attacks from Section 2.4.3. As explained in the dataset characterization, the Cabspotting and Portocabs datasets are more suitable for the map-matching attack due to being highly continuous, present no temporal discontinuities between reports and for being vehicular trajectories. Consequently, we only apply the localization attacks to the Geolife dataset, while executing both localization and map-matching attacks to the Cabspotting dataset. We use the Portocabs dataset to further validate the map-matching results.

4.1.2.1 Subsampling

To vary the frequency of reports we subsample the datasets by suppressing reports such that the interval between consecutive points is at least Δ_t. To contemplate both continuous and sporadic scenarios, several values of Δ_t are considered. For the Cabspotting and Portocabs datasets, we set Δ_t = [60, 120, 180, 240, 300, 360, 420, 480, 540, 600] seconds as our highly continuous reports. Note that in the context of map-matching, the previous values of Δ_t are already considered low sampling rate [Newson and Krumm, 2009; Kubicka et al., 2018].

For the Geolife dataset we consider a larger range of frequencies and thus set $\Delta_t = [480, 540, 600, 1800, 5400, 16200, 48600, 145800, 437400, 1312200]$ seconds. This interval goes from 8 minutes up to 15 days, and thus is comprehensive enough to encompass both continuous and sporadic scenarios.

4.1.2.2 LPPM

To each dataset subsample we apply the Planar Laplace described in Section 2.4.2.1 under multiple values of ϵ. Since map-matching is computationally expensive, we have used fewer ϵ values for the Cabspotting dataset. Specifically, for the Cabspotting and Portocabs datasets we have used $\epsilon = [16, 32, 64, 128]$ km^{-1} and for the Geolife dataset
$\epsilon = [1, 1.5, 2, 3, 4, 8, 12, 16, 24, 32, 48, 64]$ km^{-1}. The average quality loss is measured using equation (2.4).

4.1.2.3 Localization Attacks

Following the methodology from previous literature [Oya et al., 2019; Chatzikokolakis et al., 2017], we use part of the dataset for training and the remainder for testing. Thus, and as described in Section 2.4.3, we consider three types of attacks: **optHW**, the optimal attack using the training dataset to build the mobility profile π^{train}; **omniHW**, the optimal attack using the test dataset to build the mobility profile π^{test}, which corresponds to an omniscient adversary; and **PEBA** as described in Section 2.4.3.2 and using the parameters from its original work [Oya et al., 2019], with $\pi^{avg} = \pi^{train}$. The adversary error defined in equation (2.3) is used to measure the privacy level against these attacks.

The considered localization attacks assume the space of exact user locations $\mathcal{X}$ to be discrete. Therefore, and similarly to previous works [Chatzikokolakis et al., 2017; Shokri et al., 2011; Murakami, 2017], we have discretized the space for both datasets in a grid of equally spaced cells, where the center of the cell corresponds to a locationstamp that is common to any GPS observation within the cell. For the Geolife dataset, the 5^{th} ring road of Beijing was partitioned in cells of 2000×2000 meters for a total of 17×16 cells. For the Cabspotting dataset, and for a fair comparison between MM and the localization attacks we measure the adversary error not as the distance from the estimation $\hat{x}_i$ to the center of the grid x_i (as in equation (2.3)), but instead from $\hat{x}_i$ to the ground-truth point, as the tracking attack would naturally consider it. Therefore, we also evaluate the effect of the grid resolution in the adversary error. This evaluation is done for the Cabspotting dataset using the subsample corresponding to $\Delta_t = 300$s, to decrease execution time, and several grid sizes composed by squared cells of [80, 90, 100, 125, 150, 175, 200, 250] m.

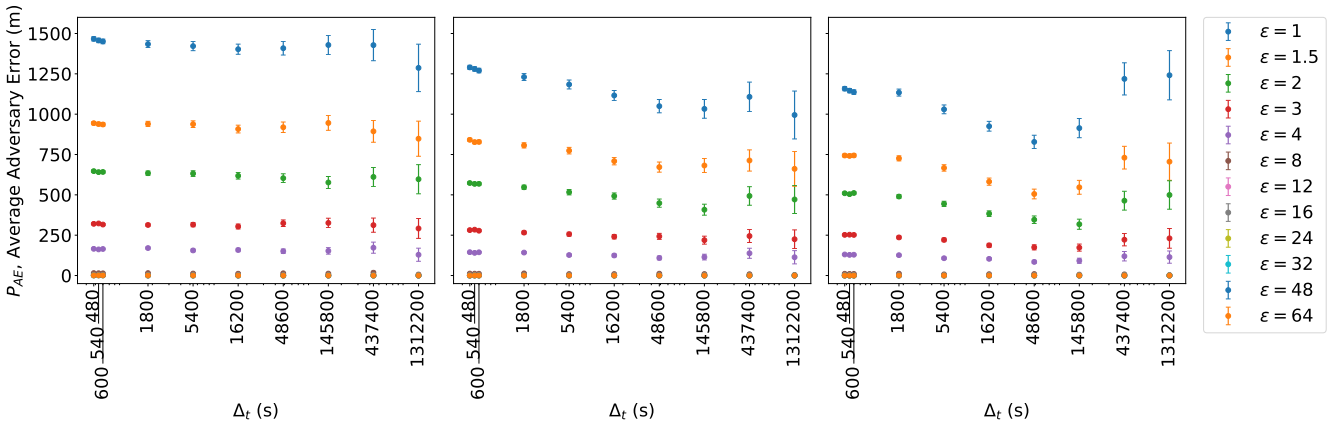
The selection of the train/test data partition for the Geolife dataset was done as follows. We select the users with at least 20 points for $\Delta_t = 1312200$ seconds, our highest Δ_t. The test data for each Δ_t is then the locations of these selected users. Using these users ensures that the training data does not contain data pertaining the victims of the attacks, the same users are present in all subsamples of the

dataset, and that enough test data is present to allow for profile tuning in the PEBA attack, even for the sparsest subsample (highest Δ_t value). The training data corresponds to using the locations of all users that were not selected as testing data for $\Delta_t = 480$ seconds, the lowest Δ_t. That is, the training data is the same for all Δ_t values. This avoids having poorer results for higher Δ_t due to the sparseness of the dataset. For the same reason, in the OmniHW attack the mobility profile π^{test} is also constant for all values of Δ_t and is built with the testing data with $\Delta_t = 480$. The mobility profiles π^{train} and π^{test} are therefore built using respectively 73.4% and 26.6% of the $\Delta_t = 480$ subsampled dataset.

For the Cabspotting dataset we use the 63 trajectories as test set and all remaining trajectories contained within the bounding box from Figure 4.1b as training data. To be precise, we use 905255 trajectories as training data. However, it should be noted that, contrary to training a classifier, using all this data as training data does not lead to overfitting. In fact, this corresponds to an adversary which has a very precise statistic model of the average mobility profile, or in other words, a model of how a "normal" individual moves in this area.

4.1.2.4 Map-Matching

The diagram from Figure 4.2 illustrates the methodology taken when using the MM technique. The "Pre-Processing MM" computes a ground-truth from the noisy dataset as explained in Section 4.1.1 to which is then applied the subsampling considering the aforementioned values of Δ_t. To the subsampled locations is applied the Planar Laplace (PL) using the described values of ϵ to obtain the obfuscated reports. Finally, MM is executed on the obfuscated locations to obtain the adversary's estimations. To assess the privacy level, we compare the ground-truth against the adversary estimations using the adversary error from equation (2.3) and the F_1 score from equation (2.21). The parameters σ, λ_y and λ_z for the MM attack were estimated following the original proposal [Jagadeesh and Srikanthan, 2017]. For the Cabspotting data we used trajectories within the bounding-box from Figure 4.1b with duration between 1 and 5 minutes with at least 2km of travelled distance (a total of 6003 trajectories). Equivalently, for the Portocabs dataset we selected trajectories within the bounding-box from Figure 4.1c with a duration of 5 minutes and with at least 2.5 km travelled distance, resulting in 4598 trajectories. For efficiency, and similarly to [Goh et al., 2012], we only consider candidates points within a radius r which we calculate using the inverse cumulative distribution function of the Gaussian distribution. The radius r is computed such that the circle centered at the observation contains the exact location with 90% probability. When this circle contains no candidates, which can happen due to the use of the LPPM and selected road network, the nearest road network node is used as candidate. The road network was obtained from OpenStreetMap using the OSMnx tool [Boeing, 2017] over the area defined by the respective bounding boxes.



(a) optHW (b) omniHW (c) PEBA

Figure 4.3.: Geolife average adversary error and respective 95% confidence intervals per Δ_t for all values of ϵ for the three localization attacks. The x axis is logarithmic and the y axis and legend are shared between the three plots.

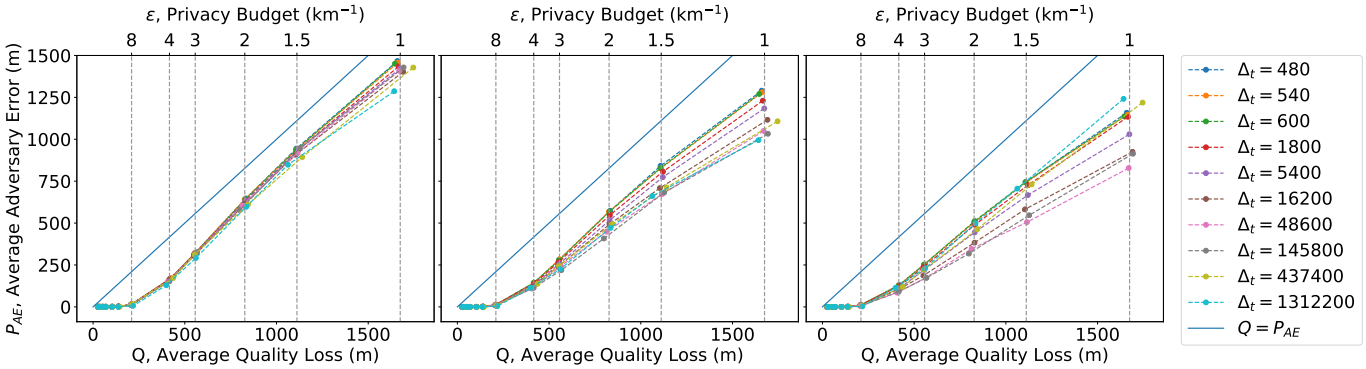
4.2 Results

This section details the obtained results. A separation based on the dataset is made, such that Section 4.2.1 details the results using the Geolife dataset, which focuses the sporadic scenario, and Section 4.2.2 describes the results using the Cabspotting and Portocabs datasets, the continuous case.

4.2.1 Geolife Results

For the Geolife dataset, only the localization attacks were executed. Figure 4.3 shows the average adversary error per Δ_t for all ϵ values and for each of the three attacks. The first thing we can observe is that the adversary error is roughly similar for any Δ_t. This allows to conclude that the frequency of reports has no significant impact on the privacy level. This is to be expected since in contrast with the tracking attack, the selected localization attacks do not take into account the temporal correlation. Consequently, the consideration of independence between reports is valid for the sporadic case. We note that while there are localization attacks which take into account the correlation between reports, such as [Murakami and Watanabe, 2016], and thus our results with such attacks could differ, the reported performance in [Murakami and Watanabe, 2016] is significantly lower to the attacks we consider.

Figure 4.3 also shows that omniHW performed better than the optHW attack, which was to be expected as the test mobility profile is used in the former. At the same time, the PEBA attack was even better than the omniHW for most values of Δ_t, thus confirming the results of the original work [Oya et al., 2019]. For the two highest values of Δ_t this was not the case, which we justify with the fact that not enough test data was present for PEBA to learn the mobility profile. Consequently, the PEBA results for these higher Δ_t are closer to the



(a) optHW (b) omniHW (c) PEBA

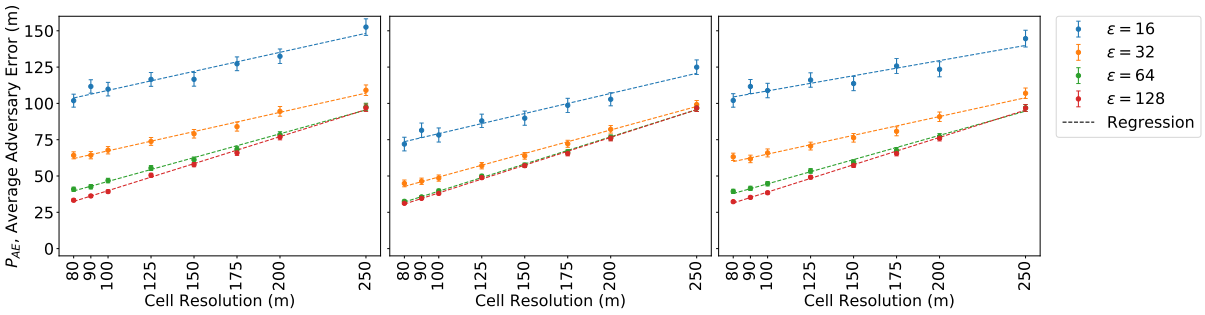
Figure 4.4.: Geolife privacy versus utility for all values of Δ_t for the three localization attacks. Each color represents a Δ_t value, where the points are the pair (P_{AE}, Q), which is obtained for a particular value of ϵ. Dashed vertical lines indicate the epsilon at the empirical quality loss averaged over all values of Δ_t. The solid line represents an adversary using the report as the estimation, for reference. The y axis and legend are shared between the three plots.

results of the optHW, which is in accordance with equation (2.13).

The last observation from Figure 4.3 is the amount of values of privacy budget (ϵ) that resolve in near zero average adversary error. Only the lowest 5 of the 12 experimented values of ϵ produced a non-negligible adversary error. For the setup we considered, values of $\epsilon \geq 8$ km^{-1} lead to basically no privacy protection. Our results indicate that for this setup a maximum value of $\epsilon = 4$ km^{-1} is needed for relevant privacy protection. As future work we intend to formulate a relation between the effectiveness of the optimal attack (measured by the adversary error) and the value of ϵ.

The last results for the Geolife dataset are displayed in Figure 4.4. These results show the average adversary error P_{AE} as a function of the average quality loss Q, which corresponds to the performance of an LPPM , for all values of Δ_t. Each color represents a Δ_t value, where the points are the pair (P_{AE}, Q), which is obtained for a particular ϵ. The dashed dark lines illustrate average quality loss averaged over all values of Δ_t for each specific ϵ. The results obtained in [Oya et al., 2019] showed that the relation between P_{AE} and Q is highly linear. Looking at Figure 4.4, we observe this to be the case only when $P_{AE} > 0$, which as we have seen from Figure 4.3 occurs for $\epsilon < 8$ km^{-1}.

The second result observable from Figure 4.4 is the similarity of the curve for the different Δ_t, which proves again that the frequency of reports has no major effect on the privacy level using these localization attacks. In fact, it is not possible to identify a specific Δ_t that has highest average adversary error for all values of ϵ.



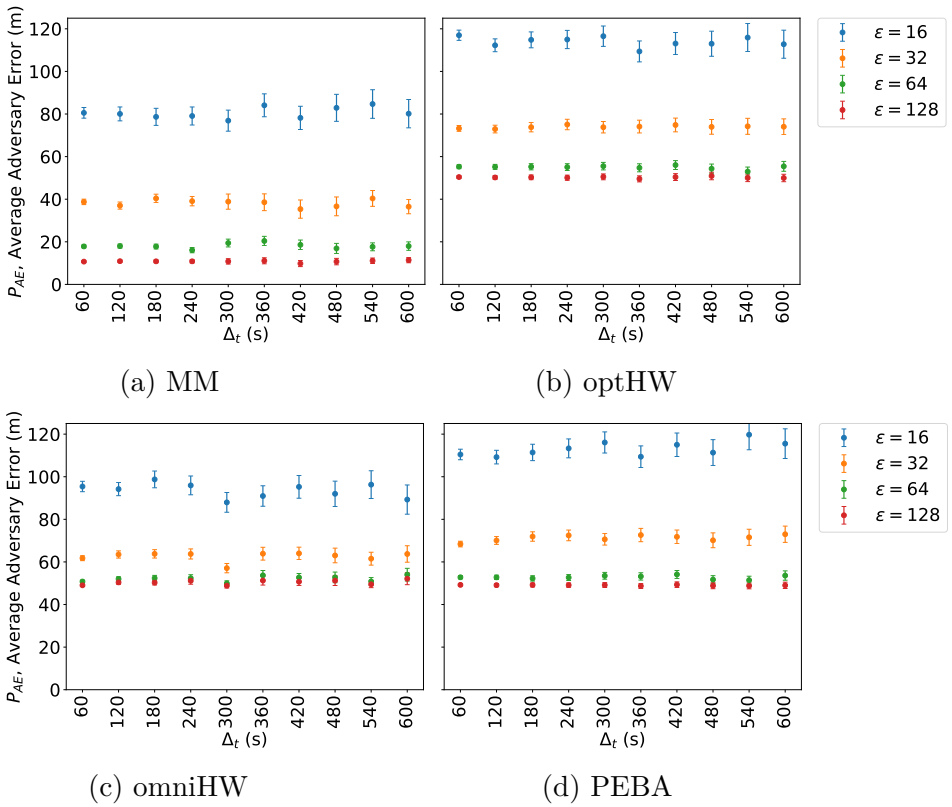
(a) optHW (b) omniHW (c) PEBA

Figure 4.5.: Effect of the grid resolution on the average adversary error (and respective confidence intervals) for each localization attack using the Cabspotting dataset with $\Delta_t = 300$ (to decrease execution time).

4.2.2 Cabspotting and Portocabs Results

The Cabspotting dataset is employed to assess the effect of attacks (both localization attacks as well as MM) on the continuous scenario. Since the effectiveness of the localization attacks is highly dependent on the grid resolution employed, we start by evaluating the effect of the cell size on localization attacks, as depicted in Figure 4.5. We can observe that for any epsilon and for any attack, there is a linear correlation between the cell resolution and the adversary error. As the cells get smaller, so does the average adversary error. Given these results, a resourceful adversary can potentially defeat obfuscation by using a very small cell resolution. However, it should be noted that as the privacy budget ϵ decreases (i.e. the obfuscation increases) the slope of the linear regression diminishes. For example, for the omniHW (Figure 4.5b), a grid of 100m squared cells is required to get an adversary error of around 75m for $\epsilon = 16$ km^{-}1. For the remaining values of ϵ ($\epsilon = [32, 64, 128]$) however, a similar adversary error is achieved using a cell resolution of 250m. That is, increasing the obfuscation also increases the computational complexity required for an attack. From the user point of view, the privacy budget ϵ thus additionally relates (with inverse proportionality) to the computational power that an adversary must employ to compromise user privacy. While the smallest average adversary error is achieved using the smallest grid resolution (80m), to decrease execution time we opt to use squared cells of 125 meters for the remainder of the results. This corresponds to a total of 189×115 cells over the peninsula of San Francisco.

Figure 4.6 shows the average adversary error per Δ_t and for all ϵ values for the MM and the localization attacks (optHW, omniHW and PEBA). Similarly to the results obtained for the Geolife dataset, we can observe that the average adversary error is similar for any Δ_t, which does not reveal the effect of the frequency of reports. Another relevant result from Figure 4.6 is that the adversary error in the map-matching is lower than the localization attacks in all epsilon values. However, as the obfuscation increases the difference in the adversary error between MM and the localization attacks diminishes. This is due to the fact that the localization attacks take into consideration the use of



(a) MM (b) optHW

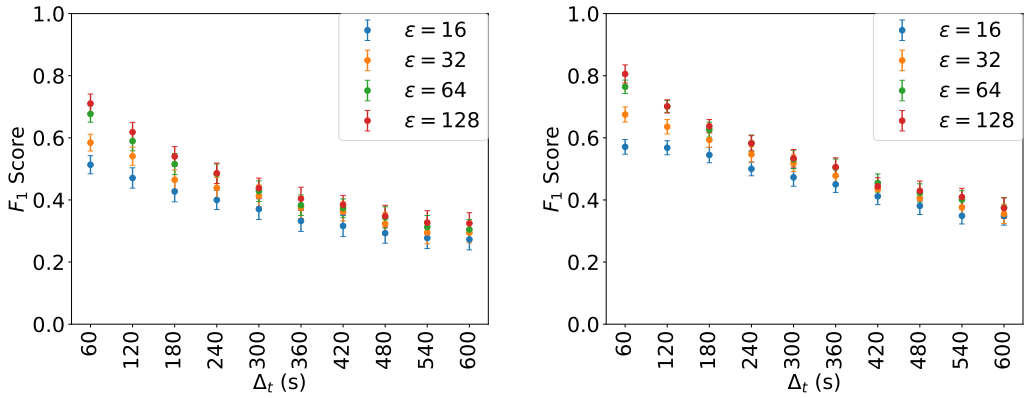
(c) omniHW (d) PEBA

Figure 4.6.: Cabspotting average adversary error and respective 95% confidence intervals per Δ_t for all values of ϵ for the MM technique and the three localization attacks.

the LPPM and hence, the localization attacks surpass the MM performance for higher obfuscation or for a smaller grid resolution. Notwithstanding note that the adversary error is not an effective privacy metric for tracking attacks. In fact, the adversary error can be close to or even zero and the F_1 score can also be zero. This extreme case occurs, for instance, when between two exact locations the matched trajectory and the true trajectory only overlap in those two points, that is, the trajectories are disjoint except in the end-points.

To assess the impact of the frequency of reports in the privacy level of geo-indistinguishability, Figure 4.7a presents the effect of the privacy budget ϵ in the F_1 score. It is visible that varying the value of ϵ has more effect when higher sampling rates (i.e. lower values of Δ_t) are employed. As the frequency becomes smaller (larger Δ_t values), there is fewer correlation between points, which naturally harms the efficacy of MM, irrespectively of the ϵ value employed. This indicates a relevant trade-off between the value of the privacy budget ϵ of geo-indistinguishability and the sampling frequency, in where lower values of ϵ can cause more obfuscation, thus possibly compensating higher frequency rates.

Comparing our results with those of the proposal of the MM technique [Jagadeesh and Srikanthan, 2017], it is clear that our F_1 scores are significantly lower. The two main differences that can be the source for this disparity



(a) Cabspotting dataset.

(b) Portocabs dataset.

Figure 4.7.: Effect of the epsilon and frequency of reports (Δ_t) in the F_1 score of the MM technique for the Cabspotting and Portocabs datasets. 95% confidence intervals are represented as the vertical lines.

are the dataset and the road network. Our dataset is from San Francisco and therefore requires the road network from San Francisco, which is significantly denser than Singapure's road network and, more importantly, highly symmetric. Consequently, multiple optimal (shortest) paths might exist between states of the map-matching leading to a F_1 score of zero for these segments.

As aforementioned, to further validate our map-matching results we considered Portocabs as an additional dataset of highly continuous location reports. Figure 4.7b presents the results obtained for this dataset. Comparing with the results obtained for the Cabspotting dataset and illustrated in Figure 4.7a, it is clear that the same conclusions can be drawn. Specifically, the degradation of the F_1 score with the increase in Δ_t (decrease in frequency of reports) and with the decrease in ϵ. It is observable that the F_1 score for this third dataset is slightly higher than for the Cabspotting case. This difference can be attributed to the already raised fact that the road network in San Francisco is highly symmetric, which can produce a relevant number of equally optimal (shortest) paths between nodes.

4.3 Limitations and Future Work

From the previous results it was possible to conclude that, given our setup, the frequency of reports does not have a significant impact on the adversary error. An inherent limitation of this finding is the limited number of attacks considered, and specifically, tracking attacks. However, we do note that, the choice of this specific attack was justified by the effectiveness in modelling temporal correlations using hidden Markov models [Xiao and Xiong, 2015; Murakami, 2017]. As future work, we would like to expand this work by considering other tracking attacks, such as Kalman filtering. Similarly, we would like to consider the effects of the frequency of reports in LPPMs which take into consideration the correlation between reports (e.g. [Xiao and Xiong, 2015; Chatzikokolakis et al., 2014; Shokri

et al., 2017]). These type of LPPMs are often output-based as opposed to the memoryless considered in this work, that is, the previous reports are considered when reporting the new obfuscated location [Oya et al., 2019]. Consequently, the frequency of reports should not only impact the attack success, but also the effectiveness of the privacy-preserving mechanism.

The conducted methodology also had some limitations. For example, the datasets are not sporadic, and arguably, subsampling for different values of minimum interval between reports (Δ_t) might not necessarily resemble a sporadic dataset. Nonetheless, to allow for fine-tuning the frequency a continuous dataset is required. Additionally, and similarly to [Chatzikokolakis et al., 2014], we argue that such subsample can be perceived as users in their quotidian trajectories making sporadic accesses to a LBS.

4.4 Chapter Summary

As users report even an obfuscated variant of their location to a Location-Based Service (LBS), information is being disclosed. The amount of usage of these services, or in other words, the frequency of reports directly impacts the correlation between reports which in turn can be used by an adversary to further degrade privacy. Geo-indistinguishability has been proposed as a formal notion based on differential privacy to bound the amount of information released on independent queries. However, the analysis on how the frequency of queries impacts the level of privacy in geo-indistinguishability was yet to be made.

In this chapter we analyze the effects of the frequency of reports in the privacy level of geo-indistinguishability. We evaluate privacy and utility against state-of-the-art localization attacks and a tracking attack. Results show that the frequency of reports has low significance in the privacy level in the sporadic release of data. These results provide practical evidence that the consideration of independence between reports can effectively be assumed in the sporadic scenario. However, in the continuous scenario, the frequency of reports directly impacts the effectiveness of the attacks, with high frequencies leading to more privacy disclosure. In such case, obfuscation degraded the correlation between reports and consequently the effectiveness of the attack, thus acting as a countermeasure to high report frequencies. Our experiments with several values of the privacy budget reveal that there is an upper bound that is required for effective privacy protection, such that values above that threshold will result in no effective privacy. Moreover, our evaluation depicts a trade-off between the frequency of reports and the privacy budget of geo-indistinguishability, showing that lowering the frequency or increasing the level of noise (i.e. decreasing the privacy budget) are effective measures that can be applied independently against continuous gathering of location data.

Based on these insights, in the following chapter we propose a novel notion that generalizes geo-indistinguishability to location traces. This notion, termed Velocity-Aware Geo-Indistinguishability (VA-GI), adjusts the privacy or utility as a function of the user velocity and report frequency, thus effectively adapting

to varying density of reports, both in the time and space continua. Additionally, this privacy and utility adjustment is automatic and requires only two user-set parameters, thus mitigating misconfigurations that can lead to no effective privacy. Finally, the mechanism can be personalized to a single user or a specific region, such that the privacy and utility adaptability is tailored to specific driving conditions and preferences. The described properties of VA-GI are paramount to the integration of such mechanism in a permission manager as it is desirable that the LPPM automatically adapts to varying conditions, such as multiple apps accessing user location or different means of transportation, while requiring minimal configuration, as a statically set parameter would result privacy or utility levels depending on situation/context.

Chapter 5.

Location Privacy-Preserving Mechanisms for Continuous Location Reports

Contents

Effective and practical privacy in mobile devices is challenging due to the continuously changing privacy and utility requirements. The previous chapter has identified two crucial issues in the context of location privacy and, particularly, in Geo-Indistinguishability. The first relates to how the frequency of reports can impact privacy, in where as the frequency increases, so does the correlation between successive reports that can be explored by an adversary. In the context of a mobile device, these varying frequency of reports can result from a single application with both sporadic and continuous modes, or from all apps accessing location in an asynchronous manner. The second identified issue is the tuning of privacy parameters in dynamic environments, in this particular case, in the presence of varying frequency of reports. Specifically, results showed that a poorly set privacy budget ϵ can result in no effective privacy. These conclusions motivate not only the need for LPPMs that are effective under continuous reports, a line of research that has recently seen an increase in interest [Liu et al., 2018a], but also LPPMs that dynamically adapt to varying contexts, such as different frequencies of reports. This adaptability should not come as a burden for users and, therefore, mechanisms for automated configuration of privacy parameters are required.

Adaptations of Geo-Indistinguishability have been proposed to the scenario of online continuous release of location data [Chatzikokolakis et al., 2014; Al-Dhubhani and Cazalas, 2018; Cunha et al., 2019]. Such approaches resort to estimations and distance metrics to evaluate the correlation between reports and subsequently apply obfuscation accordingly. However, using simple estimators such as linear regressions result in a non-negligible amount of outliers due to time-gaps in reports, which occur due to failures in the GPS/communications [Mendes and Vilela, 2018]. Additionally, dynamically adapting the obfuscation requires additional parameters that a user must configure. This is often challenging [Kaaniche et al., 2020] and potentially misleading [Clifton and Tassa, 2013; Lee and Clifton, 2011; Oya et al., 2017b], specially since users are typically unaware of the privacy risks and privacy-utility trade-offs [Acquisti et al., 2015]. Moreover, a misconfigured parameter can result in no relevant privacy protection (c.f. Chapter 4). Therefore, for practical and wide adoption, a largely under-developed aspect in this field, LPPMs should be designed such that the required user-set parameters are minimal, and can be set in an automated manner.

In this chapter we argue that the correlation between reports can be estimated by the velocity of the user and the frequency of reports. Consider the following example as an illustration of this argument: a user reporting his location every 30 seconds while walking ($\sim$5 km/h) will have a point every $\sim$ 42 meters. If the same user was driving in an highway at 120km/h, a point every 1000 meters would be reported instead. Even though the frequency of reports is the same, the correlation between points might be lower in the case of the highway, as the speed of the user is higher and therefore, the points are sparser. A similar (yet inverse) effect is observed for a constant user speed and varying frequency of

Table 5.1.: Desired behavior of a velocity-aware LPPM as a function of the velocity of the user (v_u) and of the velocity of reports (v_r). The symbols $\uparrow$ and $\downarrow$ denote a high and a low value, respectively.

Velocity		**Desired Result**
$\uparrow v_u$	$\uparrow v_r$	Balance Privacy and Utility
$\uparrow v_u$	$\downarrow v_r$	Favor Utility
$\downarrow v_u$	$\uparrow v_r$	Favor Privacy
$\downarrow v_u$	$\downarrow v_r$	Balance Privacy and Utility

reports. If the same user in the highway at 120km/h would instead report every 300 seconds (5 minutes), the distance between reports would increase to 10km. In conclusion, the reports become sparser as the velocity of the user increases or the frequency of reports decreases. Inversely, the reports become denser as the user velocity decreases or the frequency increases.

The previous example paired with the degradation of privacy with the increase in the correlation [Krumm, 2009; Liu et al., 2018a; Xiao and Xiong, 2015] and frequency of reports from Chapter 4 lead us to the following conclusion. From the point-of-view of a privacy-preserving mechanism, high frequency of reports or a low user velocity should be met with an increase in obfuscation, to increase privacy, and, a low frequency of reports or a high user velocity should be met with a decrease in obfuscation, as to increase utility. Following these desired properties, which are summarized in Table 5.1, this chapter proposes a generalization of Geo-Indistinguishability for effective and efficient privacy preservation under online continuous reports. In this proposal, termed Velocity-Aware Geo-Indistinguishability (VA-GI), the velocity of the user and the frequency of reports are used to dynamically adapt the privacy and utility level. Building on this notion, we present a VA-GI LPPM that requires only two user-set parameters, thus simplifying usability and allowing for wide deployment.

The contributions of this chapter are as follows:

- We generalize Geo-Indistinguishability by taking into account the velocity of the user and the frequency of reports in a novel notion termed Velocity-Aware Geo-Indistinguishability (VA-GI).
- We devise a VA-GI LPPM that according to our empirically evaluation with real trajectories, outperforms previous literature LPPMs regarding the dynamic adaptability between privacy and utility under different scenarios. Moreover, by using mobility data in its parametrization, the proposed LPPM requires only two user-set parameters, thus facilitating usability and mitigating misconfigurations that can lead to no effective privacy (c.f. Chapter 4). Furthermore, the considered mobility data for parameterization can be from a specific region or from a single person, thus providing an adaptability to the environment in which it is applied or personalized to the user.
- We generalize the VA-GI LPPM for wide deployment through an approximation of the formula for setting privacy parameters using publicly avail-

able data. We provide empirical evidence on the feasibility and effectiveness of doing so. Specifically, by using data from one location to formulate the LPPM and evaluating such formulation on another dataset from a different location results in relative differences of the configured privacy parameters inferior to 10%.

The remainder of this chapter is structured as follows. Section 5.1 formally describes the notion of VA-GI and presents a VA-GI LPPM. Section 5.2 describes the experimental setup, whose results are presented and discussed in Section 5.3. Section 5.4 proposes and evaluates a generalization of the VA-GI LPPM for wide deployment. Section 5.5 discusses limitations and future remarks and Section 5.6 concludes this chapter.

5.1 Velocity-Aware Geo-Indistinguishability

In differential privacy, the privacy budget ϵ is set in accordance to certain privacy and utility needs. Specifically, a decrease in the epsilon corresponds to higher obfuscation, which increases privacy but reduces utility, and vice-versa for an increase in epsilon. However, setting the value of ϵ is challenging as it highly depends on the data (or dataset), specially in the presence of correlations between the data [Clifton and Tassa, 2013; Lee and Clifton, 2011]. In fact, in the context of location privacy it has been shown that there is an upper bound on the value of the privacy budget necessary to guarantee relevant privacy protection [Mendes et al., 2020].

Geo-Indistinguishability, as detailed in Section 2.4.2.1, slightly improves the interpretability and therefore ease the configuration by defining $\epsilon = l/r$, where l is the privacy loss and r is the radius within up to l privacy loss is achieved [Andrés et al., 2013]. However, the value of l is still heuristically set and more importantly, each report is considered independent. From the composability properties of differential privacy, and hence in Geo-Indistinguishability, the privacy loss increases linearly with the number of reports. Therefore, this notion is only suitable for sporadic reports, as previously mentioned.

To solve the privacy budgeting problem under continuous reports, we propose a generalization of Geo-Indistinguishability termed Velocity-Aware Geo-Indistinguishability (VA-GI). VA-GI adjusts the privacy and utility as a function of the user's velocity and the frequency of reports in accordance with the desired behavior of a velocity-aware LPPM as described in Table 5.1. For this dynamic adaptability, we set the privacy budget ϵ as a function of both velocities. Formally, for each timestamp i, ϵ is set dynamically as:

$$\epsilon_i \coloneqq \epsilon_i(v_{u,i}, v_{r,i}) \tag{5.1}$$

where $v_{u,i}$ and $v_{r,i}$ are the velocity of the user and the velocity (or frequency) of the reports at timestamp i, respectively. This formulation lead us to definition 1.

Definition 1. An obfuscation mechanism $K(\cdot)$ is Velocity-Aware Geo-Indistinguishable iff for any timestamp i:

$$d_{\mathcal{P}}\left(K(x_i), K(x_i')\right) \leq \epsilon_i(v_{u,i}, v_{r,i}) \cdot d(x_i, x_i'), \quad \forall x_i, x_i' \in \mathcal{X}$$

Definition 1 states that the difference in the output of a VA-GI mechanism $K(\cdot)$ with input location x_i or x_i' at timestamp i differs at most by the distance between both locations multiplied by a variable privacy budget that is function of the user and frequency velocities at the same timestamp i. Note, however that contrary to Geo-Indistinguishability, the privacy bound depends the bounds of the function $\epsilon_i(\cdot)$, which we discuss next.

In order to achieve the desired behavior for a velocity-aware LPPM as described in Table 5.1, ϵ_i must increase with an increase in the velocity of the user or a decrease in the frequency of reports and decrease with the decrease of the user velocity or an increase in the frequency of reports. Formally, we can describe this requirement as:

$$\epsilon_i(v_{u,i}, v_{r,i}) \propto v_{u,i} \quad \wedge \quad \epsilon_i(v_{u,i}, v_{r,i}) \propto \frac{1}{v_{r,i}} \tag{5.2}$$

that is, ϵ_i is directly proportional to the user velocity and inversely proportional to the frequency of reports. Towards this goal, we depart from the standard Geo-Indistinguishability where $\epsilon = l/r$ [Andrés et al., 2013], and set the privacy budget as:

$$\epsilon_i = \frac{\epsilon}{m} \cdot m^{(2 \cdot f(v_{u,i}, v_{r,i}))} \tag{5.3}$$

where m is a privacy and utility multiplier (as further discussed below) with $m \in [1, \infty[$, $v_{u,i}$ and $v_{r,i}$ are the user and report velocities at timestamp i, and $f(\cdot)$ is any function of $v_{u,i}$ and $v_{r,i}$, that holds the proportionality from equation (5.2) and with $f(\cdot) \in [0, 1]$.

Equation (5.3) corresponds to the exponential regression on $f(\cdot)$ such that the following bounds for ϵ_i are achieved:

$$\frac{\epsilon}{m} \leq \epsilon_i \leq m \cdot \epsilon \;\Leftrightarrow\; \frac{l}{m} \leq r \cdot \epsilon_i \leq m \cdot l, \quad \forall i \tag{5.4}$$

where the multiplier m is used to adjust the privacy and utility bounds. Equations (5.3) and (5.4) provide a dynamic balance in where the privacy and utility levels can be increased or decreased up to m times the initial ϵ value, depending on the velocity of the user and frequency of reports. As long as $f(\cdot)$ provides the proportionality from (5.2), an increase in the user velocity ($v_{u,i}$) and/or a decrease in the frequency of reports ($v_{r,i}$) is met with an increase in the privacy budget ϵ_i, and vice-versa for a decrease in ϵ_i. Therefore, we refer to this VA-GI formulation as (m, ϵ)-VA-GI. Finally note that if $m = 1$, $(1, \epsilon)$-VA-GI becomes Geo-Indistinguishability as $\epsilon_i = \epsilon$, $\forall i$. Therefore, (m, ϵ)-VA-GI can be seen as a generalization of Geo-Indistinguishability. This result leads us to

Theorem 1.

Theorem 1. (m, ϵ)-VA-GI satisfies $m\epsilon$-Geo-Indistinguishability and guarantees a maximum privacy loss of $m \cdot l$ within a radius r. Namely, for any timestamp:

$$d_{\mathcal{P}}\left(K(x_i), K(x_i')\right) \leq m \cdot l, \ \forall x_i, x_i' \in \mathcal{X} \quad \textbf{s.t.} \quad d_{\mathrm{x}}(x_i, x_i') \leq r$$

Proof. The proof results from the proof of Geo-Indistinguishability for the Planar Laplace, with the difference being the privacy bounds due to the changes in ϵ. Without loss of generality and to simplify notation, the timestamp i is dropped from the proof. From equation (2.6) we have:

$$\frac{D_x(z)}{D_{x'}(z)} = e^{-\epsilon(d_{\mathrm{x}}(x,z) - d_{\mathrm{x}}(x',z))}$$

And from triangular inequality we obtain:

$$D_x(z) \leq e^{\epsilon d_{\mathrm{x}}(x,x')} D_{x'}(z)$$

Through integration we reach:

$$\begin{aligned} \int_Z D_x(z) \leq e^{\epsilon d_{\mathrm{x}}(x,x')} \int_Z D_{x'}(z) \quad \Leftrightarrow \\ \Leftrightarrow \quad K(x)(Z) \leq e^{\epsilon d_{\mathrm{x}}(x,x')} K(x')(Z) \end{aligned} \tag{5.5}$$

This equation proves the Geo-Indistinguishability of the Planar Laplace and can be re-written as:

$$d_{\mathcal{P}}\left(K(x), K(x')\right) \leq \epsilon d_{\mathrm{x}}(x, x') \quad \forall x, x' \in \mathcal{X} \tag{5.6}$$

where $d_{\mathcal{P}}(\cdot)$ is the multiplicative distance between two distributions, defined as $d_{\mathcal{P}}(\sigma_1, \sigma_2) = \sup_{S \in \mathcal{S}} \left|\log \frac{\sigma_1(S)}{\sigma_2(S)}\right|$, where σ_1 and σ_2 are two distributions on some set S, with the convention that $\mathcal{L} = \left|\log \frac{\sigma_1(S)}{\sigma_2(S)}\right| = 0$ if $\sigma_1(S) = \sigma_2(S) = 0$ and $\mathcal{L} = \infty$ if one of the two is 0.

Now we must prove the (m, ϵ)-VA-GI privacy bound within r, which differs from the standard Planar Laplace (PL) due to the changes to ϵ. From equation (5.4) we have:

$$\frac{l}{m} \leq r \cdot \epsilon_i \leq m \cdot l, \quad \forall i$$

Consequently, from equation (5.6) and for any two x, x' such that $d_{\mathrm{x}}(x, x') \leq r$:

$$d_{\mathcal{P}}\left(K(x), K(x')\right) \leq m \cdot l \tag{5.7}$$

■

5.1.1 A (m, ϵ)-VA-GI LPPM

Equation (5.3) defines the generic formula to achieve (m, ϵ)-VA-GI, in where any definition of the function $f(\cdot) \in [0, 1]$ that respects the proportionalities from (5.2) can be used. However, for an effective privacy and utility balance, the function should respect the nature of the velocities, specifically their distributions. Unfortunately, the velocities do not follow any unimodal distribution and in fact depend on the underlying road features and drivers [Tonguz et al., 2009]. Therefore, to design a (m, ϵ)-VA-GI LPPM one can approximate or estimate the distributions by using available data. This section describes such methodology.

Since there is no a priori best choice, we leave the comparison between VA-GI LPPMs for future work and instead choose a simple velocity function $f(\cdot)$ defined as the average between a function of the user velocity $f_u(v_{u,i})$ and a function of the report velocities $f_r(v_{r,i})$:

$$f(v_{u,i}, v_{r,i}) = \frac{1}{2} \cdot (f_u(v_{u,i}) + f_r(v_{r,i})) \tag{5.8}$$

Where $f_u(\cdot),\ f_r(\cdot) \in [0, 1]$. To take into consideration the distributions of $v_{u,i}$ and $v_{r,i}$ and to favor the variance of f_u and f_v near the typical values of $v_{u,i}$ and $v_{r,i}$, we set:

$$\begin{aligned} f_u(v_{u,i}) &= cdf(v_{u,i}) \\ f_r(v_{r,i}) &= 1 - cdf(v_{r,i}) \end{aligned} \tag{5.9}$$

where $cdf(\cdot)$ stands for the CDF. With equation (5.9) we guarantee that the codomain of $f(v_{u,i}, v_{r,i})$ as defined in equation (5.8) is in the interval $[0, 1]$ and that the proportionality of a VA-GI LPPM from equation (5.2) is respected. Additionally, because the slope of the CDF is higher in the typical values, smaller deviations from these will have a steeper privacy/utility adjustment. Combining equation (5.8) and (5.9) in equation (5.3), we reach:

$$\epsilon_i = \epsilon \cdot m^{(cdf(v_{u,i}) - cdf(v_{r,i}))} \tag{5.10}$$

The advantage of using the $cdf(\cdot)$ functions of the user velocities and frequency of reports relates to the minimization of the required parameters. Effectively setting privacy parameters can be challenging [Kaaniche et al., 2020; Clifton and Tassa, 2013; Lee and Clifton, 2011] and in fact, misconfigured parameters can result in no relevant privacy [Mendes et al., 2020]. By using equations (5.9) in the (m, ϵ)-VA-GI privacy budget equation (5.3), we limit the LPPM to 2 parameters: the initial ϵ value and the multiplier m. This is in contrast with other LPPMs for continuous report, that either require several parameters to provide the dynamic adaptability, such as the Adaptive Geo-Indistinguishability, or that have few parameters but do not adapt to the dynamics of the movement (e.g. Clustering Geo-Indistinguishability and the Planar Laplace). Sections 5.2.2 and 5.3 demonstrate these disadvantages of previous works.

One of the disadvantages of this approach is that the distribution of the velocities is unknown. To solve this issue, one can use data to estimate the CDFs using non-parametric density estimation. From the point of view of the LPPM, this data can belong to all drivers in a specific city, global or even be personalized to the user, by using their past data. Regardless of the data used, a better CDF fit will favor the privacy and utility trade-off. Nevertheless, to faithfully fit a CDF, a decent amount of data is required. In Section 5.4 we provide empirical evidence on the effectiveness of generalizing the CDFs to Gaussian distributions in the context of (m, ϵ)-VA-GI.

Since positional data often lacks the velocities, these can be estimated by using a window of previous reports with size ws as follows:

$$v_{u,i} = \frac{d_i^{ws}}{\Delta_{t_i^{ws}}} \qquad v_{r,i} = \frac{ws - 1}{\Delta_{t_i^{ws}}} \tag{5.11}$$

where d_i^{ws} is the sum of distances between reports in ws and $\Delta_{t_i^{ws}}$ the time difference between the current report of timestamp i and the last in the window of size ws, $i - ws + 1$. Formally:

$$\begin{aligned} d_i^{ws} &= \sum_{j=i-ws+1}^{i} d(x_{j-1}, x_j) \\ \Delta_{t_i^{ws}} &= t_i - t_{i-ws+1} \end{aligned} \tag{5.12}$$

Finally note that with these definitions we have that $v_{u,i}$ and $v_{r,i}$ are average velocities between points in ws. Within this context, the velocities will be most accurate when the time interval tends to 0, that is $\Delta_{t_i^{ws}} \to 0$ [Petovello, 2015]. Since the frequency of reports is application/device specific, using the minimum window size ($ws = 2$) will result in the best velocities estimation.

In summary, the (m, ϵ)-VA-GI LPPM consists of using the PL mechanism from equation (2.6) with the epsilon definition from equation (5.10).

5.1.2 Setting LPPM Parameters

One of the challenges in the wide deployment of privacy mechanisms is the configuration parameters. In differential privacy, for instance, setting the value of ϵ depends on the dataset and must take into account the presence of correlations [Clifton and Tassa, 2013; Lee and Clifton, 2011]. Additionally, the privacy guarantees can be misleading [Clifton and Tassa, 2013; Oya et al., 2017b]. This is specially true for mechanisms that act at collection time, where the responsibility to properly tune the mechanism lies on the user. However, it has been shown that the users are typically unaware of the privacy risks and trade-offs [Acquisti et al., 2015].

In Geo-Indistinguishability, as inherited from differential privacy, the privacy budget ϵ is the only parameter needed to set. To help the user choosing this value, the original authors suggested using $\epsilon = l/r$, where l is the maximum privacy loss achieved within the radius r [Andrés et al., 2013], as detailed in Sec-

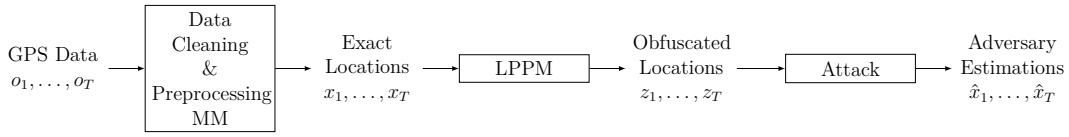


Figure 5.1.: Diagram of the followed methodology. The LPPM step is repeated for each of the LPPMs and the Attack step is repeated for each combination of LPPM/Attack.

tion 2.4.2.1. The Clustering Geo-Indistinguishability described in Section 2.4.2.2 re-uses the same variables by setting $r = l/\epsilon$. However, these two LPPMs are static with respect to the privacy and utility trade-off, as the parameters are constants, thus failing at providing effective privacy against correlated reports.

To solve this issue, the Adaptive Geo-Indistinguishability mechanism described in Section 2.4.2.3 uses a linear estimator to measure the correlation between past and current locations, thus dynamically adapting the privacy and utility. However, to do so, it introduces four parameters: the two thresholds Δ_1 and Δ_2 and the two multiplicative constants α and β. Without having data to empirically evaluate the effects of these parameters it is an impossible task to achieve an effective privacy/utility trade-off (c.f. Section 5.2.2). Furthermore, misconfigured parameters can result in no effective privacy as demonstrated in the previous chapter. In contrast, the (m, ϵ)-VA-GI LPPM only requires the initial ϵ value and a straightforward privacy/utility multiplier m. Therefore, to set the ϵ and m, one can, for instance, see the typical range of values from the literature and set ϵ to the mid value of such range and then set m such that ϵ_i automatically adjusts within the range, so as to either favor privacy or favor utility, as required by the context at hand.

5.2 Experimental Setup

This section describes the conducted simulations by detailing the datasets and experimental setup. To evaluate the effectiveness of (m, ϵ)-VA-GI, our methodology consisted in applying the LPPMs detailed in Section 2.4.2 to the data, namely, the Planar Laplace, Clustering and Adaptive Geo-Indistinguishability, followed by each of the attacks. The results are compared between the output of the attacks and the original dataset, along with the comparison between the different LPPMs and respective configurations. The diagram in Figure 5.1 summarizes the methodology, which is repeated (except the preprocessing MM) for each pair of LPPM/Attack. The following subsections detail the dataset and respective preprocessing, and the configurations/parameters of the LPPMs and attacks.

5.2.1 Dataset Characterization and Preprocessing

The dataset used in our experiments was the Cabspotting, a dataset of taxi trajectories over the city of San Francisco, California, USA. The trajectories belong to 536 taxis and were collected over a period of 30 days, containing not

only the GPS position and timestamp, but also whether the cab had a costumer at each time [Piorkowski et al., 2009].

To preprocess the dataset, we first filtered out trajectories with points outside the bounding-box defined from South and North by the latitudes 37.600, 37.811, and from West and East by longitudes -122.517, -122.354 . We additionally removed trajectories without occupancy, as to avoid trajectories were the cab is still waiting for a client. Finally, to remove spurious trajectories, we applied the data cleaning procedure from [Wang et al., 2015a]. Specifically, we discarded 1) trips with duration lower than 1 minute and higher than 3 hours; 2) trips with total displacement over 100 km; 3) trips with average velocity lower than 5 km/h or over 120 km/h; 4) non-smooth trips. Non-smooth trips were removed by using a filter with a sliding window that detects whether the average velocity between points is within normal intervals. If more than a defined percentage of points in each trajectory has abnormal average velocities, then the trajectory is rejected. The original default parameters were used for this filter [Wang et al., 2015a]. After this preprocessing, 307983 trajectories remained from the original dataset.

Because we apply four LPPMs under different configurations and multiple attacks, we further subsampled the dataset as to reduce the number of trajectories. In order to evaluate the adaptability of the LPPMs under continuous reports, we divide the trajectories in four different sets depending on the average user velocity v_u and average report velocity v_r:

1. Balance Privacy/Utility 1 ($\downarrow v_u \downarrow v_r$): trajectories with average user velocity $v_u \leq 20$ km/h and velocity of reports $v_r \leq 45$ reports/h;
2. Favor Privacy ($\downarrow v_u \uparrow v_r$): trajectories with average user velocity $v_u \leq 20$ km/h and velocity of reports $v_r \geq 100$ reports/h. This is the worst case with respect to privacy, as it has the largest density of reports per distance traveled. Therefore, and according with the desirable properties from Table 5.1, LPPMs should ideally adjust for privacy to account for the higher correlation between reports;
3. Favor Utility ($\uparrow v_u \downarrow v_r$): trajectories with average user velocity $v_u \geq 100$ km/h and velocity of reports $v_r \leq 45$ reports/h. This scenario has the lowest density of reports per distance traveled and hence, the lowest correlation between reports. Therefore, the LPPMs should ideally adjust to improve utility;
4. Balance Privacy/Utility 2 ($\uparrow v_u \uparrow v_r$): trajectories with average user velocity $v_u \geq 100$ km/h and velocity of reports $v_r \geq 100$ reports/h. This scenario is similar to the "Balance Privacy/Utility 1 ($\downarrow v_u \downarrow v_r$)" with respect to the density of reports and therefore to the desired response.

The threshold values were chosen by looking at the speed limits[1] and empirical distributions. Specifically, speed limits in alleys and residential areas are 24 and 40 km/h, respectively, and therefore, a vast number of trajectories will have an

[1] `https://data.sfgov.org/Transportation/Map-of-Speed-Limits/ttcm-fwt2`

average speed lower to 20 (c.f. Figure 5.3a). The high user velocity trajectories, with average over 100 km/h, will correspond to trajectories in highways, where the speed limit is 105 km/h. For the frequency of reports, we picked intervals directly from the empirical distribution, illustrated in Figure 5.3b.

From the four data set divisions, we picked the 100 trajectories from each partition with lowest standard deviation, as to select trajectories where the instant velocities (velocities in each report) are closest to the filtered mean values. This selection was made as to have a strong diversity of trajectories, encompassing scenarios with a high, medium and low density of reports per trajectory. In turn, this density relates directly to the difficulty of an adversary in reconstructing the real trajectory, which is the most general type of attack [Shokri et al., 2011] that then allows for further specific inference attacks [Gambs et al., 2010; Primault et al., 2014]. For the remainder of this work, we refer to the selected 400 trajectories as test data (or ground-truth) and the remaining cleaned trajectories (307583) as training data.

To the test data, we apply the map-matching technique detailed in Section 2.4.3.3 as to position each location report on the road network. This preprocessing step cleans these trajectories from noisy reports, thus forming our ground-truth. For the standard deviation of the measurement error σ, we used a typical value for GPS readings of $\sigma = 6.86$ m [Goh et al., 2012]. The parameters λ_y and λ_z were estimated following the original map-matching proposal [Jagadeesh and Srikanthan, 2017]. Namely, using trajectories from the training data with duration between 1 and 6 minutes with at least 2km of traveled distance (for a total of 4963 trajectories). This resulted in the values $\lambda_y = 0.69$ and $\lambda_z = 13.35$.

5.2.2 LPPMs

For the LPPMs, we compare the (m, ϵ)-VA-GI LPPM from Section 5.1.1, which we simple refer to as **VA-GI**, with the geo-indistinguishable LPPMs described in Section 2.4.2, that is, the Planar Laplace [Andrés et al., 2013], which we refer to as **Geo-Ind**, the Clustering Geo-Ind [Cunha et al., 2019], referred to as **Clustering**, and the adaptive Geo-Ind [Al-Dhubhani and Cazalas, 2018], or **Adaptive**.

For the privacy budget, and for all LPPMs, we used multiple values in the typical ranges of LPPMs for continuous reports [Al-Dhubhani and Cazalas, 2018; Mendes et al., 2020], specifically $\epsilon = [16, 32, 64, 128]$ km^{-1}. For the Geo-Ind LPPM, this corresponds to an average obfuscation of $[125, 62.5, 31.25, 15.625]$ m, respectively. These values of obfuscation range from city block level distances to parallel streets. For the remaining parameters we attempted to use the proposed values from the original respective papers, but we found some problems in the Adaptive as follows.

In the Adaptive mechanism, the privacy budget ϵ is adjusted for privacy or utility depending on the error in estimating the current location, as described in Section 2.4.2.3. In accordance with equation (2.8), if the estimation error is

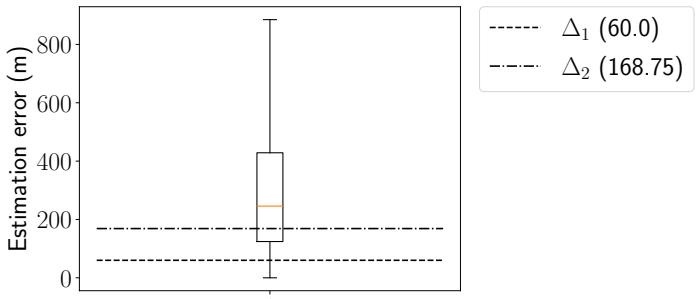
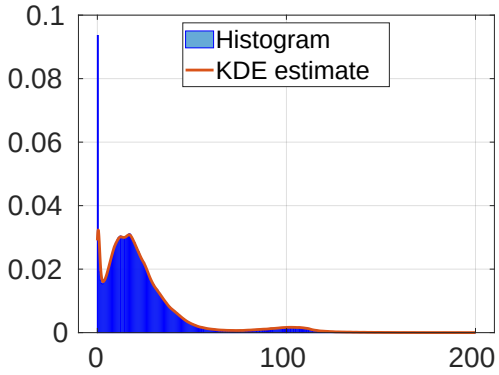


Figure 5.2.: Boxplot of the Adaptive estimation errors and the Δ_1 and Δ_2 original thresholds.

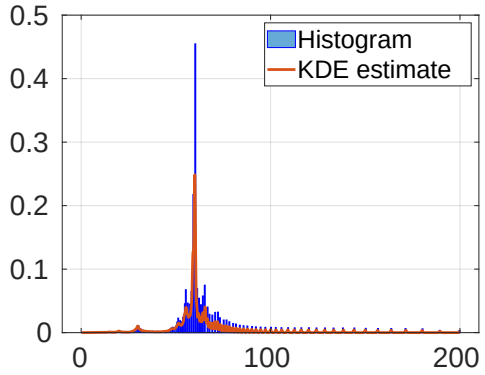
smaller than Δ_1, then privacy is increased by reducing ϵ by a factor of α. If the estimation error is higher than Δ_2 the utility is increased by increasing ϵ by a factor of β. The authors heuristically proposed setting $\Delta_1 = 0.96/\epsilon$ and $\Delta_2 = 2.7/\epsilon$. However, for the ϵ values used in our work, we found that these thresholds result in a poor privacy and utility adaptability. Figure 5.2 illustrates this problem by plotting a boxplot of the estimation errors ($d_2(x, \hat{x})$) for all points in the training data and the thresholds for $\epsilon = 16$ km^{-1}. From this plot it is clear that for almost 75% of location reports, the adaptive would adjust for utility, and only for less than approximately 15% of cases, it would adjust for privacy. This unbalance is even worse for higher ϵ values, as the estimation errors are the same, but the thresholds would be lower. In order to have a proper privacy/utility dynamic, we set the Δ_1 and Δ_2 thresholds to the first ($\Delta_1 \approx 124.29$) and third quartile ($\Delta_2 \approx 428.56$) of the boxplot. We refer to this tuned LPPM to as **Adaptive*** and only present the results for this optimized variant of the original adaptive mechanism. This example illustrates the difficulty in setting the proper parameters, as discussed in Section 5.1.2, a problem that VA-GI solves by using the cumulative density functions, as previously discussed.

As for the VA-GI, and following the description from Section 5.1.1, we use a non-parametric estimation of the Cummulative Density Function (CDF) using the training data, specifically, a Kernel Density Estimation (KDE). Figures 5.3a and 5.3b present the histograms of the user velocities and report velocities, respectively, and the respective KDE Probability Density Function (PDF). From these images it is clear that both velocities do not follow a Gaussian distribution and the Kolmogorov-Smirnoff tests reject such hypothesis at any significance level. For the KDE we used Guassian kernels and let Matlab find the optimal bandwidth. We can clearly see from the plots that while the KDE for the user velocities strongly fits, the KDE for the report velocities is more inconsistent. This is not problematic from the point-of-view of the VA-GI, as what is important are the CDF functions. Figures 5.3c and 5.3d present the empirical and KDE CDF for the user and report velocities, respectively. We can clearly see that the lines are mostly coincident in both cases, thus confirming that the KDE is a good estimator for the CDF. In summary, for VA-GI, at each timestamp i we compute ϵ_i as defined in equation (5.10) with the KDE CDFs.

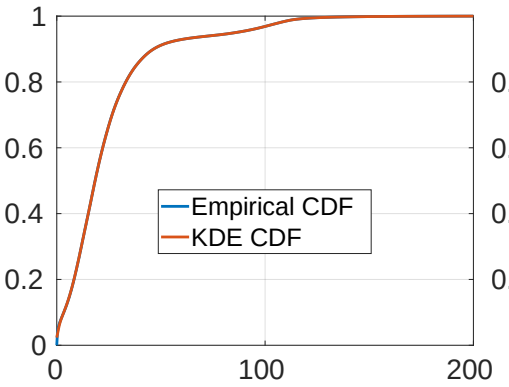
One can plot equation (5.10) as a function of the velocities of the users and



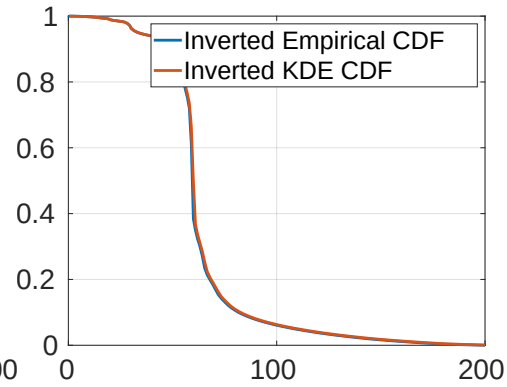
(a) Histogram of the users' velocities and respective kernel fit.



(b) Histogram of the reports' velocities and respective kernel fit.



(c) Empirical and kernel CDF of the users' velocities.



(d) Empirical and kernel CDF of the users' velocities.

Figure 5.3.: Empirical and kernel density estimation probability density functions and cumulative density functions of the reports' velocities for the training data.

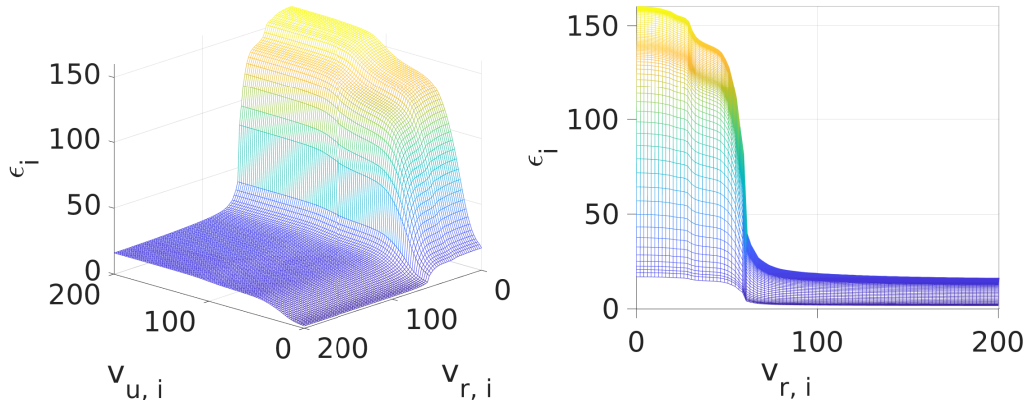


Figure 5.4.: 3-dimensional plot of the value of ϵ_i as given by equation (5.10) as a function of $v_{u,i}$ and $v_{r,i}$, with $\epsilon = 16$ km^{-1} and $m = 10$.

reports. Figure 5.4 presents this plot with $\epsilon = 16$ km^{-1} and $m = 10$. We can observe that as the velocity of the user increases, the value of ϵ_i increases as to reduce the obfuscation and therefore increase utility, and vice-versa for a decrease in the velocity as to increase privacy. The velocity of reports as the

inverse effect, which is in accordance with Table 5.1.

For the remainder of the paper, the results for the VA-GI mechanism were obtained with $m = 10$. This value was chosen such that typical epsilon values (c.f. [Andrés et al., 2013; Al-Dhubhani and Cazalas, 2018; Mendes et al., 2020]) were contained within the ϵ_i bounds defined in equation (5.4).

5.2.3 Attacks

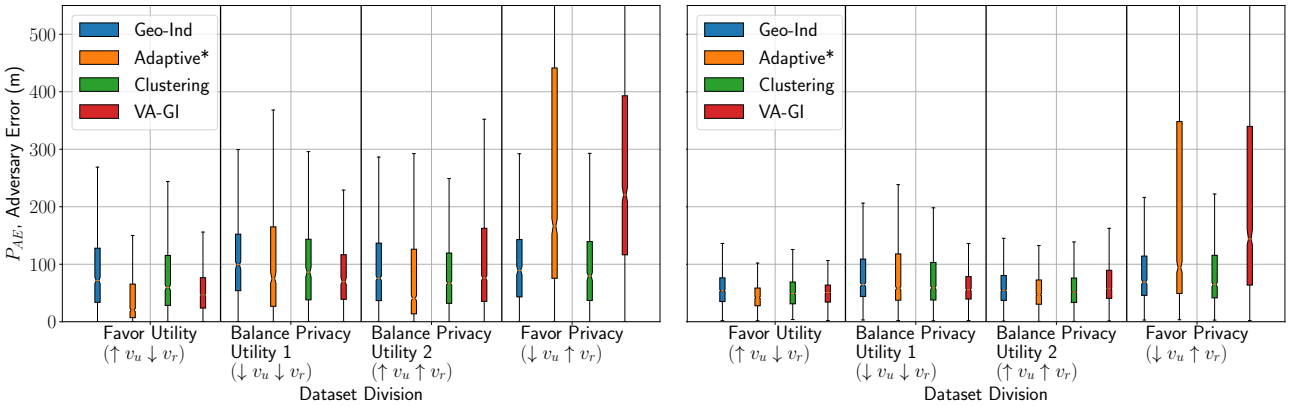
For the attacks we consider the optimal localization attack from Section 2.4.3.1 and the map-matching attack from Section 2.4.3.3. For the map-matching, while the original authors presented a complex route choice model, the increase in the accuracy was marginal when compared to using the shortest path [Jagadeesh and Srikanthan, 2017]. Therefore, in this work we opted for the simple shortest path to reduce computational complexity. For efficiency, and similarly to [Goh et al., 2012], we only consider candidates nodes within a radius r which we calculate using the inverse cumulative distribution function of the Gaussian distribution. The radius r is computed such that the circle centered at the observation contains the exact location with 90% probability given a geo-indistinguishable obfuscation. When this circle contains no candidates, which can happen due to the use of the LPPM and selected road network, the nearest road network node is used as candidate. The road network was obtained from OpenStreetMap using the OSMnx tool [Boeing, 2017] over the San Francisco bay area.

5.3 Results

This section presents the results obtained following the presented methodology. Because the findings endure for all epsilon values, we present the results only for $\epsilon = 16$ km^{-1}.

Figure 5.5 shows the adversary error for the map-matching and optimal localization attack. From this figure we can observe that Geo-Ind and Clustering have similar adversary error (privacy level) for the different dataset divisions and for both attacks. However, the Adaptive* and VA-GI largely vary. Specifically, for the "Favor Privacy ($\downarrow v_u \uparrow v_r$)" division these two LPPMs have greater adversary errors and for "Favor Utility ($\uparrow v_u \downarrow v_r$)" the lowest. These results indicate that both the Adaptive* and the VA-GI properly adapt in accordance with the desired properties of a velocity-aware LPPM, as described in Table 5.1. However, the large increase in the adversary error comes with the consequence of a high quality loss, as displayed in Figure 5.6. This is the natural and ever present trade-off between privacy and utility [Cranor et al., 2015].

According to Figures 5.5 and 5.6, the VA-GI had the strongest privacy (highest adversary error) for the "Favor Privacy ($\downarrow v_u \uparrow v_r$)" scenario, while the Adaptive* had the best utility (lowest quality loss) for the "Favor Utility ($\uparrow v_u \downarrow v_r$)" division. However, due to the fact that the quality loss and adversary error metrics do not take into consideration the continuous nature of the trajectories, these results can be inconclusive. Therefore, in the following discussion we focus on the



(a) Map-matching

(b) Optimal Localization Attack

Figure 5.5.: Boxplot of the adversary errors for the Map-matching and optimal localization attack for $\epsilon = 16$ km^{-1}.

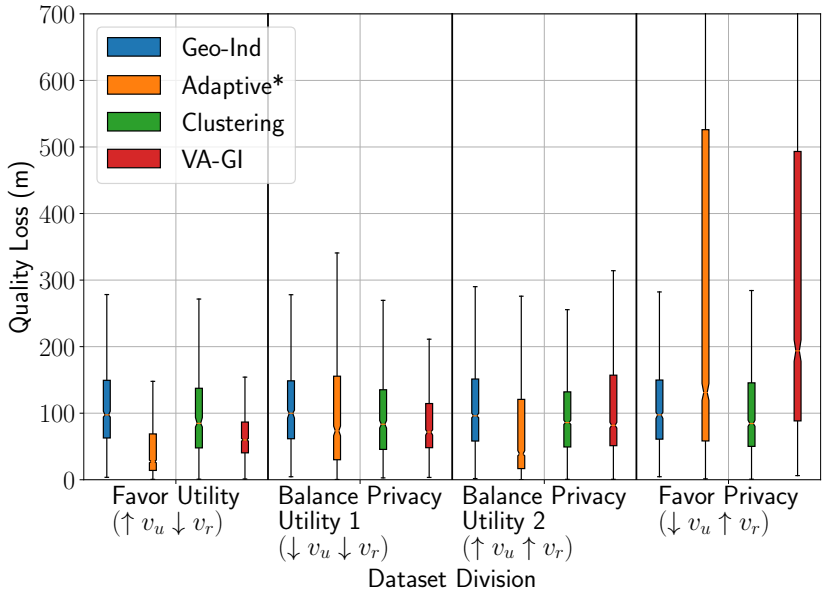


Figure 5.6.: Boxplot of the quality loss for each dataset division and LPPM, with $\epsilon = 16$ km^{-1}.

F_1-score metric as it considers not only the user individual reports but every traversed segment between each location, as described in Section 2.4.3.3.

Figure 5.7 shows the F_1-score for each dataset division and each LPPM. From the plot it is clear that the Adaptive* and VA-GI adapt to both the user and report velocities in accordance with the desirable properties of a velocity-aware LPPM, thus confirming the previous results. This is observable from the fact that for the "Favor Utility ($\uparrow v_u \downarrow v_r$)" division these two LPPMs present the highest F_1-scores, meaning that both LPPMs adjusted for utility, and the lowest F_1-scores for "Favor Privacy ($\downarrow v_u \uparrow v_r$)", signaling an adjustment for privacy. However, the VA-GI outperformed the Adaptive* in both cases, presenting higher score for the "Favor Utility ($\uparrow v_u \downarrow v_r$)" and lower for the "Favor Privacy ($\downarrow v_u \uparrow v_r$)", as desired. Furthermore, it should be noted that the displayed res-

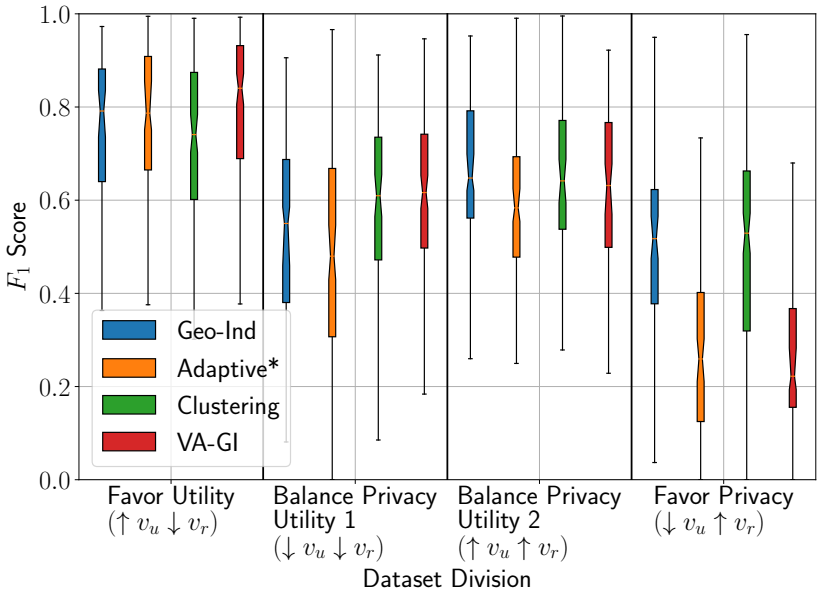


Figure 5.7.: Boxplot of the F_1-score for each dataset division and LPPM, with $\epsilon = 16$ km^{-1}.

ults for the adaptive mechanism were obtained with an improved selection of parameters, the Adaptive* - the default values would have lead to an ineffective privacy-utility adaptability, as depicted in Fig. 5.2. Figure 5.7 also shows that Geo-Ind and Clustering present similar yet smaller fluctuations in the F_1-score for the different dataset divisions. This is due to the underlying selection of trajectories for the division. Specifically, the selected traces with high user velocity correspond to movements in highways, in where there is less entropy in finding the right trajectory with the map-matching, thus resulting in a higher F_1-score. As for lower v_u trajectories, these correspond to alleys and residential areas, in where the density of the road network is higher and, therefore resulting in a lower F_1-score. Nevertheless, these fluctuations in the scores for the different divisions are inferior to the ones obtained with the Adaptive* and VA-GI, signaling that the latter two LPPMs effectively adapt to the velocities.

The variations in the F_1-score for the Adaptive* and VA-GI originate from the dynamic adaptability of the ϵ_i value according to the velocities as in equation (5.10). Therefore, it is useful to look at the distribution of these values to confirm the aforementioned findings. Figure 5.8 presents the distributions of the ϵ_i values for each dataset and LPPM, with $\epsilon = 16$ km^{-1}. Note that Geo-Ind and Clustering are a single scatter point as the ϵ is constant, while the Adaptive* presents three possible values as per equation (2.8). Results for VA-GI are presented as a boxplot, due to the continuous nature of the epsilon values obtained. These plots firmly agree with F_1-score results. Namely, both the Adaptive* and VA-GI adapt for privacy for the "Favor Privacy ($\downarrow v_u \uparrow v_r$)" by decreasing ϵ_i and for utility for the "Favor Utility ($\uparrow v_u \downarrow v_r$)" by increasing ϵ_i. Notice however, that while the VA-GI has continuous spectrum of values for ϵ_i, the Adaptive* mechanism considers only three values, resulting from the application of formula (2.8). Therefore, the VA-GI is able to provide a more fine

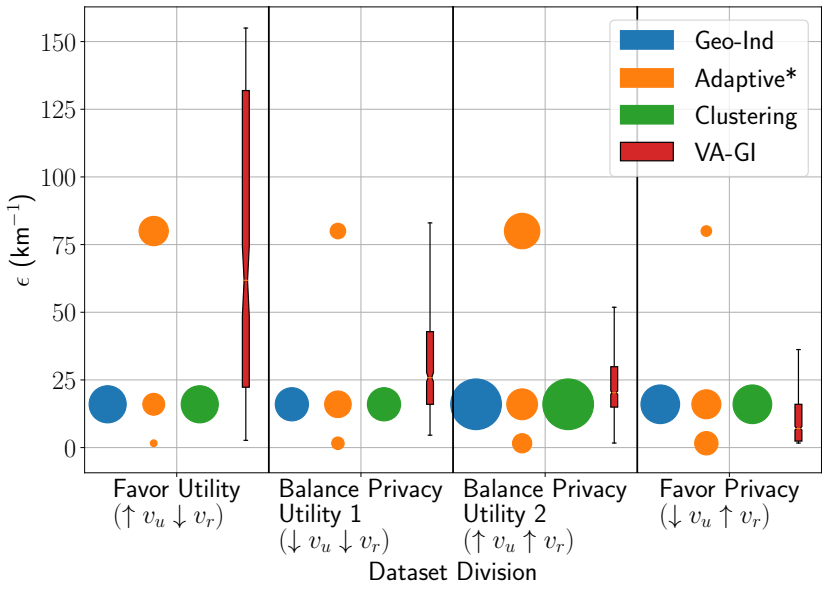


Figure 5.8.: Distribution of the ϵ_i values for each dataset division and LPPM, with $\epsilon = 16$ km^{-1} in the form of a scatter plot for Geo-Ind, Adaptive* and Clustering and a boxplot for VA-GI. The size of the scatter points is the absolute frequency of the corresponding ϵ value. Note that Geo-Ind and Clustering are a single scatter point as the ϵ is constant, while the Adaptive* presents three possible values as per equation (2.8). Results for VA-GI are presented as a boxplot, due to the continuous nature of the epsilon values obtained.

grained privacy/utility adaptability. This is relevant as the Adaptive* might erroneously not adapt for privacy or utility in cases where it should, which is further aggravated by possibly misconfigured threshold values Δ_1 and Δ_2 (c.f. Figure 5.2). The VA-GI mitigates this problem by using the CDF of the velocities for defining the system parameters, as previously discussed.

In summary, both the VA-GI and Adaptive* adapt in accordance to the desired properties of a velocity-aware LPPM. Specifically, an increase in the density of reports per distance traveled is met with higher obfuscation (i.e. lower ϵ_i) to improve privacy and a decrease in the density of reports is met with a smaller obfuscation (higher ϵ_i) as to increase data quality. For this dynamic adjustment, the VA-GI outperfoms all other tested LPPMs from both privacy and utility metrics. Additionally, and in contrast with the Adaptive, it provides finer grained and continuous adaptability, while requiring fewer parameters and therefore mitigating misconfiguration issues that can lead to no privacy [Mendes et al., 2020]. However, unlike the Geo-Ind and the Clustering LPPMs, the VA-GI LPPM requires data to estimate the CDFs of the velocities, which can limit the wide deployment of such mechanism. From the CDFs of the user and report velocities plotted in Figure 5.3, we observed that these might be approximated to a Gaussian distribution. Taking advantage of this fact, in the following section, we propose a methodology to generalize the VA-GI LPPM by using data from one location to model the CDF of velocities of another location.

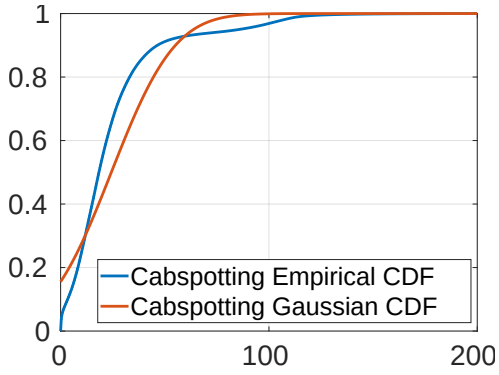
5.4 Generalizing the VA-GI LPPM

In order to use the CDF of the velocities for the definition of the function $f(\cdot)$ in (5.8), real mobility data is required, thus posing a limitation on the practicability of the VA-GI. However, from the plots of the CDFs for the Cabspotting data illustrated in Figure 5.3, we can observe that an approximation to a Gaussian distribution might fit as an estimation. In this section we do such evaluation, specifically, we use publicly available data to generate a CDF and then measure the fitness of this approximation to a new dataset. Without loss of generality, we focus on fitting the CDF of the user velocity, since the same methodology could be used to approximate the CDF of the report velocity, with the difference that different applications might use different sampling frequencies. To solve such dissimilarity, a normalization of the distribution would suffice.

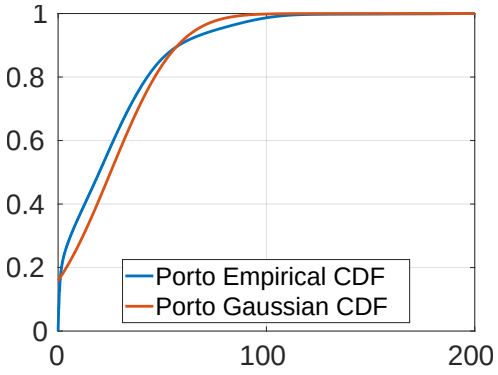
Vehicular velocities have been previously shown to follow Guassian distributions in highway scenarios [Boban et al., 2011]. However, urban traffic is more complex due to intersections, traffic lights and signals, congestions and other factors [Tonguz et al., 2009]. Therefore, in order to visually compare the goodness of fit of the Gaussian distribution, we use the Cabspotting dataset as publicly available data to form the CDFs and a second dataset from a different geographic location to evaluate the goodness of fit. This second dataset is also composed of vehicular trajectories belonging to 441 taxis in the city of Porto, Portugal, with a sampling rate of 15 seconds and collected over a full year [Moreira-Matias et al., 2013].

Figure 5.9 shows the obtained CDFs for a subsample of 10000 velocities from the Cabspotting and Porto datasets. From Figure 5.9a it can be seen that for the Cabspotting dataset, the distributions differ considerably. However, for the Porto dataset, Figure 5.9 reveals a high similarity between the empirical and Guassian distributions. A Kolmogorov-Smirnov normality test confirms that both velocity sets do not follow a Gaussian distribution for any confidence level (p-value is 0). Additionally, a two-sample Kolmogorov-Smirnov goodness-of-fit hypothesis test also discards the possibility of both velocities following the same distribution (p-value is also 0). Finally, a Wilcoxon rank sum test and a Mood's median test reject the hypothesis that both velocities have the same mean and median, respectively.

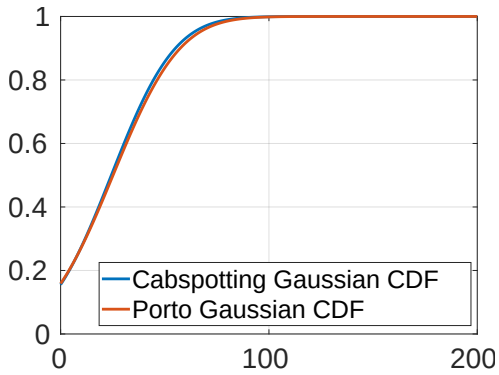
Despite the fact that the velocities follow an unknown seemingly multimodal distribution, the use of the Guassian distribution as an approximation might suffice for the purpose of the function of the user velocities $f(v_{u,i})$. We can see this effect by observing Figure 5.9c, where the Gaussian CDF for both datasets is similar, even though both cities have different speed limits, and hence different velocity distributions. In fact, in Figure 5.9d, the Gaussian CDF obtained from the Cabspotting dataset is similar to the Porto empirical CDF. This suggests that using publicly available data, even from different geo-locations might result in effective approximations. Note, however, that the training data must be diverse, as training with data from a rural area and applying it to a metropolitan area might result in poor results.



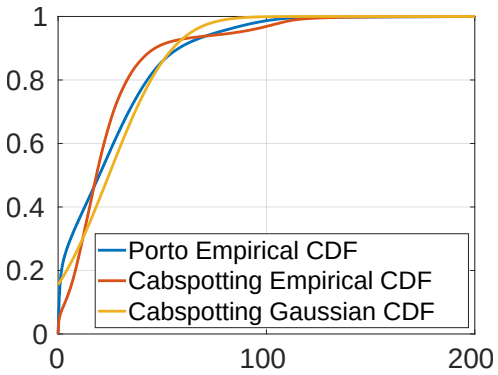
(a) Cabspotting Empirical and Gaussian CDFs.



(b) Porto Empirical and Gaussian CDFs



(c) Cabspotting and Porto Guassian CDFs.

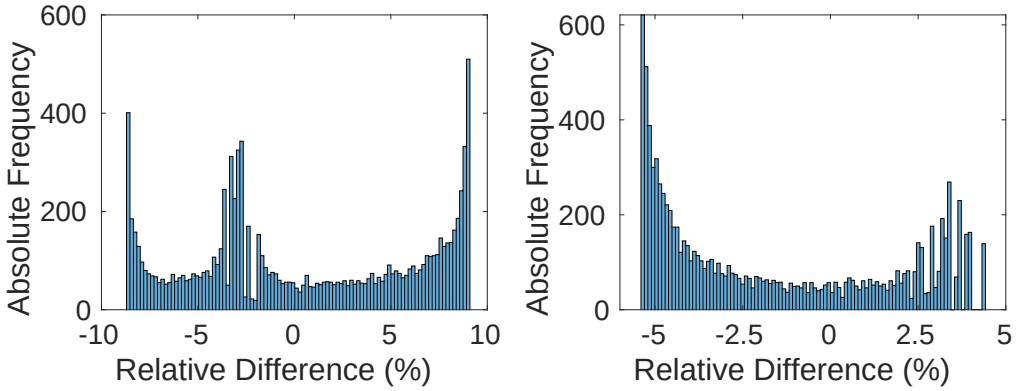


(d) Porto empirical CDF and Cabspotting Empirical and Gaussian CDFs.

Figure 5.9.: CDF and PDF for the user velocities of the Cabspotting and Porto datasets.

From the point of view of VA-GI, an approximation of the definition of $f(\cdot)$ will result in sub-optimal velocity awareness, which can cause the LPPM to overshoot the privacy or utility in certain cases. In order to measure this unbalance, we use a subsample of the velocities from the Porto dataset and plot the differences between the ϵ_i values obtained using the Cabspotting approximations and the Porto KDE CDF, which is the baseline reference. Recall that the KDE (Kernel Density Estimation) smoothly describes the empirical CDF, as illustrated in Figure 5.3. In other words, we use the Cabspotting dataset as public available data, from which we extract the KDE and Gaussian CDFs and then we apply these distributions in the form of equation (5.10) to a subsample of the Porto dataset. To focus on user velocity, without loss of generality, we set the value of $cdf(v_{r,i}) = k, \ \forall i$ with $k = 0.5$, such that equation (5.10) becomes $\epsilon_i = \epsilon.m^{(cdf(v_{u,i})-0.5)}$, with the bounds $\epsilon \cdot m^{-0.5} \leq \epsilon_i \leq \epsilon \cdot m^{0.5}, \quad \forall i$ and $m = 10$, as defined previously.

Figure 5.10a presents the differences between the epsilons obtained with the Cabspotting KDE and the Porto KDE, while Figure 5.10b shows the differences between the epsilons obtained with the Cabspotting Gaussian and the Porto KDE, with an initial ϵ value of $16km^{-1}$. A negative value in these plots corres-



(a) Cabspotting KDE and Porto KDE epsilon values

(b) Cabspotting Guassian and Porto KDE epsilon values

Figure 5.10.: Relative differences for the ϵ_i values between each pair of distributions, with initial $\epsilon = 16$.

ponds to an overshoot in the privacy adjustment, as the ϵ_i in the Porto KDE is higher than the same ϵ_i for the Cabspotting estimation, and vice-versa for a positive value, thus corresponding to an overshoot in the utility adjustment. From these plots we can clearly see that even though the velocity distributions might differ, the ϵ_i obtained using equation (5.10) are similar. Specifically, the ϵ_i values differed less than 10% when using the Cabspotting KDE and up to approximately 5% when using the Gaussian approximation, for both privacy and utility adjustments. Nevertheless, these results confirm that it is possible to use publicly available data, even from a different geo-location and even with different speed limits, to approximate the function $f(\cdot)$ of VA-GI. Furthermore, from a practitioner perspective, the Gaussian distribution as an estimator can be used in production by simply setting the mean and standard deviation parameters, thus giving a practical benefit over the KDE. Under these configurations, the user just has to provide the parameters ϵ and m as previously discussed, where m can be used to adjust VA-GI for more privacy or utility, as desired. By configuring two parameters alone, our proposed scheme enables adaptation of the privacy and utility levels according to the user and report velocities. To the best of our knowledge, our proposal is the first to consider adaptation of privacy and utility levels according to user and report velocity.

5.5 Limitations and Future work

The use of correlations between subsequent requests is an important venue for LPPMs. In this paper we have used the velocity of the user and the frequency of reports as metric for the correlation between reports and proposed particular instance of a VA-GI LPPM. While our results have shown an effective adaptability we leave for future work the comparison between different VA-GI formulations. Furthermore, the use of the velocity of the user and the frequency of reports might not be an efficient proxy for the correlation between reports in some cases. For instance, in a highway, the density of reports can be low, but the

correlation might be high due to the lack of intersections/exits. As future work, we intend to consider the underlying map in the design of the LPPM, similarly to the work in [Wang et al., 2015c] but in the context of differential privacy. For example, the consideration of the road-network in vehicular trajectories or the density of buildings, can allow for a metric on the adversary confusion, thus potentially resulting in a more effective privacy and utility adjustment.

For a practical implementation of VA-GI, one has to consider its impact from both the user and application perspective. Considering a service for the continuous scenario, VA-GI can incur in significant quality loss, for instance, due to the creation of spurious routes in a navigation service. In this regard, the system would need to consider filtering out the obfuscation on the side of the user, potentially through the incorporation of local and offline processing. To be more concrete, in the context of a navigation system, the routing system could use the true location of the user locally, while updating the necessary information with every obfuscated report (e.g. the traffic conditions and estimated time of arrival). These online updates would have a precision loss as a function of the obfuscation radius, which in turn depends on the privacy parameters, velocity of the user and frequency of reports.

Our scheme requires access to user velocity data that may not always be available. We have shown in Section 5.4 how to generalize VA-GI through approximation to known CDFs, whereby the Gaussian CDF generalized better than the KDE CDF. However, further validation using other datasets is required to confirm these results.

Inherited from differential privacy, the repeated use of any geo-indistinguishable LPPM, including VA-GI, results in increasing and unbounded information leakage, which can be measured through the composition properties Dwork [2008]. In practice, one can define a maximum privacy budget, such that after exhausting it, no more data is sent to the service provider. Unfortunately, this would result in not having access to the service. Practical implementations of differential privacy incur in a trade-off where the privacy budget is reset after a certain amount of time, thus limiting exposure within a given time frame, while (misleadingly) considering contributions between periods independent [Tang et al., 2017]. This is an active line of research that could spark future work in LPPMs at data collection.

Finally, location data has been considered personal data under privacy laws such as the General Data Protection Regulation (GDPR) [Union, 2016] and the California Consumer Privacy Act (CCPA) [of California, 2018]. VA-GI preserves location privacy even against the service provider, thus providing some degree of anonymity. In practice, however, LBSs require an account, thus identifying even obfuscated reports. Therefore, VA-GI provides location privacy, but not necessarily anonymity. Regardless, the legal requirement to anonymize the data lies on the service provider, which becomes the data owner/controller, and therefore, further anonymization might be required from the service provider before sharing/storing the data. Related, but from a reverse perspective, one should consider the impact of such obfuscation in existing business models and

data ecosystems. The degradation of the quality of data not only directly impacts the user due to the service quality loss, but also indirectly. For example, the reported location data might be further used for traffic analysis or for pay-as-you-go insurance companies, in turn resulting in potential suboptimal traffic routing and more expensive insurance, respectively.

5.6 Chapter Summary

The widespread of mobile and connected devices has lead to the pervasiveness of LBSs. While vast, the research on location privacy has fallen behind this development, specially in Location Privacy-Preserving Mechanisms (LPPMs) that act at collection time. In this context, for effective privacy protection, LPPMs must take into consideration the potential threat that arises from the correlation of reports. In this chapter we adopted the velocity of the user and the frequency of reports as metric for the correlation and proposed a generalization of Geo-Indistinguishability termed Velocity-Aware Geo-Indistinguishability (VA-GI). Under such notion, we design a VA-GI LPPM that according to our results, outperforms previous LPPMs in adapting the privacy and utility under different dynamic scenarios. The proposed LPPM adapts as a function of the user and report velocities, while requiring only two user-set parameters, thus mitigating misconfigurations that can lead to no effective privacy. This adaptability can be tuned for general use, by using city or country-wide data, or for specific user profiles, thus warranting fine-grained tuning for users. Finally, we generalized our proposed VA-GI LPPM by using publicly available data for defining system parameters, thus facilitating effective wide deployment. Results show that the generalization produces a privacy and utility variance that differs by at most 10% from the non-generalized counter-part.

In the context of permission managers, we argue that the proposed VA-GI LPPM could be effectively deployed towards the obfuscation of location data in mobile devices. Particularly, the frequency of reports could be measured either as the frequency of calls to the location APIs across all apps, thus granting protection even against potentially colluding providers at the expense of a more degraded data quality, or on a per-app basis. Regardless, the automatic adjustment avoids the otherwise mandatory constant tuning, for instance in the form of a prompt, that would result in significant user burden. Furthermore, the automatic adaptability could additionally be personalized to the specific user, as the driver velocities could be used to form the CDF, or to specific regions, by using data from drivers in these areas.

We leave for future work the integration of this LPPM in a permission manager and the respective evaluation, which requires a field-study under real world conditions. Related, the consideration of multiple means of transportation could potentially warrant the need for multiple configurations of the LPPM, such that the adaptability is properly tuned. Furthermore, either the user would have to input the means of transportation, or an automatic detection would be required, which could be a challenge on itself.

Chapter 6.

Conclusions and Future Work

Contents

Advances in mobile devices and communication infrastructure fostered the proliferation of applications capable of providing localized and user-tailored services. The benefits of these technologies are undeniable as society strongly embraces its presence in the daily lives. However, the drawback of this paradigm, which is not always perceived by individuals, is the digital trail that continuously grows as companies harvest the necessary, or even the unnecessary, data to provide the referred services. Despite advances in privacy regulations, such as the General Data Protection Regulation (GDPR), after being collected, users have limited control over their data. In fact, most control is waived under complicated and often unread privacy policies or service terms and conditions.

To empower users with control over their data, mobile devices have implemented permission managers. This mechanism allows users to control application access to sensitive data, thus protecting privacy before the data is sent to the service providers. Unfortunately, in the context of mobile devices the highly dynamic environment and continuously growing services has led to a trade-off that focus on usability in detriment of privacy, as to simplify configurations and alleviate user-burden. However, in this thesis we argue and prove that it is possible to improve both usability and privacy control through automation, personalization, context-awareness and obfuscation. This chapter summarizes this thesis, respective contributions and present future work remarks.

6.1 Synthesis of the Thesis

This thesis focuses on improving privacy in mobile devices, while keeping or even increasing the usability of privacy-preserving mechanisms. Towards this challenging objective we strongly focused on automation, for either privacy enforcement or for auto-configuration of the privacy-preserving mechanisms. Such automation was supported through personalization and/or context-awareness, as to account for privacy subjectiveness and context dependency.

The focus of this thesis was motivated in Chapter 2, where limitations on existing privacy mechanisms for mobile devices are identified. Of particular relevance, we highlight the amount of automatically allowed permissions that defy contextual-integrity and user expectation, the lack of personalization and the poor controls over the privacy-utility trade-off. To tackle this latter aspect, we focus on obfuscation to increase the control and note that such techniques are data type specific.

In the road to automated privacy protection, we started with a field study to collect privacy decisions and user expectations in-situ in Chapter 3. The collected data served as motivation for enhancing privacy managers, as almost half of requests are unexpected and the default permission manager incurred in privacy violations 15% of times. In fact, our analysis revealed a strong,

yet subjective impact of user expectations in privacy decisions, and uncovered an intrinsic relation between application usage, and therefore the permission requests that are made to the user, and the surrounding context.

With the collected data, we developed predictive models to automate privacy decisions, which we also present in Chapter 3. These models build up on previous works by using privacy profiles towards personalization, and by considering contextual features towards context-awareness. We innovate by incorporating user expectations and by considering contextually-aware privacy profiles. Our automated, personalized and context-aware approach is able to reduce the amount of privacy violations by almost 60% when compared to the Android permission manager based on runtime permissions. Without using the expectation as input feature for the prediction, as this requires getting input from the user, these violations can still be reduced by 28% under our approach.

In an effort to increase user control over the trade-off between privacy and utility in the permission manager, we focused on obfuscation. The advantage of obfuscation is that it can be effectively applied at data collection, thus preserving privacy even against a potentially untrustworthy service provider. Unfortunately, this type of privacy-preserving mechanisms is data type dependent and we therefore, focused on location data, a prevalent and critical data type in mobile devices.

Location data has particular privacy considerations which steam from the nature of this type of data. Specifically, the frequency of reports directly impacts the temporal correlations between user locations. In Chapter 4 we evaluate the impact of the frequency of reports on the privacy level of location traces, with a particular focus on geo-indistinguishability, a differentially-private formal notion. We have empirically validated the consideration of independence between reports under sporadic release of location updates, while reaffirming the need for better mechanisms under continuous reports. Additionally, we show that a misconfigured privacy parameter can result in no effective privacy. These two main findings from Chapter 4 served as motivation to a novel Location Privacy-Preserving Mechanism (LPPM) termed Velocity-Aware Geo-Indistinguishability (VA-GI) which we propose in Chapter 5.

VA-GI is a generalization of geo-indistinguishability to provide automatic privacy and utility adjustment, while minimizing the amount of user-set parameters that can result in misconfigurations. VA-GI thus provides geo-indistinguishability with a variable privacy budget, that is automatically adjusted based on situation/context of the user, as measured by its velocity and frequency of location reports. It can additionally be personalized to specific drivers or regions/road networks, and effectively generalized for wide deployment. These usability properties are crucial towards the implementation of VA-GI in a permission manager as to minimize user-burden and to mitigate potential misconfigurations. Our empirical results prove the desired adaptation of VA-GI to both privacy and utility, in fact outperforming previous geo-indistinguishable LPPMs.

6.2 Contributions

This research has made several contributions that were enumerated in the previous chapters. In this section we highlight the main contributions:

- **A dataset of over 65000 user-answered privacy decisions and the respective user expectations and surrounding context.** This dataset was collected from 93 participants and we make it available to the research community [Mendes, 2021a], alongside our data collection tool [Mendes, 2021b]. This dataset served as motivation for this work, as it showed how current permission managers are insufficient to effectively control privacy. Specifically, a runtime permission manager such as the one used in Android 9 would have in incurred in a violation of privacy in 15% of all user-answered permissions, that is, it would have allowed permissions that our users explicitly denied. Furthermore, it uncovered a strong misalignment between apps practices and user expectations, as almost 50% of permission requests were unexpected.
- **An automated, personalized and context-aware permission manager that achieves a ROC AUC of 0.96 and an F1 score of 0.92.** This mechanism reduces the amount of privacy violations by 60% when compared to the standard android permission manager. However, this value is only possible when knowing the user expectation regarding each permission request. Without this piece of information, we are still able to reduce the number of privacy violations by 28%, with a prediction model that achieves a ROC AUC of 0.9 and an F1 score of 0.88.
- **An empirical analysis that evidences the need for the minimization of the number of user-set parameters in LPPMs and for their automatic and dynamic adjustment in accordance with varying frequency of updates.** Our experiments evidence how a misconfigured parameter can result in no effective privacy and how a strong attacker can potentially defeat obfuscation.
- **We propose VA-GI, a generalization of geo-indistinguishability to location traces that allows for the automatic adjustment of the privacy budget in accordance with varying velocity of the user and frequency of reports, while reducing the amount of required user-set parameters.** Our experiments show how a VA-GI LPPM outperforms previous geo-indistinguishable LPPMs in the privacy-utility adaptability, and how it can be personalized to specific drivers or areas, or generalized for wide deployment.

These proposals, as well as the literature review performed, are published in five international conference papers and one journal article. Additionally, they have sparked further research that was supervised by the candidate and culminated in two masters' thesis and four cooperation papers.

6.3 Future Work

Each chapter in this thesis has provided future work remarks, contextualized by the respective detailed subject. In this section we focus only on the main future research paths.

A possible improvement to automated privacy enforcement is to incorporate a richer contextual model, one that could semantically describe the user and device situation. Examples include inferring the semantic location of the user, the activity being practiced and the people in vicinity. Towards achieving this goal, additional data sources could be considered, such as activity recognition models that are based on motion sensor data. In this subject, it would be interesting to evaluate which situations warrant changes in privacy preferences, as the consideration of an ever increasing number of contexts also increases the amount of data required to train the predictive model.

Collecting data regarding permission decisions and application usage towards building automated privacy mechanisms, as described in this thesis, also raises privacy issues. In a master thesis supervised by the candidate it was explored the possibility of creating the privacy profiles and prediction models while preserving privacy even against the entity that performs this task. An unexplored venue in that line of work would be to develop a monitoring framework that would be able to update both the profiles and predictions models in real time, while maintaining the same privacy guarantees.

Our collected data revealed a strong misalignment between apps' practices and user expectations as nearly half of requests were unexpected. This value motivates the need for raising privacy awareness, potentially through better privacy indicators, such as, the camera and microphone in-use indicators; privacy nudges, that not only incentivize reviewing permission but also educate on apps' practices; and educational applications that can convey the risks using examples from installed applications.

In the context of location privacy, the VA-GI proposal uses the velocity of the user and frequency of reports as a metric for the correlation between user locations, which might be inaccurate by not considering the map of possible locations. A potential enhancement is to use the underlying map as a metric of entropy or confusion of the adversary in pin-pointing the user-location.

Finally, a natural departure of this work would be the incorporation of obfuscation in the permission manager, such as using the VA-GI for location data and similar techniques for the other data types. Two different evaluation studies would be required in this line of work. The first would be the measurement of the adoption and user perceived utility of the obfuscation techniques. The second would be to evaluate the performance w.r.t. privacy and utility, of the techniques, including their adaptability to different contexts, such as, in the case of location data, varying frequency of location updates. Similar methodology to the one taken in this thesis could be conducted for other data types.

Bibliography

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer.

Acquisti, A., Brandimarte, L., and Loewenstein, G. (2015). Privacy and human behavior in the age of information. *Science*, 347(6221):509–515.

Acquisti, A. and Grossklags, J. (2008). What can behavioral economics teach us about privacy. *Digital Privacy: Theory, Technologies, and Practices*.

Al-Dhubhani, R. and Cazalas, J. M. (2018). An adaptive geo-indistinguishability mechanism for continuous lbs queries. *Wireless Networks*, 24(8):3221–3239.

Alegre, U., Augusto, J. C., and Clark, T. (2016). Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, 117:55–83.

Almuhimedi, H., Schaub, F., Sadeh, N., Adjerid, I., Acquisti, A., Gluck, J., Cranor, L. F., and Agarwal, Y. (2015). Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 787–796. ACM.

Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., and Palamidessi, C. (2013). Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, page 901–914, New York, NY, USA. Association for Computing Machinery.

Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., and Palamidessi, C. (2013). Geo-indistinguishability: Differential Privacy for Location-based Systems. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pages 901–914.

Andriotis, P., Stringhini, G., and Sasse, M. A. (2018a). Studying users' adaptation to Android's run-time fine-grained access control system. *Journal of Information Security and Applications*, 40:31–43.

Andriotis, P., Stringhini, G., and Sasse, M. A. (2018b). Studying users' adaptation to android's run-time fine-grained access control system. *Journal of Information Security and Applications*, 40:31–43.

Android Developers (2018). Distribution dashboard. https://developer.android.com/about/dashboards/. Accessed: 2022-01-07.

Android Developers (2019). Privacy changes in android 10. https://developer.android.com/about/versions/10/privacy/changes#non-resettable-device-ids. Accessed: 2022-01-07.

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*, pages 1–11.

Banisar, D., Davies, S., et al. (1999). Privacy and human rights: an international survey of privacy laws and practice. *Global Internet Liberty Campaign*.

Beresford, A. R., Rice, A., and Skehin, N. (2011). MockDroid : trading privacy for application functionality on smartphones Categories and Subject Descriptors. *HotMobile*, pages 49–54.

Beresford, A. R. and Stajano, F. (2003). Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55.

Bertino, E., Lin, D., and Jiang, W. (2008). A survey of quantification of privacy preserving data mining algorithms. In *Privacy-preserving data mining*, pages 183–205. Springer.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180.

Bettini, C., Wang, X. S., and Jajodia, S. (2005). Protecting privacy against location-based personal identification. In *Workshop on Secure Data Management*, pages 185–199. Springer.

Bifet, A. and Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM.

Boban, M., Vinhoza, T. T. V., Ferreira, M., Barros, J., and Tonguz, O. K. (2011). Impact of vehicles as obstacles in vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(1):15–28.

Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.

Bokhorst, M. (2013). Xprivacy - the ultimate, yet easy to use, privacy manager. https://forum.xda-developers.com/t/xprivacy-the-ultimate-yet-easy-to-use-privacy-manager.2320783/. Accessed: 2022-01-07.

Bonné, B., Peddinti, S. T., Bilogrevic, I., and Taft, N. (2017). Exploring decision making with android's runtime permission dialogs using in-context surveys. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 195–210, Santa Clara, CA. USENIX Association.

Brandão, A., Mendes, R., and Vilela, J. P. (2021). Efficient Privacy Preserving Distributed K-Means for Non-IID Data. In *Advances in Intelligent Data Analysis XIX*, pages 439–451, Cham. Springer International Publishing.

Brandão, A., Mendes, R., and Vilela, J. P. (2022). Prediction of mobile app privacy preferences with user profiles via federated learning. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*. ACM. In press.

Brandão, A. (2021). Prediction of privacy preferences with user profiles: A federated learning approach. Master's thesis, Universidade do Porto.

Calciati, P., Kuznetsov, K., Gorla, A., and Zeller, A. (2020). Automatically granted permissions in android apps: An empirical study on their prevalence and on the potential threats for privacy. In *Proceedings of the 17th International Conference on Mining Software Repositories*, MSR '20, page 114–124, New York, NY, USA. Association for Computing Machinery.

Chatzikokolakis, K., Elsalamouny, E., and Palamidessi, C. (2017). Efficient utility improvement for location privacy. *Proceedings on Privacy Enhancing Technologies*, 2017(4):308–328.

Chatzikokolakis, K., Palamidessi, C., and Stronati, M. (2014). A predictive differentially-private mechanism for mobility traces. In De Cristofaro, E. and Murdoch, S. J., editors, *Privacy Enhancing Technologies*, pages 21–41. Springer International Publishing.

Chitkara, S., Gothoskar, N., Harish, S., Hong, J. I., and Agarwal, Y. (2017). Does this app really need my location?: Context-aware privacy management for smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):42.

Chow, C.-Y., Mokbel, M. F., and Liu, X. (2011). Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *GeoInformatica*, 15(2):351–380.

Clifton, C. and Tassa, T. (2013). On syntactic anonymity and differential privacy. In *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pages 88–93. IEEE, IEEE.

Conti, M., Nguyen, V. T. N., and Crispo, B. (2010). Crepe: Context-related policy enforcement for android. In *International Conference on Information Security*, pages 331–345. Springer.

Cranor, L., Rabin, T., Shmatikov, V., Vadhan, S., and Weitzner, D. (2015). *Towards a Privacy Research Roadmap for the Computing Community*: A

white paper prepared for the computing community consortium committee of the computing research association.

Cunha, M. (2019). Privacy-preserving mechanisms for location traces. Master's thesis, Universidade de Coimbra.

Cunha, M., Mendes, R., and Vilela, J. P. (2019). Clustering geo-indistinguishability for privacy of continuous location traces. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, pages 1–8. IEEE.

Cunha, M., Mendes, R., and Vilela, J. P. (2021). A survey of privacy-preserving mechanisms for heterogeneous data types. *Computer Science Review*, 41:100403.

Das, P. K., Joshi, A., and Finin, T. (2016). Capturing policies for fine-grained access control on mobile devices. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pages 54–63.

De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376.

Developers, G. (2022a). Privacy changes in android 10: Identifiers and data. `https://developer.android.com/about/versions/10/privacy/changes`. Accessed: 2021-09-09.

Developers, G. (2022b). Request app permissions. `https://developer.android.com/training/permissions/requesting`. Accessed: 2021-08-29.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dwork, C. (2008). Differential privacy: A survey of results. In Agrawal, M., Du, D., Duan, Z., and Li, A., editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg. Springer Berlin Heidelberg.

Elbitar, Y., Schilling, M., Nguyen, T. T., Backes, M., and Bugiel, S. (2021). Explanation beats context: The effect of timing & rationales on users' runtime permission decisions. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 785–802.

ElderDrivers (2020). Edxposed framework. `https://github.com/ElderDrivers/EdXposed`. Accessed: 2021-08-29.

Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., and Sheth, A. N. (2014). Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 32(2):5.

Felt, A. P., Ha, E., Egelman, S., Haney, A., Chin, E., and Wagner, D. (2012). Android permissions: User attention, comprehension, and behavior. In *Proceedings of the eighth symposium on usable privacy and security*, page 3. ACM.

Feng, Y., Yao, Y., and Sadeh, N. (2021). A design space for privacy choices: Towards meaningful privacy control in the internet of things. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA. Association for Computing Machinery.

Fu, H., Zheng, Z., Zhu, S., and Mohapatra, P. (2019). Keeping context in mind: Automating mobile app access control with user interface inspection. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2089–2097. IEEE.

Gama, J., Medas, P., Castillo, G., and Rodrigues, P. (2004). Learning with drift detection. In *Brazilian symposium on artificial intelligence*, pages 286–295. Springer.

Gambs, S., Killijian, M.-O., and del Prado Cortez, M. N. (2010). Show Me How You Move and I Will Tell You Who You Are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, page 34–41, New York, NY, USA. Association for Computing Machinery.

Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., and Tan, K.-L. (2008). Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM.

Gluck, J., Schaub, F., Friedman, A., Habib, H., Sadeh, N., Cranor, L. F., and Agarwal, Y. (2016). How short is too short? implications of length and framing on the effectiveness of privacy notices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 321–340.

Goh, C. Y., Dauwels, J., Mitrovic, N., Asif, M. T., Oran, A., and Jaillet, P. (2012). Online map-matching based on hidden markov model for real-time traffic sensing applications. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 776–781. IEEE.

Google (2018). Use of sms or call log permission groups. `https://support.google.com/googleplay/android-developer/answer/10208820`. Accessed: 2022-01-07.

Google Developers (2020). Meet google play's target api level requirement. `https://developer.android.com/distribute/best-practices/develop/target-sdk`. Accessed: 2021-09-21.

Gruteser, M. and Grunwald, D. (2003). Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM.

Harbach, M., Hettig, M., Weber, S., and Smith, M. (2014). Using personal examples to improve risk communication for security & privacy decisions. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2647–2656. ACM.

Hashemi, M. and Karimi, H. A. (2014). A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*, 48:153–165.

Hoh, B., Gruteser, M., Xiong, H., and Alrabady, A. (2010). Achieving guaranteed anonymity in gps traces via uncertainty-aware path cloaking. *IEEE Transactions on Mobile Computing*, 9(8):1089–1107.

Holly, R. (2015). Cyanogen OS Privacy Guard – keeping apps from seeing your data. `https://www.androidcentral.com/cyanogen-os-privacy-guard-keeping-apps-seeing-your-data`. Accessed: 2022-01-07.

Hornyack, P., Han, S., Jung, J., Schechter, S., and Wetherall, D. (2011). These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 639–652. ACM.

Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., and Roth, A. (2014). Differential privacy: An economic method for choosing epsilon. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 398–410. IEEE.

Huang, H., Gartner, G., Krisp, J. M., Raubal, M., and Van de Weghe, N. (2018). Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, 12(2):63–93.

Jagadeesh, G. R. and Srikanthan, T. (2017). Online map-matching of noisy and sparse location data with hidden markov and route choice models. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2423–2434.

Kaaniche, N., Laurent, M., and Belguith, S. (2020). Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey. *Journal of Network and Computer Applications*, 171:102807.

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 *International Conference on Financial Cryptography and Data Security*, pages 68–79. Springer.

Krumm, J. (2007). Inference attacks on location tracks. In LaMarca, A., Langheinrich, M., and Truong, K. N., editors, *Pervasive Computing*, pages 127–143, Berlin, Heidelberg. Springer Berlin Heidelberg.

Krumm, J. (2009). A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399.

Kubicka, M., Cela, A., Mounier, H., and Niculescu, S.-I. (2018). Comparative Study and Application-Oriented Classification of Vehicular Map-Matching Methods. *IEEE Intelligent Transportation Systems Magazine*, 10(2):150–166.

Langheinrich, M. (2009). Privacy in ubiquitous computing. In *Ubiquitous Computing Fundamentals*, chapter 3, pages 95–159. CRC Press.

Lee, J. and Clifton, C. (2011). How much is enough? choosing ϵ for differential privacy. In Lai, X., Zhou, J., and Li, H., editors, *Information Security*, pages 325–340, Berlin, Heidelberg. Springer Berlin Heidelberg.

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*, pages 501–510.

Lin, J., Liu, B., Sadeh, N., and Hong, J. I. (2014). Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 199–212, Menlo Park, CA. USENIX Association.

Liu, B., Andersen, M. S., Schaub, F., Almuhimedi, H., Zhang, S. A., Sadeh, N., Agarwal, Y., and Acquisti, A. (2016). Follow my recommendations: A personalized privacy assistant for mobile app permissions. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 27–41.

Liu, B., Lin, J., and Sadeh, N. (2014). Reconciling mobile app privacy and usability on smartphones: could user privacy profiles help? In *Proceedings of the 23rd international conference on World wide web - WWW '14*, pages 201–212.

Liu, B., Zhou, W., Zhu, T., Gao, L., and Xiang, Y. (2018a). Location Privacy and Its Applications: A Systematic Study. *IEEE Access*, 6:17606–17624.

Liu, H., Li, X., Li, H., Ma, J., and Ma, X. (2017). Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE.

Liu, X., Leng, Y., Yang, W., Wang, W., Zhai, C., and Xie, T. (2018b). A large-scale empirical study on android runtime-permission rationale messages. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 137–146. IEEE.

Mendes, R. (2021a). COP-MODE Dataset Guide. `https://cop-mode.dei.uc.pt/dataset`. Accessed: 2022-01-20.

Mendes, R. (2021b). COP-MODE Naive Permission Manager. `https://cop-mode.dei.uc.pt/cm-npm`. Accessed: 2021-08-29.

Mendes, R., Brandão, A., Vilela, J. P., and Beresford, A. R. (2022a). Effect of user expectation on mobile app privacy: A field study. In *2022 IEEE international conference on pervasive computing and communications (PerCom)*, pages 207–214. IEEE.

Mendes, R., Cunha, M., and Vilela, J. P. (2020). Impact of frequency of location reports on the privacy level of geo-indistinguishability. *Proceedings on Privacy Enhancing Technologies*, 2020(2):379 – 396.

Mendes, R., Cunha, M., and Vilela, J. P. (2023). Velocity-aware geo-indistinguishability. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY)*. ACM. In press.

Mendes, R., Cunha, M., Vilela, J. P., and Beresford, A. R. (2022b). Enhancing user privacy in mobile devices through prediction of privacy preferences. In *Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, pages 153–172. Springer.

Mendes, R. and Vilela, J. P. (2017). Privacy-preserving data mining: Methods, metrics, and applications. *IEEE Access*, 5:10562–10582.

Mendes, R. and Vilela, J. P. (2018). On the effect of update frequency on geo-indistinguishability of mobility traces. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '18, page 271–276, New York, NY, USA. Association for Computing Machinery.

Meyerowitz, J. and Roy Choudhury, R. (2009). Hiding stars with fireworks: location privacy through camouflage. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 345–356. ACM.

Moreira-Matias, L., Ferreira, M., and Mendes-Moreira, J. (2015). Taxi Service Trajectory Prediction Challenge @ ECML PKDD 2015. `https://github.com/achm6174/ecmlpkdd2015-challenge`. [Online; Accessed: 2022-03-20].

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., and Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402.

Murakami, T. (2017). Expectation-Maximization Tensor Factorization for Practical Location Privacy Attacks. *Proceedings on Privacy Enhancing Technologies*, 2017(4):138–155.

Murakami, T. and Watanabe, H. (2016). Localization attacks using matrix and tensor factorization. *IEEE Transactions on Information Forensics and Security*, 11(8):1647–1660.

Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE.

Nauman, M., Khan, S., and Zhang, X. (2010). Apex: extending Android permission model and enforcement with user-defined runtime constraints. *Asiaccs*, pages 328–332.

Neisse, R., Steri, G., Geneiatakis, D., and Fovino, I. N. (2016). A privacy enforcing framework for android applications. *computers & security*, 62:257–277.

Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, page 336–343, New York, NY, USA. Association for Computing Machinery.

Nissenbaum, H. (2004). Privacy as contextual integrity. *Wash. L. Rev.*, 79:119.

Norberg, P. A., Horne, D. R., and Horne, D. A. (2007). The privacy paradox: Personal information disclosure intentions versus behaviors. *Journal of consumer affairs*, 41(1):100–126.

of California, S. (2018). The california consumer privacy act of 2018. *California Civil Code*, Assembly Bill No. 375:1–24.

Olejnik, K., Dacosta, I., Machado, J. S., Huguenin, K., Khan, M. E., and Hubaux, J.-P. (2017). Smarper: Context-aware and automatic runtime-permissions for mobile devices. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 1058–1076. IEEE.

Oya, S., Troncoso, C., and Pérez-González, F. (2017a). Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1959–1972. ACM.

Oya, S., Troncoso, C., and Pérez-González, F. (2017b). Is geo-indistinguishability what you are looking for? In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, WPES '17, page 137–140, New York, NY, USA. Association for Computing Machinery.

Oya, S., Troncoso, C., and Pérez-González, F. (2019). Rethinking location privacy for unknown mobility behaviors. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 416–431.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454.

Petovello, M. (2015). How does a gnss receiver estimate velocity? *Inside GNSS*, pages 38–41.

Piorkowski, M., Sarafijanovic-Djukic, N., and Grossglauser, M. (2009). CRAWDAD dataset epfl/mobility (v. 2009-02-24). https://crawdad.org/epfl/mobility/20090224. [Online; Accessed: 2019-12-12].

Piorkowski, M., Sarafijanovic-Djukic, N., and Grossglauser, M. (2009). A parsimonious model of mobile partitioned networks with clustering. In *2009 First International Communication Systems and Networks and Workshops*, pages 1–10. IEEE.

Primault, V., Ben Mokhtar, S., Lauradoux, C., and Brunie, L. (2014). Differentially Private Location Privacy in Practice. In *Third Workshop on Mobile Security Technologies (MoST) 2014*, San Jose, United States. IEEE.

Primault, V., Boutet, A., Mokhtar, S. B., and Brunie, L. (2019). The long road to computational location privacy: A survey. *IEEE Communications Surveys Tutorials*, 21(3):2772–2793.

Rakowski, B. (2015). Get ready for the sweet taste of android 6.0 marshmallow. https://android.googleblog.com/2015/10/get-ready-for-sweet-taste-of-android-60.html. Accessed: 2022-01-07.

Ravichandran, R., Benisch, M., Kelley, P. G., and Sadeh, N. M. (2009). Capturing social networking privacy preferences. In *International symposium on privacy enhancing technologies symposium*, pages 1–18. Springer.

Reinfelder, L., Schankin, A., Russ, S., and Benenson, Z. (2018). An inquiry into perception and usage of smartphone permission models. In *International Conference on Trust and Privacy in Digital Business*, pages 9–22. Springer.

rovo89 (2012a). Xposed Development Tutorial. https://github.com/rovo89/XposedBridge/wiki/Development-tutorial. Accessed: 2021-08-29.

rovo89 (2012b). Xposed Installer – Xposed Module Repository. https://repo.xposed.info/module/de.robv.android.xposed.installer. Accessed: 2021-08-29.

rovo89 (2018). [OFFICIAL] Xposed for Lollipop/Marshmallow/Nougat/Oreo [v90-beta3, 2018/01/29]. https://forum.xda-developers.com/t/official-xposed-for-lollipop-marshmallow-nougat-oreo-v90-beta3-2018-01-29.3034811/. Accessed: 2021-08-29.

Sanchez, O. R., Torre, I., He, Y., and Knijnenburg, B. P. (2020). A recommendation approach for user privacy preferences in the fitness domain. *User Modeling and User-Adapted Interaction*, 30(3):513–565.

Schaub, F., Balebako, R., Durity, A. L., and Cranor, L. F. (2015a). A design space for effective privacy notices. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 1–17. USENIX Association.

Schaub, F., Könings, B., and Weber, M. (2015b). Context-adaptive privacy: Leveraging context awareness to support privacy decision making. *IEEE Pervasive Computing*, 14(1):34–43.

Schilit, B., Adams, N., and Want, R. (1994). Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE.

Schilit, B. N. and Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *IEEE network*, 8(5):22–32.

Shebaro, B., Oluwatimi, O., and Bertino, E. (2015). Context-based access control systems for mobile devices. *IEEE Transactions on Dependable and Secure Computing*, 12(2):150–163.

Shen, B., Wei, L., Xiang, C., Wu, Y., Shen, M., Zhou, Y., and Jin, X. (2021). Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model. In *30th USENIX Security Symposium (USENIX Security 21)*.

Shih, F., Liccardi, I., and Weitzner, D. (2015). Privacy tipping points in smartphones privacy preferences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 807–816.

Shklovski, I., Mainwaring, S. D., Skúladóttir, H. H., and Borgthorsson, H. (2014). Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2347–2356. ACM.

Shokri, R. (2015). Privacy games: Optimal user-centric data obfuscation. *Proceedings on Privacy Enhancing Technologies*, 2015(2):299–315.

Shokri, R., Theodorakopoulos, G., Danezis, G., Hubaux, J.-P., and Le Boudec, J.-Y. (2011). Quantifying location privacy: the case of sporadic location exposure. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 57–76. Springer.

Shokri, R., Theodorakopoulos, G., Le Boudec, J., and Hubaux, J. (2011). Quantifying location privacy. In *2011 IEEE Symposium on Security and Privacy*, pages 247–262. IEEE.

Shokri, R., Theodorakopoulos, G., Papadimitratos, P., Kazemi, E., and Hubaux, J. P. (2014). Hiding in the mobile crowd: Location privacy through collaboration. *IEEE Transactions on Dependable and Secure Computing*, 11(3):266–279.

Shokri, R., Theodorakopoulos, G., and Troncoso, C. (2017). Privacy games along location traces: A game-theoretic framework for optimizing location privacy. *ACM Transactions on Privacy and Security (TOPS)*, 19(4):11.

Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J.-P., and Le Boudec, J.-Y. (2012). Protecting location privacy: Optimal strategy against localization attacks. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 617–627, New York, NY, USA. Association for Computing Machinery.

Smullen, D., Feng, Y., Zhang, S. A., and Sadeh, N. (2020). The best of both worlds: Mitigating trade-offs between accuracy and user burden in capturing mobile app privacy preferences. *Proceedings on Privacy Enhancing Technologies*, 1:1–21.

Song, C., Qu, Z., Blumm, N., and Barabási, A.-L. (2010). Limits of predictability in human mobility. *Science*, 327(5968):1018–1021.

Song, Y., Dahlmeier, D., and Bressan, S. (2014). Not So Unique in the Crowd: a Simple and Effective Algorithm for Anonymizing Location Data. In *PIR@ SIGIR*, pages 19–24.

StatCounter (2022). Mobile Operating System Market Share Worldwide. `https://gs.statcounter.com/os-market-share/mobile/worldwide`. Accessed: 2022-03-20.

Strang, T. and Linnhoff-Popien, C. (2004). A context modeling survey. In *Workshop Proceedings*.

Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 3(3):34–36.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570.

Tan, J., Nguyen, K., Theodorides, M., Negrón-Arroyo, H., Thompson, C., Egelman, S., and Wagner, D. (2014). The effect of developer-specified explanations for permission requests on smartphone user behavior. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pages 91–100.

Tang, J., Korolova, A., Bai, X., Wang, X., and Wang, X. (2017). Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*.

Tonguz, O. K., Viriyasitavat, W., and Bai, F. (2009). Modeling urban traffic: A cellular automata approach. *IEEE Communications Magazine*, 47(5):142–150.

Tsai, L., Wijesekera, P., Reardon, J., Reyes, I., Egelman, S., Wagner, D., Good, N., and Chen, J.-W. (2017). Turtle guard: Helping android users apply contextual privacy preferences. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 145–162, Santa Clara, CA. USENIX Association.

Tsoukaneri, G., Theodorakopoulos, G., Leather, H., and Marina, M. K. (2016). On the inference of user paths from anonymized mobility data. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 199–213. IEEE.

Union, E. (2016). Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *Official Journal L110*, 59:1–88.

United Nation General Assembly (1948). Universal Declaration of Human Rights.

Van Kleek, M., Liccardi, I., Binns, R., Zhao, J., Weitzner, D. J., and Shadbolt, N. (2017). Better the devil you know: Exposing the data sharing practices of smartphone apps. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5208–5220. ACM.

Wagner, D. T., Rice, A., and Beresford, A. R. (2013). Device analyzer: Understanding smartphone usage. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 195–208. Springer.

Wang, W., Pan, L., Yuan, N., Zhang, S., and Liu, D. (2015a). A comparative analysis of intra-city human mobility by taxi. *Physica A: Statistical Mechanics and its Applications*, 420:134–147.

Wang, X., Sun, K., Wang, Y., and Jing, J. (2015b). DeepDroid: Dynamically Enforcing Enterprise Policy on Android Devices. *Symposium on Network and Distributed System Security (NDSS)*, (February):8–11.

Wang, Y., Xia, Y., Hou, J., meng Gao, S., Nie, X., and Wang, Q. (2015c). A fast privacy-preserving framework for continuous location-based queries in road networks. *Journal of Network and Computer Applications*, 53:57–73.

Weiser, M. (1999). The computer for the 21st century. *Mobile Computing and Communications Review*, 3(3):3–11.

Wernke, M., Skvortsov, P., Dürr, F., and Rothermel, K. (2014). A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175.

Westin, A. F. (1968). Privacy and freedom. *Washington and Lee Law Review*, 25(1):166.

Wijesekera, P., Baokar, A., Hosseini, A., Egelman, S., Wagner, D., and Beznosov, K. (2015). Android Permissions Remystified: A Field Study on Contextual Integrity. In *USENIX Security Symposium*, pages 499–514.

Wijesekera, P., Baokar, A., Tsai, L., Reardon, J., Egelman, S., Wagner, D., and Beznosov, K. (2017). The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 1077–1093. IEEE.

Wijesekera, P., Baokar, A., Tsai, L., Reardon, J., Egelman, S., Wagner, D., and Beznosov, K. (2018). Dynamically Regulating Mobile Application Permissions. *IEEE Security and Privacy*, 16(1):64–71.

Xiao, Y. and Xiong, L. (2015). Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1298–1309, New York, NY, USA. Association for Computing Machinery.

Xu, F., Tu, Z., Li, Y., Zhang, P., Fu, X., and Jin, D. (2017). Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1241–1250. International World Wide Web Conferences Steering Committee.

Yu, S. (2016). Big privacy: Challenges and opportunities of privacy study in the age of big data. *IEEE Access*, 4:2751–2763.

Zang, H. and Bolot, J. (2011). Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th annual International Conference on Mobile Computing and Networking*, pages 145–156. ACM.

Zavala, L., Dharurkar, R., Jagtap, P., Finin, T., and Joshi, A. (2011). Mobile, collaborative, context-aware systems. *Proc. AAAI Workshop on Activity Context Representation: Techniques and Languages*, pages 79–84.

Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM.

Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2012). Geolife gps trajectories. `https://www.microsoft.com/en-us/download/details.aspx?id=52367`. [Online; Accessed: 2019-12-12].

Appendix A.

Top Installed Applications

Table A.1 presents the top 2 installed applications per category and respective install/user count.

Category	Application	Count
WEATHER	Meteo@IPMA	5
	AccuWeather: Weather alerts & live forecast info	3
VIDEO_PLAYERS	Google Play Movies & TV	38
	VLC for Android	8
UNIDENTIFIED	cn.wps.xiaomi.abroad.lite	20
	com.micredit.in	17
TRAVEL_AND_LOCAL	Booking.com: Hotels, Apartments & Accommodation	14
	Tripadvisor Hotel, Flight & Restaurant Bookings	13
TOOLS	Google Translate	26
	Samsung Calculator	23
SPORTS	FlashScore - resultados desportivos	11
	Placard	8
SOCIAL	Instagram	68
	Facebook	44
SHOPPING	Cartão Continente	22
	OLX - Compras Online de Artigos Novos e Usados	21
PRODUCTIVITY	Google Drive	45
	Google Docs	38
PHOTOGRAPHY	Google Photos	41
	Adobe Lightroom - Photo Editor & Pro Camera	5
PERSONALIZATION	Mi Wallpaper Carousel	20
	Nova Launcher	4
NEWS_AND_MAGAZINES	Google News - Top world & local news headlines	12
	Quora — Ask Questions, Get Answers	4
MUSIC_AND_AUDIO	Spotify: Listen to podcasts & find music you love	74
	YouTube Music - Stream Songs & Music Videos	34
MEDICAL	Lady Pill Reminder ®	2
	Medscape	2
MAPS_AND_NAVIGATION	Uber - Request a ride	50
	Waze - GPS, Maps, Traffic Alerts & Live Navigation	24
LIFESTYLE	Glovo: Order Anything. Food Delivery and Much More	26
	Google Home	17
LIBRARIES_AND_DEMO	Pydroid repository plugin	1
HOUSE_AND_HOME	Imovirtual Real Estate Portal	2
	idealista	2
HEALTH_AND_FITNESS	STAYAWAY COVID	28
	Samsung Health	20
GAME	Among Us	26
	Pokémon GO	11
FOOD_AND_DRINK	Uber Eats: Food Delivery	47
	McDonald's	27
FINANCE	MB WAY	65
	Revolut	30
EVENTS	RHI Think	1
	Viral Agenda - Event Guide	1
ENTERTAINMENT	Google Play Games	55
	Netflix	40
EDUCATION	Photomath	19
	uni - A FEUP no teu bolso	13
COMMUNICATION	WhatsApp Messenger	86
	Messenger – Text and Video Chat for Free	73
COMICS	ComicScreen - ComicViewer	2
	WEBTOON	1
BUSINESS	ZOOM Cloud Meetings	38
	LinkedIn: Jobs, Business News & Social Networking	28
BOOKS_AND_REFERENCE	Amazon Kindle	7
	Google Play Books - Ebooks, Audiobooks, and Comics	7
BEAUTY	Barbearia Asgard	1
AUTO_AND_VEHICLES	Standvirtual Carros: Comprar melhor, vender melhor	3
	Fuelio: gas log, costs, car management, GPS routes	2
ART_AND_DESIGN	PENUP - Share your drawings	2
	Canva: Graphic Design, Video Collage, Logo Maker	1

Table A.1.: Top 2 installed applications per category and respective install count in the static data.

Appendix B.

Feedback Questionnaire

At the end of the data collection campaigns and after handing the reward voucher, we would email to the participant a link with an optional questionnaire. This survey was completely anonymous and all questions were optional. What follows is some of the translated questions present in the questionnaire.

- After your participation, do you consider the default Android permission manager sufficient? {Yes, No, Maybe}
- On a scale of 0 (nothing) to 5 (a lot), how much were you surprised by the number of permission requests made by the applications during the campaign?
- Was there any permission request that concerned you? For instance, because you were not expecting it? Which app or apps and respective requested permissions did concern you? (Open question)
- Was there any behavior, configuration or application that you have changed or will change as a result of the campaign? Could you detail what you will change or have changed? (Open question)
- Which information of the context do you consider more important when deciding whether to allow or deny a permission? (Open question)

Appendix C.

Information Gain

Tables C.1a and C.1b show the information gain for the expectation and grant result with each other feature, respectively.

	wasRequestExpected
grantResult	0.174932
isRequestingAppVisible	0.034572
permission_PHONE	0.026482
isTopAppRequestingApp	0.017188
permission_STORAGE	0.014735
category_VIDEO_PLAYERS	0.013996
selectedSemanticLoc_Home	0.013355
networkStatus_NOT_METERED	0.009829
category_COMMUNICATION	0.009160
permission_CAMERA	0.008898
category_HEALTH_AND_FITNESS	0.005937
category_PRODUCTIVITY	0.005420
category_LIFESTYLE	0.005284
permission_CALL_LOG	0.005077
category_PHOTOGRAPHY	0.004513
category_SOCIAL	0.003996
networkStatus_METERED	0.002889
category_EDUCATION	0.002439
category_NEWS_AND_MAGAZINES	0.002396
selectedSemanticLoc_Travelling	0.002374
category_GAME	0.002355
hour	0.001745
callState	0.001625
category_AUTO_AND_VEHICLES	0.001536
permission_SMS	0.001520
category_SHOPPING	0.001340
category_COMICS	0.001339
category_BOOKS_AND_REFERENCE	0.001272
networkStatus_DISCONNECTED	0.000925
isWeekend	0.000287
category_EVENTS	0.000282
category_WEATHER	0.000075

(a) User Expectation

	grantResult
wasRequestExpected	0.182551
permission_STORAGE	0.018125
category_VIDEO_PLAYERS	0.016729
category_COMMUNICATION	0.013858
permission_PHONE	0.013843
selectedSemanticLoc_Home	0.011375
networkStatus_METERED	0.008086
isRequestingAppVisible	0.007845
networkStatus_NOT_METERED	0.007624
permission_CAMERA	0.006610
permission_CONTACTS	0.005629
selectedSemanticLoc_Travelling	0.005558
category_GAME	0.005136
category_TRAVEL_AND_LOCAL	0.004675
plugState	0.003084
category_WEATHER	0.002793
isTopAppRequestingApp	0.002483
category_TOOLS	0.002355
category_MUSIC_AND_AUDIO	0.002348
permission_LOCATION	0.002234
hour	0.002170
permission_CALL_LOG	0.001951
category_PERSONALIZATION	0.001940
selectedSemanticLoc_Work	0.001710
permission_SENSORS	0.001467
category_BUSINESS	0.001452
category_SOCIAL	0.001350
category_SPORTS	0.000973
category_SHOPPING	0.000963
category_HEALTH_AND_FITNESS	0.000801
isWeekend	0.000623
category_MEDICAL	0.000485
callState	0.000389
category_LIFESTYLE	0.000265
category_ENTERTAINMENT	0.000156
category_FOOD_AND_DRINK	0.000122
networkStatus_DISCONNECTED	0.000103

(b) Grant Result

Table C.1.: Information Gain for the expectation and grant result with every other feature. Showing only values greater than 0.

Appendix D.

Grid-Search For The Best Global Predictor

To evaluate the best model for predicting privacy decisions, a 5-fold cross-validated grid-search was implemented for each of the experimented parameters. What follows is a list of parameters that were experimented in the grid-search, whose meaning and implementation details can be consulted in the SciKit documentation [Pedregosa et al., 2011]. Unmentioned parameters were used with the default values.

- Linear SVM:
  - C: 16 values evenly spaced on a log scale from 10^{-8} to 10^{3}. This range extends the ones used in [Liu et al., 2016], as the performance was increasing for lower values.
  - Maximum number of iterations: 500000.
- RBF SVM:
  - C: 8 values evenly spaced on a log scale from 10^{-5} to 10^{8}
  - Gamma: [scale, auto, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}]
- Decision tree:
  - Criterion: [giny, entropy]
  - Splitter: [best, random]
- Bagging:
  - Boostrap classifiers: [True, False]
  - Boostrap features: [True, False]
  - Number of estimators: [5, 10, 15, 30, 50]
  - Maximum number of samples: [0.6, 0.8, 1.0]
- Ada Boosting:
  - Number of estimators: [10, 20, 30, 50, 100, 150]

  - Learning rate: [0.01, 0.05, 0.1, 0.2, 0.5]

- Random Forest:
  - Boostrap: [True, False]
  - Number of estimators: [5, 10, 15, 30, 50]
  - Maximum Features: [sqrt, log2]
- Neural Network:
  - Hidden Layer Sizes: 7 topologies of equally spaced number of neutrons ranging from 1 neutron to the number of input features
  - Maximum number of iterations: 500000.