PRIVACY CONFLICT ANALYSIS IN WEB INTERACTION MODELS

PETER INGLIS

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy

SCHOOL OF COMPUTING SCIENCE

COLLEGE OF SCIENCE AND ENGINEERING
UNIVERSITY OF GLASGOW

APRIL 2022

Abstract

User privacy has become an important topic with strong implications for the manner by which software systems are designed and used. However, it is not a straightforward consideration on how the instrumentation of data processing activities contribute to the privacy risk of data subjects when interacting with data processors online. In this work, we present a series of methods to assist Data Protection Officers (DPOs) in the modelling and review of data processing activity between data processors online. We articulate an awareness formalism to model the knowledge gain of data processors and the privacy expectations of a data subject. Privacy conflict is defined in this work as an event where the expectations of the data subject do not align with the data processors knowledge gain resulting from data processing activity.

We introduce a Selenium workflow for the elicitation of data processing activity of web services online in the creation of an information flow network model. We further articulate a series of privacy anti-patterns to be matched as attributes on this model to identify data processing activity between two data processors facilitating conflict between data subjects and processors. Each anti-pattern illustrates a distinct manner by which conflict can arise on the information flow model. We define privacy risk as the ratio of third party data processors that facilitate an anti-pattern to the total number of third party data processors connected to a first party data processor. Risk in turn quantifies the privacy harm a data subject may incur when interacting with data processors online.

Pursuant to the reduction of privacy risk, we present a multi objective approach to model the inherit tensions of balancing the utility of a data subject against the cost incurred by a data processor in the removal of anti-patterns. We present our approach to first elicit the Pareto efficient set of anti-patterns, before operating on a utility function of programmable biases to output a single recommendation. We evaluate our approach against trivial selection strategies to reduce privacy risk and illustrate the key benefit of a granular approach to analysis. We conclude this work with an outlook on how the work can be expanded along with critical reflections.

Acknowledgements

I would like to extend my sincere gratitude to Dr Inah Omoronyia for his support throughout this PhD. His guidance and supervision have been instrumental in the success of this project and his teachings an invaluable contribution to my growth as a researcher.

Further thanks to Dr Angelos Marnerides and Dr Mathias Humbert for a constructive and insightful examination of the work, along with Dr Matthew Barr for taking on the role of convener to organise the event.

I would like to thank my colleagues in the Glasgow Systems Section. You all made me feel welcome at the University of Glasgow and I have learned much from all of you. Paddy, I will cherish our coffee breaks, and impromptu German lessons. To everyone involved in the D&D group, I'm glad the games took off and I will fondly remember those chaotic nights for a long time to come.

A thank you to my friends, who have shown exceptional patience and understanding during my studies. I'm truly blessed to have such a network of support. Further thanks to Gordon, for taking interest in what I do and being a positive force in my life.

Special thanks of course to my family for their love, guidance and unyielding encouragement. I cannot thank my Mum and Dad enough for everything they have done to nurture my aspirations. Finally, I would like to thank Zena for being there with me on this journey, I could not have done this without you. Sincerely, thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction and Motivation

Numerous contemporary software systems collect and process user data for the provision of social, commercial and professional services to data subjects [1]. Such information can facilitate analytics, enabling businesses to better understand how their services are utilised to forecast changing demand for products or features. Targeted, or 'smart advertising' models have greatly incentivised the mass collection and processing of user data through its commodification with a market value of $10 billion in 2017 [2] [3]. The application of smart advertising has been criticised as 'invasive' and 'predatory' leading to the introduction of contemporary data protection regulation [4] [5]. The EU General Data Protection Regulation (GDPR) is the most prolific regulation aiming to promote data subject control by providing a series of statutory obligations for both data controllers and data processors [6] [7]. Data controllers are entities responsible for determining the purpose and conditions of data processing, whilst data processors carry out processing activity on behalf of controllers under agreement.

According to GDPR, controllers are liable for the activities of their processors[1] and under the Fair Information Principle are expected to identify and communicate a legal basis for data processing to data subjects. Consent and legitimate interest are two legal basis recommended by the Information Commissioners Office for data processing activity with intended commodification involving external processors and controllers. The significance of such legal basis concerns the right of data subjects to express their preferences regearding consent on such data processing activity. To this end, Consent Management Platforms (CMPs) are commonly instrumented to provide notice of data processing activity and to elicit preferences

[1] In this work we utilize the terminology 'data processor' and 'data controller' interchangeably. Data controllers that operate on behalf of another controller are treated as affiliated processors of the controller per the targeted GDPR.

or acknowledgement towards such data processing, depending on the selected legal basis [8]. The nature of control and levels of transparency offered to data subjects is often dependant on the specific CMP solution adopted by a data controller/processor. Some CMPs operate on explicit consent, such that the data controllers and processors involved in the dissemination of data subject information are listed as 'third party vendors' with a series of check boxes to elicit granular consent from data subjects on what data processing is acceptable. Other CMPs operate on implied consent and only provide data processing notice to data subjects, without providing them the option to reject or otherwise express their objection to such data processing activity [9], [10], [11].

When interacting with data controllers, the CMP mechanisms will impact the capacity for the data subject to understand how their data will be disseminated to external data processors and controllers [12], [13]. This in turn will inform their decision on whether to enter into business relations with data controllers. However, the diverse levels of transparency offered by data controllers through CMPs can undermine the privacy expectations of data subjects when interacting with multiple controllers. Assume we have two data controllers F_1 and F_2 with F_3 as a mutual connection. Here, the CMP operated by F_1 operates on explicit consent and provides the data subject (su) the option to reject data processing from F_3. However, the CMP operated by F_2 operates on implied consent and only provides notice that data processing involving F_3 occurs. If su interacts with F_1, rejects data processing and subsequently interacts with F_2, then F_3 will be involved in data processing, conflicting with the previous consent preferences of su. Further, if F_2 did not exhibit any data processing transparency involving F_3 through CMPs to su again the involvement of F_3 in data processing will conflict with the previously indicated preferences of su . Both scenarios mean the expectations of a data subject may not accurately reflect the data processing activity that occurs. This disadvantages the data subject and undermines their capacity to make decisions that can satisfy their privacy concerns. Data subjects are often pointed towards the associated privacy policy of the responsible data controller with the assumption they have read, understood and agree to the terms set out in the policy. However, the challenges with privacy policies are well understood in privacy literature. The abstract language of policies does not provide exacting information to data subjects and the number of policies they would have to read is (in the worst case) proportional to the total number of distinct controllers/processors affiliated with the service they are interacting with [14].

For data controllers to demonstrate compliance with GDPR, there is a mandate to provide demonstrable evidence that their business process enshrines the principles of Fair Information Practice through the enactment of Data Protection Impact Assessment (DPIA) [15], [16], [17]. The Information Commissioners Office (ICO) convey that DPIA procedure allows for the systematic review of inherit privacy risk faced by data subjects as a result of business models, with a particular focus on the data processing activities that support them. A key consideration for businesses (acting as data controller) with a portfolio of properties is effective scrutiny of data processing activity that occurs on these properties. A DPO tasked with evaluating the risk to data subjects through DPIA on the portfolio of a data controller will be interested in scrutinising the implemented CMP solutions involving both data controllers within the portfolio along with externally affiliated data controllers or processors. Understanding how privacy expectations can be undermined by data processing activity occurring on their portfolio will afford responsible data controllers the capacity to demonstrate steps taken to reduce privacy risk faced by data subjects through refactoring of data processing activity on its portfolio.

A key issue faced by DPOs when enacting DPIA is the lack of objective metrics to quantify privacy risk in software. This makes it challenging to articulate clear measurements of risk introduced by data processing activity, resulting in the application of subjective assessments in their analysis. Quantifying privacy risk with objective metrics will assist DPOs in both identifying granular mitigation strategies and further articulate their efficacy in reducing risk to other stakeholders. In this thesis, we aim to address the key challenge of objectively quantifying privacy risk by leveraging on a formalism to model privacy conflict between the privacy expectations of data subjects and the consequence of data processing activity between data controllers. This is one of the novel contributions distinguishing this work from other privacy research focusing on the nature of privacy conflicts.

Typically, privacy conflict is defined as an interdependent concern between multiple data subjects where the intended actions of one party may violate the privacy requirements of individuals, or collectives in an online setting [18]. In this work we define privacy conflict through incompatible knowledge state of data subjects and controllers to understand how the instrumentation of CMPs at design time can introduce privacy risk at runtime to data subjects. We unpack a solution framework, by first discussing a methodology to model the data processing activity between data controllers. We introduce privacy anti-patterns, to identify usage contexts facilitating privacy conflict through misaligned privacy expectations. Finally, we discuss a method to investigate refactoring options to reduce observable privacy risk. We unpack metrics to model the utility of refactoring options to both the data subject and data controllers, to arrive at an optimal refactoring outcome in consideration of these conflicting interests. It is the intention of this work to contribute to the paradigm of privacy by design through the provision of a decision support tool to assist DPOs communicate to data controllers how they can uphold the GDPR principle of accountability through the analysis of privacy risk introduced through data processing activity, and remedial steps to reduce privacy risk.

1.2 Thesis Statement

Consent Management Platforms are often implemented in online web settings to provide data subjects technical means by which they can exert control over how they wish their personal information to be processed. The manner by which CMPs are implemented are often interdependent, involving varying levels of data processing transparency. CMPs can involve the implementation of consent elicitation modules, the forced acceptance of data processing activities or the absence of any processing transparency information. The resulting privacy expectations held by a data subject, are tied to both the level of transparency offered by a data processor and if provided, their indicated preferences on consent modules.

The interaction with data processors can in turn lead to the establishment of mutually exclusive privacy expectations from a data subject, which can result in privacy harm. This harm arises from the fact that the expected level of privacy loss a subject realises when interacting with data processors may not align with the actual privacy loss experienced resulting from data processing activity. Further, the capacity for data subjects to perceive such conflict is often diminished in online environments, yet responsibility for identifying and mitigating such conflicts, falls to the end user.

In this thesis, we propose a series of methods to assist Data Protection Officers (hence analysts) involved in Data Protection Impact Assessment (DPIA). The principle is to afford analysts the capacity to model the data processing behaviour of a series of data processors as an information flow network. We articulate a series of usage contexts that can be cross examined with the information flow representation to understand what subject control implementations can facilitate misaligned privacy expectations. By introducing a metric of privacy risk, we can quantify the likelihood that a data subject will incur privacy harm via their interactions with data processors. This in turn will allow for the subsequent review of connections on an information flow model that can be refactored to reduce observable privacy risk. We introduce a series of methods that can support analysts in making optimal refactoring decisions in the event the interests of both the data subject and data processors compete with one another.

The work discussed in this thesis is intended to contribute to the paradigm of Privacy by Design to allow for the reduction of observable privacy risk associated with data processors through refactoring suggestions. Such activities will benefit data subjects indirectly by minimising the privacy harm that will be incurred when interacting with such data processors.

1.3 Research Questions

This thesis aims to investigate and discuss the following research questions:

RQ1 - Quantifying the impact of data processing activity on the privacy of data subjects involves applying objective metrics to measure a subjective social construct, which is not a straightforward consideration. Being able to understand how the knowledge of data processors evolve as a result of such activity and how the expectations of data subjects change when interacting with CMPs is one approach to address this challenge. What models can be leveraged on to articulate the expectations of data subjects and the evolution of data processors knowledge?

RQ2 - When modelling the activity of data processors there is a need to scrutinise and distinguish data processor behaviour. Constructing an information flow network that delineates observable levels of transparency offered by data processors is one approach to address this requirement. However, such modelling activities often incur high time requirements for analysts. This is particularly true when the analysis involves multiple data processors. Is there a methodology that can be followed to reduce such time requirement?

RQ3 - DPIA procedure requires analysts to document perceivable risk to data subjects. However, quantifying the impact of interactions between data subjects and multiple data processors as a measurement of risk is not always a straightforward design time consideration. Being able to understand what data processing activities contribute to the privacy risk of a data subject will help address this challenge. What method can be used to validate that a data processing activity will contribute to privacy risk?

RQ4 - A crucial aspect of DPIA process is the enactment of mitigations to reduce identifiable risk to data subjects. The challenge with risk mitigation is the requirement for analysts to selectively recommend outcomes as mitigating all possible risks is often infeasible for data processors. The development of a decision support system would help support analysts elicit appropriate recommendations. What methods can be used to reduce the space of candidate refactoring recommendations to realise an optimal outcome for data subjects and processors?

1.4 Contributions

The contributions from this thesis can be summarised as follows:

A privacy awareness model:

In this work we introduce an awareness model that leverages off Epistemic Modal Logic to represent a software objects[2] understanding of information pertaining to both itself and others. We apply this model to represent the privacy expectations of data subjects when interacting with data processors in addition to the consequences of data processing activity.

A formalism for privacy conflict representation:

We present a series of semantic rules leveraging off the awareness formalism to articulate various incompatible states of awareness for a software object. We discuss three such rules including explicit, implicit and uncertainty conflict along with their implications. We further provide an exhaustive account on how such conflicts can be facilitated.

A series of privacy anti-patterns:

We discuss a series of privacy anti-patterns composed of a usage context, involving a data processor and data subject, along with a collection of resulting awareness transforms. The awareness transforms are used in this work to verify the existence of conflict by virtue of incompatible awareness state. The usage context illustrates the run-time interaction scenario resulting in the specific awareness transforms occurring. This is used to unpack the manner by which misaligned privacy expectations can manifest when data subjects interact with multiple data processors.

Strategies for conflict resolution:

We introduce a number of decision making strategies an analyst can leverage on to determine how to best reduce the observable privacy risk to data subjects. We unpack strategies focusing on single decision criteria, and modelling conflicting objectives. We discuss the benefits and drawbacks to both strategies before articulating the application of a utility function to base decision making on granular bias towards data subjects or data processors.

A workflow to elicit an information flow model

We unpack a methodology to elicit a series of data processing activities from data processors to model such activity as an information flow network. We articulate three information flow types, mandatory, optional and covert to delineate observable behaviour of data processors. This approach allows an analyst to output an annotated graph representing the data processing activities between multiple data processors.

[2] A software object specifically refers to a data subject, controller or processor

A case study on privacy risk refactoring

We unpack a case study to elicit observable data processing behaviour of 51 data processors. We discuss how we assign privacy anti-patterns to edges on the information flow model and how these annotations are used to gauge the degree of privacy risk incurred by the CMP implementations of data processors. We model the refactoring options available to an analyst, which involves mapping the utility of removing different privacy anti-patterns from the information flow model. We explore the output from a single decision strategy, multi objective strategy and finally a weighted bias utility strategy to provide optimal recommendations to analysts. We discuss why these strategies in turn are pragmatic options to assist stakeholders identify desirable outcomes when reducing privacy risk.

1.5 Benefactors

Data Protection Officers:

The main benefactor we target with this work are Data Protection Officers. This project aims to assist DPOs in the modelling of software systems. It is the intention of this work that DPOs are provided insight into how the refactoring choices available to them will impact both the privacy risk of the data subject, along with the interests of the data processors. This in turn will allow DPOs to better articulate the perceivable benefits in a quantifiable manner to other stakeholders involved in data protection discussions.

Data Subjects:

We see the secondary benefactors of this work being data subjects. By giving DPOs the appropriate tools required to both model and conduct analysis of privacy risk in software, we envision that data subjects will benefit from the re-conceptualised data processing activity exhibited by data processors they interact with online.

1.6 Thesis Structure

The thesis structure is as follows:

Section 2 presents an overview of privacy literature. The literature review is split into 2 main sections. It begins with a focus on general privacy concepts from multiple perspectives before leaning into a more focused discussion on how privacy conflict analysis is practised in both industry and academia.

Section 3 details a brief overview of the proposed solution framework in this thesis. Here, the three key workflows of data processing modelling, conflict modelling and refactoring analysis are detailed, along with a mapping of targeted privacy legislation concerns to the framework. Finally, we discuss envisioned usage contexts with the framework.

Section 4 details the privacy awareness model developed over the course of research. It details the underlying semantics of possible worlds and Epistemic Logic that the awareness operator leverages on. It also details a series of conflict formalisms that are used to denote incompatible awareness states of objects that are being modelled. We discuss explicit, implicit and uncertainty conflict and how pairings of awareness instances facilitate such conflict.

Section 5 details the methodology followed in this work to elicit the information flow network representation of data processing behaviour online. It illustrates a Selenium based workflow, focusing on the elicitation of an information flow model representing online web services.

Section 6 presents the privacy anti-patterns conceptualised in the course of research. It details the different aspects of conflict that are verified by the awareness model and presents a series of interaction contexts involving the data processor and data subject.

Section 7 presents our approach to conflict analysis that involves the application of economic models to assist in decision making. We demonstrate 3 methods to conflict analysis and illustrate the efficacy of each method under different operational contexts. We conclude this chapter discussing of the implications of analysts utilising the three methods in practice.

Section 8 concludes this thesis with a critical reflection on the work conducted. We discuss the research questions and decompose how we addressed each research question. We present limitations, threats and opportunities in this work before finishing with concluding remarks in section 9.

Chapter 2

Literature

2.1 Introduction

Privacy literature co-evolved with the advent of technologies and business ventures that commodified the mass collection and processing of personal information. Some contemporary literature on privacy leans towards better informing the creation of business practices or software products that are sensitive to the nature of privacy. A key requirement for such literature is the capacity to express privacy in a quantifiable manner. Translating the subjective concept of privacy into an objective technical interpretation is therefore a key research challenge that underpins several research contributions.

In this chapter, we aim to provide a (brief) overview of some key privacy research themes, along with a critical analysis on how we distinguish the contributions of this work from the literature before positioning this thesis in the privacy research ecosystem. We begin this chapter in section 2.2 with a discussion of some key fundamental concepts in order to achieve privacy in software. We convey the paradigm of Privacy by Design before unpacking several underpinning interpretations on how the principles of Privacy by Design can be achieved in a technical context. We discuss industrial attitudes towards achieving privacy and focus on differential methods to both safeguard privacy on data held by a business, and data that is published by a business.

In section 2.3 we introduce the sub-discipline of privacy conflict analysis and begin with a discussion of privacy policy analysis in section 2.3.1. We focus on work that looks to parse policies into a formal notation to cross reference with modelled data processing activity. In section 2.3.2 we discuss a series of rule based approaches to identify privacy conflicts through formal policy specification languages. We start with industrial policy specification languages such as XACML and EPAL before illustrating various initiatives to extend the expressive capacity of such models. We further look to approaches leveraging off privacy calculus theory to model conflicts through metrics to capture subjective modalities of privacy such as the sensitivity of information. Finally, in section 2.3.3 we look to unpack a series of conflict analysis methods which fall within the domain of social networks. We discuss work applying consensus techniques in multi agent models such as single round negotiation and k-round negotiation before unpacking alternative methods such as argumentation and trust modelling. We conclude this chapter with a summary in section 2.4

2.2 Engineering Privacy in Software

Privacy by Design (PbD) is a broad and interpretable concept commonly described in privacy literature as a series of methods, or guiding principles that aim to not only inform the design, development and subsequent instrumentation of software systems, but also aim to help refine the business processes instrumenting them [19]. The concept of PbD was originally introduced in the work of Ann Cavoukian and is a school of thought towards business and technical processes that aims to address the challenge of developing privacy preserving software. [20].

The work from Cavoukian serves to articulate a series of principles to guide stakeholders in addressing the challenge of incorporating privacy into systems at various stages of development [21]. What it does not do, is serve to provide concrete steps to achieve said principles [22]. Given there is no consensus on what modalities should be involved in the quantification of privacy, the research landscape is consequently diverse with multiple independent, yet comparable research initiatives. These initiatives champion the attainment of requirements to create software exhibiting certain 'privacy-centric' traits in its operation. To achieve this, developers can incorporate Privacy Enhancing Technologies (PETs) as off the shelf solutions and integrate them into their development life-cycle.

PETs are described as a separate, yet relatable concern to PbD describing a suite of tools, or a series of methodologies to secure the informational privacy of users [23]. According to Burket, the implementation of PETs is to generally provide greater levels of privacy assurances to end users by enabling them to control the circumstances by which personal information is revealed. PETs vary wildly in scope, both in terms of how they are instrumented, and what each technology aims to achieve. PETs have been described as the resulting product of software development initiatives following PbD principles [24]. Conversely, PETs have also been described as a component of PbD such that their integration in the development cycle can lead to privacy preserving systems.

Both PETs and PbD share a common design philosophy to provide privacy safeguards to end users. In literature, there exists numerous terminologies and ontologies from the legal, academic and software sectors that discuss a series of intermediate approaches or design strategies to achieve PbD from different perspectives. These often overlapping concepts can be consolidated into the following key 'privacy requirements[1]' for developers to observe PbD principles.

Anonymity is arguably the most discussed term associated with privacy. In the context of PbD, the objective of a system providing anonymity to a data subject is to ensure a state of non-identifiability for a given actor within a set of users. To maintain non-identifiability for the sender and receiver is to make it impossible for a third party to effectively distinguish the activities of one party over another. Virtual Private Networks are one example of a centralised solution to providing anonymity by encrypting end to end communication channels. Decentralised solutions such as the Tor network also aim to provide anonymity through communication relays.

Unlink-ability refers to the capacity for a given attacker to be able to make inferences on observable, or published data. In the context of PbD, unlink-ability is typically a design objective of several approaches within database applications to protect the privacy of data subjects within data tables by making it difficult for attackers to infer additional information about the data subjects. In brief, common approaches to modelling unlink-ability reference the a-priori and posteriori knowledge an attacker possesses. As suggested, a-priori knowledge relates to the information an attacker may have regarding a data subject whilst posteriori knowledge relates to the information gained by an attacker regarding data-subject attributes. The objective and capacity for unlink-ability approaches to be evaluated in software is to minimise the deltas between the a-priori states and posteriori states.[2] Anonymity and unlinkability are often discussed interchangeably as both concepts are frequently supported through complimentary solutions.

Data Minimisation relates to the amount of personal data requested, or required by a software system when servicing the requests of users. Web browsers are a good example of a technical implementation of data minimisation, as several privacy-evangelising web browsers distinguish themselves as not requiring the collection or processing of user data for the provision of web browsing services. Firefox is an example of a privacy aware web browser that by default actively takes measures to protect the browsing habits of users online by blocking tracking scripts and third party cookie files. Conversely in the same ecosystem, DuckDuckGo is an example of a search engine which minimises the processing activity of data subjects. Ephemeral communication tools are another class of methods under the data minimisation category as these software systems are designed without data persistence

[1] Note the term privacy requirement is not often considered a solid definition in the literature, but is rather a result of desired services being provided to end users. This can be a result of architectural decisions being made along with the implementation of specific security requirements such as end to end encryption to provide security in the form of anonymising communication channels through encryption.

[2] There are exceptions to this in the event that false knowledge gain is the objective of a given privacy solution.

in mind. The principle of ephemeral systems is to limit the availability of data, promoting interaction over a brief time period before the information that is disclosed is permanently deleted from the system. Work from Timpthy and Sehn discuss a patent where objects delivered to edge systems are scheduled for automatic deletion after a specific criterion is met, such as time, viewer count, or access. Dunn et al discuss the implementation of a Linux project Lacuna focusing on the erasure of execution traces from the host operating system to provide privacy through forensic protections.

When reviewing the data processing activities involving personal information from data subjects focus is often paid to the manner by which data subject information is stored and processed. Privacy preserving statistical analysis is an active research area concerned with the privacy protections of data subjects within databases. The research area developed as a response to the challenge of mitigating the privacy harms arising from the mass collection and analysis of user data [25] [26]. Traditional access control approaches do not provide sufficient guarantees in this problem context as the privacy harms arise from the actions taken by authorised bodies on data records. Thus, the objective of 'privacy preserving statistics' is not to restrict access to a database, but rather to minimise the information attainable through queries about a specific database record, and to prevent inference through external linking of multiple datasets [27] [28]. Proposed privacy solutions in this domain do not typically offer individual guarantees of privacy; But rather operate on the statistical probability of an individual users privacy being compromised whilst opting for the provision of group privacy. This is a consequence of the challenge in balancing the trade-off between adequate utility of a dataset and the individual protections afforded to all data subjects within the data set.

Privacy preserving data mining is a term typically referring to differential privacy. The principle behind differential privacy is to return responses of data queries in a manner as to not violate the privacy of individual users. To achieve this, a series of algorithms are used to inject noise into a resulting query from a dataset through randomiser functions [29]. The objective of these randomiser functions is to obscure the truth values of observed data in queries.

Differential privacy is described by Dwork and Roth as "a definition of privacy tailored to the problem of privacy-preserving data analysis". The randomiser function (or algorithm) provides ϵ privacy in the event it is ϵ hard to distinguish between two databases differing by a single entry. Differential methods are reliant on strong randomisation algorithms to provide their privacy guarantees. Several of these randomiser functions exist and are the subject of research on 'demonstrable privacy' approaches[30] [31] [32].

ϵ privacy has been coined as the definitive approach to safeguarding user privacy and has seen adoption in industry to support the paradigm of Big Data. Dwork and Rothblum discuss in their work that one limitation that can be observed with differential privacy methodologies is upholding the minimal privacy loss value ϵ. They argue the ϵ values associated with cumulative privacy loss may fall outwith of acceptable loss values, as the measured privacy loss in multiple computations are based on estimated means, rather than guaranteed values on each computational step. Traditional ϵ value methods, they argue results in an exponential probability of privacy loss being in excess of the preferred value as the number of consecutive computations are increased. In their paper they address this weakness of differential privacy through a relaxation technique. The objective of their approach is to limit the probability of privacy loss exceeding the mean value. Their approach favours higher probability bounds for multiple queries over single queries, aiming to constrain the probability of cumulative privacy loss at the expense of single query privacy loss, as is the case with other approaches to differential privacy [33].

Similarly, Moronov discusses another relaxation method to differential privacy based on Renyi divergence. Their approach to differential privacy relaxation similar to Dwork and Rothblum provides greater guarantees for composite privacy queries via the measurement of cumulative privacy loss. Their approach, called Renyi Differential Privacy (RDP), shares several characteristics with the standard relaxation technique (epsilon, delta) but its implementation is streamlined, allowing for stronger probability bounds when its parameter alpha is of a lower value than delta [34].

A second vein of research discusses approaches under privacy preserving data publishing. The principle is similar to differential privacy, to provide data subjects privacy guarantees within a database. However, unlike differential privacy where randomiser functions are used to obscure the truth values of resulting database sets, syntactic privacy approaches operate on a 'publish and forget' principle, where the database itself is randomised when published. Generally, a database is partitioned into blocks, and a series of randomisation operations are enacted on the blocks with the objective of achieving privacy through record indistinguishability.

k-anonymity is an anonymisation metric relating closely to the concept of differential privacy with similar objectives, to minimise the level of entropy that can be elicited from the linking of distinct data sets. In brief, a given dataset is said to be k-anonymous if the data for each record is indistinguishable from at least k-1 records whose information is also within the same dataset [35].

The original concept introduced by Sweeny has received some criticisms by other researchers. Machanavajjhala et al demonstrate homogeneity and background knowledge attacks against k-anonymity tables under the assumption of Bayes-optimality. The paper assumes an attacker will have partial or background knowledge regarding records found within a k-anonymity table [25]. In their paper the concept of "Bayes-optimal privacy" utilises a probability distribution along with Bayesian Inference [36] for the purpose of reasoning about privacy. The concept of posteriori beliefs are used to measure the level of knowledge entropy an attacker can gain when observing a value at random from the published database.

In their contributions they define a table $q*$ block to be a set of tuples T where the non-sensitive attributes can be generalised. Therein, a table is considered l-diverse where, given a known set of metrics, an attacker is unable to make inferences on the sensitive information of a data subject as a result of there being l number of distinct sensitive attributes associated with the non sensitive ones. In their paper they demonstrate the l-diversity approach on a table of medical information and compare their methods with the k-anonymity model.

The topics of l-diversity and t-closeness are refinements of the k-anonymity system which serves to act as a metric for assessing the degree of privacy offered to users. The advantage is a measurement of the trade-off between the utility offered to data processors on database applications along with a measurement of privacy protection offered to data subjects in a manner that can be configured. The limitation of this metric is the fact the metric is only generally applicable to databases. In more general settings, such as agent based privacy scenarios, the application of k-anonymity and its derivatives becomes less straightforward, especially in scenarios where there is no central repository of user records.

2.3 Analysing Privacy Conflict

2.3.1 Privacy Policy Analysis

Privacy conflict analysis is a research theme involving the understanding of intended information dissemination activities of one or multiple actors. The principle is to cross examine the consequences of such activities to determine whether or not these actions are in violation of legislative, or social contracts. From an industry perspective, conflict analysis typically encapsulates concerns relating to the detection of data handling practices that are in violation of data protection legislation. The main motivation behind such analysis is to either demonstrate regulatory compliance or understand how the business processes diverge from such requirements. The evaluation of regulatory compliance involves the modelling and subsequent analysis of data flow networks which aim to represent the data processing activities between one, or multiple data processors in an enterprise context. A second component of such analysis involves the parsing of natural language used to compose privacy legislation. Typically, a machine readable interpretation of one or multiple policies allows for the evaluation of modelled data flow networks to consider whether the identified processing activities adhere to, or violate policies. Being able to parse privacy policy into machine readable representation further allows for the evaluation of distinct privacy policies to identify discrepancies when looking to enact such policies on a technical solution.

There are several research veins proposing a series of formal semantics to represent the intended processing actions that should, or should not occur according to the privacy policy. One of the earlier initiatives to parse privacy policy into machine readable form involves the IBM Watson group. The approach in [37] leverages on a grammar along with a parser to identify rule elements along with a policy authoring utility SPARCLE. The SPARCLE workbench allows for the specification of privacy policies in natural language through the implementation of semi-structured rules which are then translated into XML. When inputting a policy description into the parser, the analyst is prompted to specify the policy as a single sentence following the structure Category, Action, Category, Purpose, Condition/Obligation where 'Category' can refer to either users (roles) or data (attributes). The challenge of formally articulating privacy policies in a manner to autonomously validate adherence in instrumented systems is an observable theme in more contemporary literature. Breaux, Hibshi and Rao introduce a formal approach to modelling privacy requirements through the analysis of privacy policies specified by data processors online. In their work, they articulate Eddy, a formalisation that leverages on Descriptive Logic (DL) to express unambiguously how data is to be collected, disseminated and for what purpose using the natural language of policies.

Their application of DL allows for the articulation of data, purpose and actors in a formal syntax. An encoding process is detailed which allows for the raw text of policies to be classified and later statements associated to the text. These statements are further refined and abstracted in order to elicit roles and subsumptions in order to arrive at a compiled specification in DL. This approach allows for the formal expression of privacy policies that map to a semantic expression in DL where permissions and prohibitions can be articulated. Conflicts are modelled in the work as a misalignment between privacy policies from multiple data processors online. Specifically, elicited DL statements that fall under both permissions and prohibitions are paired for conflict detection. The conflict detection itself is the result of ontological alignment between the formalised statements articulated in DL. Their approach allows analysts to identify conflicting intentions between the privacy policy documentation of multiple data processors with the outlook for such stakeholders to use this information to conduct review to ensure the intentions of privacy policies are better enshrined in developed software systems [38].

Follow up work from Breaux, Smullen and Hibshi leverage on the DL formalization's to propose an approach in [39] to detect non compliance with Fair Information Principles. They expand on the previous work to perform data flow tracing on the data transfer intentions from one data processor as specified in privacy policy with the collection activities provided by another data processor's privacy policy. A service mapping is used to assign data processors to roles and terminology mapping between distinct privacy policies removes the requirement for complex ontologies. The methodology followed is expanded to identify policy statements from the privacy policy elicited as data requirements. Each requirement is then annotated as a collection, usage or transference and is associated with agent roles. Refinement and abstraction of policy statements are mapped to the formal DL syntax before being compiled as a DL expression. Conflict this time, occurs in the event the DL generated between data processors lacks specificity, resulting in ambiguous conflict. Electable permissions conflicts occurs in the event that permission and prohibitions occur dependant on the consent preferences of a data subject to a data processor. Specifically, this conflict is considered a potential conflict to analysts. Finally, direct conflicts occur in the event there is a clear alignment problem with the permissions and prohibitions of privacy policies between data subjects.

The detection of conflicts constitute a manner by which the concept of risk can be expressed to analysts for the purpose of refinement of either privacy policies, or the technical mechanisms implementing them. Typically, the notion of risk relates to the possibility of inappropriate disclosure actions, or the elicitation of data not discussed within the privacy policy. In mobile settings, focus is often paid to understanding the behaviour of an application. Static flow analysis methods are utilised to identify execution traces involving API calls or parameter values which constitute dissemination of data subject information. The estimation of risk therefore, is largely determined by how closely aligned such behaviour is within permitted or prohibited actions that can be extracted from one or multiple privacy policies.

Work from Zhang et al aim to understand the behaviour of different codebases on the Android framework and introduce a static analysis approach to identify inappropriate attribute setting methods for apps. The aim of the approach is to consider whether the use of personally identifiable information as subject attributes in apps violate the privacy policies associated with the application. Their approach involves the creation of sentinel values in a user profile on an emulated Android device. The analysis involves checking for these sentinel values on sinks corresponding to attribute setting methods provided as client APIs in different analytic services. The work reported being able to detect usage of such methods without appropriate encryption or anonymisation efforts being used on the input data. The next step of the work involved cross referencing the privacy policy of the application and the terms of service supplied by the analytics processor to determine whether the usage context of the attribute setting methods constitute privacy risk for the end user. Risk is defined in the work as a consequence of the following two phenomena. Privacy misalignment, where the actions of the code differ from what is communicated in the privacy policy. Vague policies, such that the recommendations from the terms of service from analytics are not strictly upheld in code [40].

Salvin et al further discuss an approach to understanding privacy policy violations in Android systems. Their approach involves the construction of a hierarchical ontology to describe the relation between various technical terms within privacy policies, for the purpose of mapping such terms to low level Android API calls. Their method involves the annotation of API technical terms in provided Android SDK documentation that can impact the privacy of data subjects using the app. Coding frames were utilised to bound the criteria by which information within the API specs were annotated. The ontology was formulated with DL, applying subsumption rules before mapping the annotations to the ontology. Their approach to conflict analysis involves the investigation of application byte code, to determine whether method invocations (identified as taint sinks) occur out-with the privacy policy. This involves the generation of a mapping collection, where the collection can be queried to identify instances where API calls exist out-with this mapping. Two types of violations are unpacked, potential strong violations occur in the event an API call does not exist within the identified mappings and does not exist within the ontology structure, where the method call can be attributed to more abstract terminology as opposed to more explicit representations. If there is no mapping relation but the ontology can assign meaning to the suspected method call there is a potential weak violation. The principle of the work is to identify conflicts owed to the omission of the runtime behaviour of an application, or the ambiguous declaration of such behaviour in the privacy policy [41].

Caramujo et al discuss a privacy policy specification language to provide analysts with the capacity to determine whether privacy policies adhere to Fair Information Practice principles such as minimisation. In their work they discuss a language metamodel to create privacy requirements composed of statements concerning what data is collected by which entities, how such data is used and for how long the retention period is. Deontic modes are used to articulate whether such statements are permissions, obligations or prohibitions. Statements are associated with the following entities. Recipient elements determine who may receive information from the data processor. PrivateData elements illustrate what data is to be processed by the data processor. Service discusses the services of functionality that is provided to the data subject.

The metamodel is associated with three distinct syntaxes and grammars aligning with different usage contexts. The principle is to allow for the articulation of policies created or imported into the supported artefact to allow for the specification or validation of policies in different forms, visual, textual or tabular. The purpose of which is the creation of a tool that can operate with privacy policy specifications in multiple representations. Conflict analysis on this model leverages off the DL specifications provided by Eddy, where a policy either expressed natively, or transformed into RSL-IL4Privacy is in turn translated into a DL representation. The resulting DL output allows for the cross examination of prohibition and permissions for the purpose of validating policies [42].

It is intuitive to consider analysis of privacy policy for the detection of conflicts as paradoxically, despite the aforementioned difficulties with privacy policies, they remain an excellent source of information for data subjects looking to acquire insight in the data processing activities of a data processor. Naturally, such analysis techniques require the presence of complete, updated privacy policies to remain effective in their analysis. Where this work distinguishes itself from the literature is in the methodology to elicit conflicts. The approach discussed in this work does not require the provision of privacy policies. This is for three reasons. Firstly, the challenge with privacy policy is the frequency by which it can be updated. Being a legal document, the time requirement for appropriate legal teams to draft and approve new versions of a privacy policy can be significant. Conversely, the development of software systems tied to such policies handling data subject information can be realised much quicker than the legislative documents can be updated. Secondly, ambiguous language from the policy can challenge effective analysis in the event analysts have no way of identifying potential recipients of subject data. Third, an effective analysis requires understanding the prohibitions articulated in privacy policy. In the event policies do not include such prohibitions, the analysis will become challenging. This approach in turn operates on observable data flows to elicit the behaviour of data processors from the 'ground up' which does not require the need for analysis of privacy policy text.

2.3.2 Rule Based Analysis

Research aligning with the refinement of access control mechanisms often articulates privacy conflict as a phenomena where a series of rules are in a state of conflict with a resource request. Such work often focuses on marrying a distinct hierarchy of access permissions associated with individuals or groups to a technical enforcement architecture. Typically, such endeavours aim to assist stakeholders by proposing rule precedence to determine how conflict can be resolved from the perspective of access control rights management. Such initiatives are often used in the proposal, or refinement of underpinning semantics supporting the specification of privacy policies.

Early attempts at creating 'privacy aware' user clients can be seen in the Platform for Privacy Preferences Project (P3P). This initiative was one of the first engineering efforts to codify both privacy legislation and user privacy preferences into a XML based standard [43]. The objective of P3P was to streamline the privacy conflict detection process for end users. P3P enabled users to specify the information they were comfortable sharing when interacting with web services. Conversely, web services specify resources necessary from the client for the provision of services [44]. Conflict occurs when requested data from the web service exists in a prohibited list specified by the end user. In this instance, conflict resolution is delegated to the end user who determines whether they trust the system to access their data.

In the industrial sector, policy based access control models are used to instantiate privacy policy. The Extensible Access Control Model (XACML) and the Enterprise Privacy Access Language (EPAL) are two commonly used policy specification languages to standardise access control specification. Both standards operate on the ISO Abstract Policy Enforcement Model. Typically, privacy conflict is detected as a result of the Policy Decision Point returning multiple conflicting permissions (i.e. the authorisation decisions are not unanimous) associated with a given request. In such an instance, the Policy Enforcement Point has the responsibility to enforce arbitration to determine which authorisation request takes precedence in context of the request. The manner by which this is achieved (i.e. the precedence rules themselves), can be instance specific, defined in the business logic that leverages on the policy enforcement framework, or can leverage on standard precedence rules that are prescribed by the standard itself. XACML as an example operates as both a policy specification language and enforcement architecture, which presents a series of combinatory algorithms to determine which rules take precedence.

Ardagna et al discuss a method to extend the XACML architecture to create a hybrid architecture supporting the functionality of anonymous credential based systems for the provision of privacy. In their work they introduce the PRIME project architecture which targets legacy enterprise applications that utilise XACML policy specification for database driven access control. The principle behind the work is to support privacy preserving credential based authorisation in database systems where privacy preserving protocols are not natively supported. Their approach to privacy conflict detection and resolution leverages off the XACML methodology, where multiple conflicting decisions returned from the Policy Decision Point (PDP) are input into the rule precedence algorithms in the Policy Enforcement Point (PEP) for the purpose of resolving conflicts. It is the intention of the project to operate as a decision support tool for developers whom can work with the output from the enforcement module and apply it to their external business logic [45].

A common scenario with techniques employing policy based access control semantics is the challenge imposed by role hierarchies [46]. Several industrial standards leverage on the concept of roles in the semantics permitting rule checking to occur. In XACML and EPAL, users making requests exist as a role in the formalism, where permissions are typically mapped between attributes and roles on a system [46] [47] [48].

The problem with having users assigned multiple roles is the potential for conflicting decisions to be returned from a decision point. EPAL does not provide native support for resolving these conflicts unlike XACML which provides a series of 'hard and fast' precedence rules that can, but not always guarantee a resolution to the indecision faced with the enforcement point. A common research initiative focusing on XACML and EPAL semantics involves approaches to improve their expressive power and provide more sophisticated rules to handle the additional complexities of hierarchical role based permissions. The general aim is to improve legacy enterprise systems that do not natively handle hierarchical challenges. The majority of these approaches continue to leverage on rule bases (albeit more sophisticated) to determine how to resolve conflict.

Hoang and Son convey in their work a challenge with leveraging on rule precedence to resolve conflicts without additional decision making frameworks in place is the relatively binary output of the decision making process. They argue rule precedence frameworks, such as those observed in the industrial sector can be relaxed to output a 'softer' decision involving compromise. Their approach achieves compromise between conflicting access control permissions through a 'smoothing' approach permitting partial data access depending on the rights of the user and specific information being requested. The approach leverages on a series of privacy functions that are passed the permissions a user has along with the information being requested for the purpose of outputting modified results adhering to the fine grained permissions a user has. Additional rule precedence is used to determine which privacy functions take priority when multiple (conflicting) access rules are permissible.

Their approach to privacy conflict analysis is similar to the method seen in the abstract policy architecture, where the evaluation of a resource access request is the responsibility of the enforcement point. A policy is composed of security and privacy elements, where the security elements hold a series of preconditions causing the actuation of the associated privacy obfuscation functions. The privacy functions contain a mapping of actions to be performed on a given requested attribute. The aim of these functions is to provide varying levels of obfuscation to a requested resource to preserve the privacy policy of users. A privacy function may permit full, partial, or no access to an associated attribute. Each request made for a resource will involve the subsequent request for multiple attribute values from a database. The objective of their work is to evaluate and instrument a privacy obfuscation function on each attribute prior to the fulfilment of the request. The challenge of deciding which privacy function to enact over another in the event multiple privacy functions are applicable to a given request, is handled in the same vein as access control decisions in the XACML architecture. A series of rules are employed to determine which privacy functions take priority when multiple rules apply. The benefit this approach has over the standard XACML implementation is the attainment of compromise, instead of simply denying or granting the request fully to a user, in the event the enforcement point has not received unanimous clearance decisions from the decision point, partial data access can be granted through the obfuscation of certain attributes from the returned data set [49].

Work from Xia discuss the operational challenge posed with different cooperative industry bodies utilising access control policies specified in XACML [50] [51]. They discuss the potential conflicts arising from divergent access schemes when each organisation has multiple resources to manage. Their work leverages on symbolic model checking as a method for conflict detection. In their work they leverage off a Direct Acyclic Graph (DAG) to represent a hierarchical permission tree structure representing the different resources elicited from the XACML specification. The DAG representation is then used for the generation of Kripke structures which, combined with Computation Tree Logic (CTL) are used to determine conflicts between permissions associated with the nodes in a hierarchical manner. The conflicts detected from this method stem from role authorization and resource availability incompatibilities.

Ni et al discuss an approach to support the authoring and management of Role Based Access Control (RBAC) policies, with particular focus on the formalisation of usage context. Their approach extends classic role based access control models to allow for the expression of scenarios involving aspects of user privacy. Their work interfaces with the SPARCLE workbench to allow for a structured specification of Privacy-aware Role Based Access Control (P-RBAC) policies in natural language which in turn, allows for the translation of the policy into machine readable representation. They present a 4 step framework. SPARCLE is initially used in order to author the policies before the policies are parsed with permissions being assigned. Constructed permissions are analysed in order to detect inconsistencies as privacy conflicts. The classic RBAC model consists of users, roles, objects and actions. The core P-RBAC model extends this to include purposes, conditions and obligations to allow for the expression of necessary pre-requisites are formally articulated as context variables to express pre-requisites that need to hold in a given privacy permission. Constant condition and atomic conditions are paired in a context variable to express attributes relatable to privacy policy along with contextual information such as the requirement for user consent, along with atomic policy attributes. These formulate a condition language to express policy conditions.

An obligation model is introduced to formally articulate in P-RBAC those operations tied to the natural language policy. The principle behind the expression of obligations, and relations is to model instances whereby ambiguity exists on what actions should be taken, whre the RBAC model checked a series of permissions against an intended action and authorized multiple actions/obligations for the user associated with a particular role. Conflict in this work largely stems from the analysis of specified roles, permissions and obligations in the P-RBAC syntax. If two distinct permission assignment objects have similar object roles along with permitted actions, the conditions must be mutually exclusive. A series of outcomes are articulated in the work to delineate various contexts by which two permission assignments can be cross referenced. Conflict can occur in the event the permission conditions are mutually exclusive, or potential conflict can occur in the event some degree of ambiguity exists when permissions are compatible.

More research in the academic sector focuses on the instantiation of formalisms leveraging on metrics from social network domains that cannot be easily captured in traditional RBAC semantics [52] [53] [54]. Formal semantics within the RBAC paradigm typically operate on a series of metrics representing the relationship status between users and the sensitivity of the information they share on systems. Determining how to address the intended requests of users in social network settings presents different modelling challenges for privacy research, where an effective formalism requires unambiguous representations of concepts measurable in social networks.

Hu et al discuss an approach to improving access control approaches implemented on social network systems through a quantitative approach to measuring the privacy risk and sharing loss of a disclosure action. In their work a privacy policy is defined as a 4 tuple structure that contains the controller, accessor, data specifications and effect. A controller is a user that can be an owner, contributor, stakeholder or disseminator. An accessor specification details the individuals or groups that are granted access to a piece of information. A data specification is is a tuple containing either a user profile, friendship and consent paired with an information sensitivity value between 0 and 1.

Their approach to detecting privacy conflicts leverages on the concept of identifying users in an accessor space (a set of trusted users) where a user in the trusted accessor space for one agent is not trusted by another agent. These conflicting set items form what is described as conflicting segments. Their approach to resolving privacy conflicts is a systematic process, leveraging on the concept of measurable privacy risk. Privacy risk is measured as the possible privacy loss for a given data controller. Generally speaking, the privacy risk for a data controller is affected by the following aspects. The trust placed in the accessor agents, the sensitivity value, the cardinality of controllers that trust accessors found within the segment and the distance the information is disseminated. Each agent controller is assigned its own privacy risk value, whilst the aggregate value is used to represent the overall privacy risk found on the network.

The act of resolving privacy conflicts is determined by the manner by which privacy loss and sharing loss can be negotiated. Their approach, similar in concept to the notion of privacy calculus theory, aims to minimise the additional privacy loss whilst maximising utility. This is done by calculating the ratio of cost to utility for each conflict systematically, and if the sharing loss metric is lower than the estimated privacy risk the action is permitted, otherwise denied. The result of these actions will be used to generate a new accessors list to finish the sharing of user data [55].

The theory applied in privacy calculus involves the instantiation of a cost benefit function to output a utility score that can be applied in a larger decision making framework. In relation based settings, a common metric that affects this function is the information sensitivity, described by Aldini et al as ”a function of user expectations and context”. Information sensitivity as a consequence, is challenging to measure without user involvement and is typically treated as a configurable metric in several multi agent systems. Rios et al utilise information sensitivity values to annotate data resources to denote the level of granularity a piece of information is. In their work, they leverage on an abstraction model for information to assign an information sensitivity value to information at various levels of abstraction. The less abstract a piece information is, the greater the level of associated privacy risk the user is exposed to.

Information sensitivity and relation strength are often discussed as inter-relating concerns wherein the strength of ties between connected users on a social platform will in turn affect the perceived privacy risk of information disclosure. Typically the sensitivity of information is treated as an inverse function of the relation strength metric on a social network. Zhong et al discuss a method for detecting privacy conflicts from photo sharing systems on social networks. Their privacy classification model leverages on the concept of relation strength between the up-loader and recipients, along with the sensitivity of the content being shared to generate a privacy score [56].

It is intuitive to consider that the detection of rule violations can be used in order to express and quantify the concept of privacy risk. Fundamentally, the vast majority of contributions in the privacy conflict research space operate on rule bases in some form to identify instances of conflict. This work focuses on the articulation of object knowledge states which in turn are used to identify conflicts resulting from an incompatible manner by which knowledge states of actors evolve on a data flow network.

2.3.3 Behavioural Analysis

Research in social networks often focus on modelling privacy conflict as a tension between different actors within online social groups. Research in this category is often distinguished by both the granularity afforded to the modelling of conflicts, and the metrics involved in such models. Some work in this category focuses on the application and training of classifiers to understand normative behaviour of groups. In such instances, conflict is often articulated as a series of individualistic behaviours which differ from the established nomenclature of a collective. In other areas, more granular definitions of conflict often articulate the phenomenon as an intended action to be performed by an individual which will incur some degree of harm or otherwise violate the social contract with another. Rule bases, formed from intimacy analysis determine if such actions are a violation. This is often followed by a series of bartering algorithms to arrive at a consensus vote on whether such activities should go ahead. Such research is often positioned as a decision support system that operates on multi agent systems on social networks.

Negotiation approaches to privacy conflict analysis are typically referred to as decentralised decision making, where the outcome of privacy conflict is not the sole discretion of a single entity consulting precedence rules. Instead, the goal of negotiation approaches is to arrive at a consensus on what concessions are acceptable to the individuals within the group prior to disclosure taking place. Negotiation approaches are often implemented in multi agent frameworks, where each agent on the Multi Agent System (MAS) is programmed with local objectives and bias parameters. Agents that are considered rational, will operate according to their best interests. With respect to privacy, this involves attempting to safeguard the privacy requirements of users provided to them. In an interaction scenario, agents involved in a negotiation may be agnostic to the local objectives and prejudices held by those they interact with. Such interaction scenarios are typically referred to as adversarial interactions, where the observable behaviour of an agent is determined by how it perceives the possible interaction strategies affecting its own local objectives without consideration for others.

Conversely, it is also possible given an interaction scenario, agents will be aware of the local objectives and prejudices of those they interact with. These interaction scenarios are commonly referred to as cooperative settings, where an agents behaviour is motivated by not only their own objectives, but also of those with whom they interact with. The level of importance an agent assigns to its own objectives and those of others is implementation specific, and such weights are often treated as configurable metrics. From an engineering perspective, negotiation approaches are characterised by the implementation of decision algorithms stemming from economic theory. These algorithms require the implementation of fitness functions enabling agents to measure the benefit they will receive in the event a specific interaction strategy is agreed between all parties.

The benefit an agent will receive is often clarified, and subsequently expressed as a function involving the measured cost and utility involved with the execution of a strategy. Such functions allow for the output to be benchmarked against a threshold value which in turn serves as a fitness score. To relate back to the negotiation scenario, the fitness score is used by agents to determine how acceptable a possible strategy is, wrt the objectives they are catering for. The complexities involved in codifying fitness functions for each interacting agent on a MAS, along with the provision of individual programmable bias metrics has been argued to result in a more positive sum outcome in privacy analysis scenarios as opposed to more heuristic approaches.

Such and Rovatsos argue initiatives involving relation based metrics instrumenting centralised decision making may be problematic if implemented in social network scenarios involving multiple users. They clarify co-ownership scenarios may involve conflicting granule privacy policies. Therefore, having a centralised decision making system in place may lead to unsatisfactory outcomes. They argue the instrumentation of negotiation based solutions effectively remedies this problem and discuss their artefact to provide autonomous privacy conflict detection and operates on a consensus algorithm to resolve identified conflicts. For the detection of conflicts, Such leverages on intimacy values and action vectors. Intimacy values are floats determining different relationship thresholds between agents on a network. An action vector is a resulting function determining whether or not to grant or deny access to a resource dependant on whether the target agent matches the intimacy threshold held by an agent [57] [58] [59].

Their implementation operates on a single step negotiation strategy aiming to negotiate an agreed consensus within one round. All agents involved in conflict are contained in a set wherein the aim of conflict resolution is to negotiate a common action vector across all set members. Each agent within the conflict set ranks available option to them (i.e. each action vector is assigned a preference value to each agent) wherein the highest ranked possibility for an agent reflects the greatest utility score an agent has calculated. Such describes the utility function as being a delta measurement for the utility offered by an action vector under review and the action vector proposed natively. The delta values are specified with Euclidean distance between the intimacy metrics of two policies. The justification of this measurement stems from literature suggesting the strength (intimacy) of relationships informs the information disclosure behaviour of users on social networks. Each agent proposes a single strategy which is added to a pool, once all strategies have been proposed, the strategy which globally minimises the observable delta values will be implemented across all agents. To constrain the state-space of possible strategies, a series of heuristics are implemented utilising the sensitivity values to prune strategies that do not meet a configurable threshold value.

Negotiation acts as a manner to codify the provision of privacy concessions in a cost benefit analysis formalism. Generally speaking, a round of negotiation involves each agent proposing an offer best suiting their objectives from a state-space of possible strategies they can play. The artefact developed by Such is an example of an implementation strategy that involves only a single round of negotiation where the selected output is taken from one of the initial proposals submitted from all the agents [60] [61].

Such mentions in their work a stable approach for agents in single step negotiation is to select solutions maximising their individual utility values. In Game Theory, the concept of stability relates strongly to the probability of a single agent, or a set of agents modifying their behaviour in response to the potential gain or loss of utility in doing so [62]. In brief, a game is modelled as a series of strategies playable by an agent, where each strategy provides a degree of either positive or negative utility to that agent. A game is said to be stable in the event an optimal state of play is recognised by the agents, and there is no benefit to be had by any agent electing to execute a different strategy, and diverge from its selected strategy.

An alternative negotiation strategy involves multiple (k) rounds of negotiation wherein negotiation will terminate at k rounds. If unanimous agreement is not met after k rounds, it is the responsibility of the supporting business logic to determine how to select between the proposed strategies. Mosca et al leverage on a k-round negotiation approach to conflict resolution in their artefact that leverages on the concept of information sensitivity and relation strength to detect conflicts in a social network graph. In their work, a sharer of content and a recipient of content are not assumed to be directly connected on a social network, but rather their interactions are mediated and depend on a series of actors connecting them. In such scenarios, the concept of user trust and belief become more of a determining factor in the behaviour of users. They formally represent a policy as a tuple containing the maximum distance a user can be from the owner of the content being shared along with a threshold value for intimacy to be held on each path connecting the two users. Intimacy, measures the relationship strength between users and is represented as an edge on the social network graph. Distance is the hop count from a source to a target on a network.

Their work leverages on the Schwartz theory of values [63] to cluster agents based on observable behavioural trends. Self direction details agents willing to enter negotiation strategies. Power denotes adversarial agents not willing to accept compromise. Security denotes agents preferring to maximise the secrecy of their information. Conformity denotes agents which defer to the modal decision arising from a set of agents. Finally, Benevolence describes agents willing to defer to the judgement of their close friends, or to the judgement of others. The approach follows a series of negotiation rounds until a consensus is reached. An initial policy is proposed to a co-owner, who evaluates the proposition based on a utility function. If the policy is accepted then negotiation is over and the information is disclosed, otherwise the receiving party will propose a counter offer to the initiator. In the event no agreement can be reached, disclosure is abated and negotiation is considered a failure.

Manhattan distance determines the distance between a proposed policy and a given alternative a user has. When a policy is to be generated, such as in the case an alternative is to be presented, a proposal generator function provides the closest suitable alternative policy which best aligns with selected values. For this, the work leverages on value ordering (preference ordering) to determine how their agent fits into each category. The generator operates a series of value functions determining the suitability of a policy subject to each individual value listed and returns the most appropriate proposition. This is then used in the next round of negotiation, until either the proposal is accepted or negotiation is cancelled. The generation of privacy policies that align with the values of users they argue is the manner by which users can be kept informed of the processes followed in safeguarding their privacy needs [64].

The act of providing justification for executed actions is a concept routinely discussed in philosophical domains, but is also investigated in Computing Science. Dialogue Games are generally discussed in the literature as a protocol for interaction between smart objects involving the proposition and countering of arguments supported by logic based formalisms. Dialogue based games are typically employed in economic domains, where agents engage in dialogues with one another for the purpose of recommending autonomous purchasing decisions. However, some privacy literature has also investigated the utility of argumentation in approaches wherein agents attempt to make decisions based on logical arguments, or to present a trace history to users to provide justification for their behaviour, to increase operational transparency between users and software.

Kokciyan et al discuss an approach to conflict resolution leveraging on argumentation based negotiation on social networks to allow agents to exchange arguments in an attempt to reach a consensus. In their work, each agent is equipped with a social network ontology to model the privacy constraints contained within an agent. When an initiating agent wishes to disclose information pertaining to another agent, prior to disclosure proceeding, the agent creates a proposition which is supported with logical assumptions detailing the reasoning for disclosure. This argument is disseminated to the stakeholders of the information for examination. In the event an argument conflicts with the privacy constraints of an agent, a counter point can be constructed and conveyed or the assumptions composing the argument can be attacked. The final decision on whether or not to disclose information is determined by whether or not the argument to disclose is still considered valid after several rounds of argumentation. If the attacks do not invalidate a proposed argument, then it will hold.

In their work they present the PriArg framework composed of a four tuple structure containing the language, rules, assumptions and a mapping of contradictions between the assumptions and the language. Arguments are composed of statements from the language along with a claim (or assertion) which is supported with a series of assumptions. The process by which an argument is attacked, is by presenting a case contradicting one of the supporting statements of a claim.

Agents when evaluating the suitability of a received proposal can operate with either local or consulting knowledge, where an agent consults the understanding of other agents. When contradictions are made, the data structure is said to be extended, that is, the opposing case is added along with supporting statements. The back and forth argumentation continues until the the case status flag on the data structure is set to stop. The status flag is set to stop in the event that an argument cannot be further extended by an agent.

One interesting aspect of this work involves the discussion surrounding the dissemination of arguments themselves. As an example, Kokciyan et al specify the revelation of specific details on why an agent does not want an image disclosed may inadvertently cause privacy violations to occur. To counter this, they leverage on abstraction in their ontology. Predicate statements within the tuple data structure inherit from classes. Abstracting the argumentation is to present the parent classes in the argumentation as opposed to the more specific argumentation. Whilst the use of abstraction will help preserve privacy when extending the argument that has been received. The challenge with this solution is that agents may reject the argumentation if too abstract [65].

The idea of commodifying trust to encourage cooperation in a MAS is an idea explored by Kekülluoglu et al where they leverage on a reciprocal trust framework to conflict resolution. Their paper presents a hybrid negotiation architecture applying semantic representations of conflicts and the implementation of individual utility functions for the negotiation rounds.

Trust is explored in the work as an incentive to promote cooperation between agents for the provision of privacy protections.

Their work operates on a Web Ontology Language (OWL) to represent the privacy constraints of each agent connected on a network. Agents, when they wish to disclose information, make post requests to other agents where the request is evaluated according to local fitness functions. A distinction is made between the negotiator and the initiating agent. Initiators present the initial proposal for disclosure whilst negotiators are responsible for arguing for their privacy. If the initial proposition is met with unanimous agreements from all negotiators, then disclosure occurs, otherwise the initiator will attempt to cater to the rejection reasons presented by negotiators. The utility functions for both agent roles differ. For a negotiator the fitness score is calculated by taking the maximal utility against the summation of applicable privacy constraints multiplied by their weighting parameters. For initiators the fitness score is represented as the inverse of the number of agents removed from the original post request in response to objections from negotiators.

There are multiple negotiation strategies to resolve conflicts between the initiator and negotiating parties. The fair product strategy aims to normalise the values between the different utility values of multiple agents. The rationale for this is to prevent skewed or otherwise 'unfair' low utility values in the event that one agent is disadvantaged. The good enough privacy strategy involves multiple granule negotiations rounds wherein only a single reason for rejection is ever presented to the initiator from a negotiator. There are typically multiple rounds of negotiation involving balancing the utility concessions made by the initiator, and the utility gained by the negotiators. If no reasons for rejection are proposed after n rounds have passed, and utility thresholds are met, then disclosure commences, otherwise aborted. Finally, the maximal privacy strategy involves providing an upper bounds on the number of permissible negotiation rounds, where compound rejection reasons can be provided to an initiator with a final action being taken after k rounds.

Commodifying trust in this work leverages on a trust currency for agents to seek favour during negotiation. The principle is as follows, agents exhibiting cooperative tendencies by accepting concessions to their utility values, are in turn compensated with trust points. Such points, in future negotiations can be used to request concessions be made by those the agent interacts with. The greater the concession made to the utility values the greater the amount of points that can be earned. The points system serves as a trust metric such that the more points an agent has, the greater the indication of their cooperative tendencies [66] [67].

2.4 Summary

In this section we have outlined some key research themes aiming to measure privacy conflicts in sociotechnical systems. Firstly, we looked to unpack research to translate the natural language specification of privacy policies into formal interpretations. Some works leverage on formal logic such as Descriptive Logic to achieve this, whilst others leverage on descriptive ontology. In both cases, the objective is similar, to articulate privacy risk as a result of mismatches between parsed privacy policies and the behaviours observable within software systems. We touched on how insight can be acquired on the data processing activities of data processors through analysis of multiple privacy policies, which allows for the modelling of data processing activity without the requirement to measure such activity. We further unpacked a manner by which data flows can be elicited from code bases on client applications on the Android framework. We unpacked a series of works on the formulation of privacy policies such as EPAL and XACML. We discussed several research initiatives aiming to extend the expressive capacity of policy specification languages to model and resolve conflict. Finally, we unpacked a series of bartering approaches taken within social network contexts to analyse and resolve privacy conflict through the notion of compromise.

This work aims to model the data flow networks associated with data processors. Our methodology as discussed in section 5 leverages on the instrumentation of consent modules to elicit outbound requests to third parties to acquire an understanding of the data processing activities that occur as observable from a data subjects device. Further, given our approach to modelling the privacy expectations of data subjects by leveraging on Epistemic Modal Logic, we can leverage on this to articulate conflict as an outcome where the evolution of a data processors awareness of data subject attributes do not align with the perception of the data subject.

Chapter 3

Framework Overview

3.1 Introduction

In this chapter we aim to provide a brief overview of the solution proposed in this thesis to address the research questions posed in section 1. We start with an introduction to the solution concept, illustrating input, procedures and output. We delineate network elicitation, conflict analysis and refactoring as three distinct concerns of operation. Next, we unpack a logical mapping to illustrate how this framework targets the GDPR legislation. Specifically we describe some of the principles enshrined within the GDPR legislation that act as high level objectives we articulate as attainable goals through application of this solution. Finally, we unpack a usage context involving the main benefactor.

3.2 Solution Overview

Figure 3.1 illustrates a technical overview of the proposed solution in this thesis. The solution is articulated as a linear workflow to perform an assessment of the privacy risk exhibited on one, or multiple data controllers. The framework itself is composed of three key elements each encapsulating different concerns of the analysis. The first concern in the framework relates to the modelling of data flow activity between data controllers and external affiliates under the term 'web interactions'. The input into this process is a collection of URLs which can be sourced from data controller portfolios, or may contain a single data controller as input. This collection is used as input to the network elicitation process. This process aims to achieve two things:

1. Model the data processing activity involving any external data processors/controllers to those provided by the input dataset.

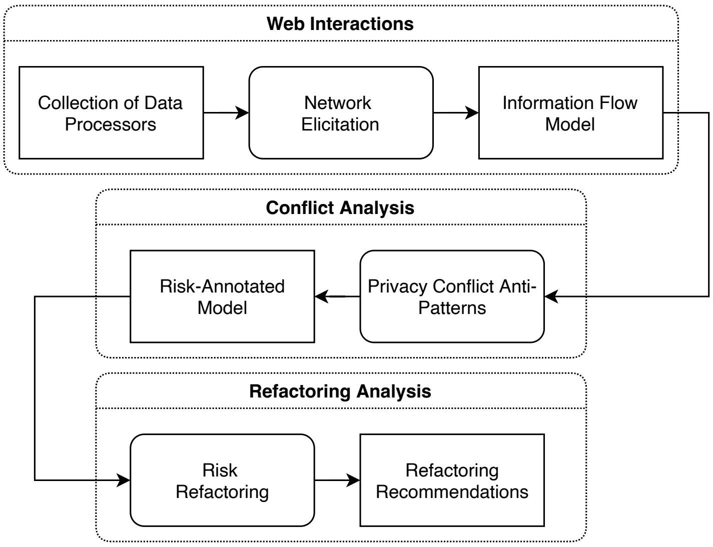


Figure 3.1: Solution overview proposed in this thesis.

2. Annotate the data processing activity to categorise specific data processing behaviours between data controllers and processors.

The solution operates on a Selenium workflow to identify a full network of data processing activity. This raw data is parsed to remove irrelevant entities. Relevance is determined through an external corpus providing insight into such entities involved in tracking data subjects online. The resulting network will model data processing activity between data controllers and external controllers/processors recognised as data trackers. We further annotate the model by classifying data processing behaviour. The classification process involves scrutiny of implemented CMPs on identifiable processors/controllers to understand the degree of control data subjects have when interacting with such entities online. For this work, we articulate the following three distinct behaviours:

- Mandatory: Where the data processing activity occurs irrespective of user choice and interactions with the data controller in turn will result in data processing occurring involving an affiliated controller/processor.
- Optional: Where the data processing activity occurs as a result of explicit consent being indicated by the data subject. Data subjects in such instances have the capacity to reject data processing activity on the instrumented mechanisms offered by the data controller.
- Covert: Where data processing activity occurs involving external data controllers/processors without any notice of levels of data processing transparency being provided to the data subject.

The significance of understanding the data processing activities on the information flow model concerns the levels of understanding the data subject will incur when interacting with instrumented CMPs. We formally articulate the knowledge acquisition of both data subjects and data controllers/processors through an awareness model. We apply privacy anti-patterns to the information flow model to identify where incompatible knowledge acquisition can occur on the information flow model. This in turn allows us to understand how the instrumentation of CMPs at design time can incur privacy risk to data subjects at runtime by virtue of their privacy expectations conflicting with the knowledge gain arising from data processing activity. Understanding what data processing activity leads to misaligned privacy expectations allows us to annotate the data processing model to articulate risk and where it is facilitated. finally, this annotated model is input into the final procedure where refactoring strategies can be explored to consider how a single data controller, or multiple data controllers can refactor their data processing activities to reduce privacy risk to data subjects.

3.3 Privacy Legislation Mapping

The targeted legislation in this research is the General Data Protection Regulation which provides a series of principles impacting the manner by which software systems are designed, operated and retired. The solution framework illustrates a series of technical mechanisms and methodology to (independently and holistically) cater to the following two specific operating principles of GDPR:

1. Accountability.
2. Lawfulness, fairness and transparency.

3.3.1 Fairness, Lawfulness and Transparency

Article 5 of the GDPR states that data subject information will be ”processed lawfully, fairly and in a transparent manner in relation to the data subject” [68]. This in turn is articulated as a guiding principle of lawfulness, fairness and transparency. The lawfulness of data processing activity relates to identifiable legal basis for data processing, which is a mandatory requirement of the legislation. Fairness relates to the justification for select legal basis and the behaviours of controllers and processors involved in the elicitation of data subject information. Finally, transparency concerns the mechanisms in place to appropriately communicate with data subjects the intended data processing activity that occurs.

Figure 3.2 illustrates a mapping between the two key GDPR principles targeted in this work along with a breakdown of such principles. The web interactions workflow relates to the

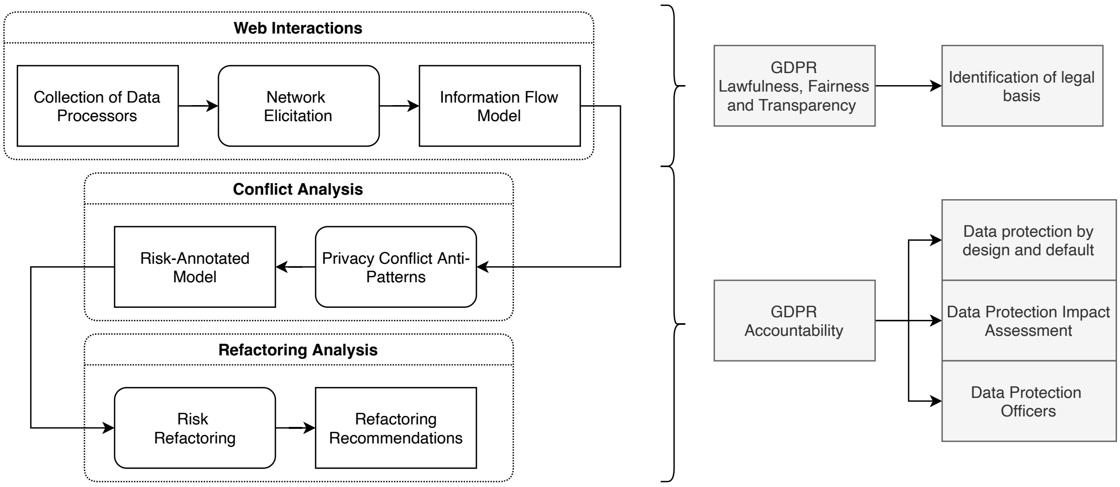

Web Interactions
Collection of Data Processors
Network Elicitation
Information Flow Model
Conflict Analysis
Risk-Annotated Model
Privacy Conflict Anti-Patterns
Refactoring Analysis
Risk Refactoring
Refactoring Recommendations
GDPR Lawfulness, Fairness and Transparency
Identification of legal basis
GDPR Accountability
Data protection by design and default
Data Protection Impact Assessment
Data Protection Officers

Figure 3.2: Solution overview mapped to GDPR principles.

steps that can be taken by analysts to model the data processing activities that occur on a single, or multiple data controllers involving external controllers and processors. The point of such modelling is to understand how CMP mechanisms are instrumented. Given the control and transparency afforded to data subjects relates with the selected legal basis for processing, scrutiny of such mechanisms in turn will allow for the identification of instantiated legal basis for data processing in the data subject facing CMPs. Furthermore, analysis of instrumented CMPs in eliciting the information flow model affords us the capacity to identify potential non-compliance owed to the omission of minimum levels of data processing transparency mandated by the lawfulness, fairness and transparency principle. This is an emergent contribution of the work that can be used by analysts to identify ways to better demonstrate compliance through instigating appropriate notice or consent mechanisms through CMPs.

3.3.2 Accountability

Secondly, the principle of accountability is encapsulated in the conflict and refactoring analysis workflows. Accountability in GDPR is a suite of concerns involving the demonstration of technical processes to safeguard user privacy in software systems. Accountability is the main targeted principle in this work, given the purpose of our solution is to first model privacy risk before investigating refactoring solutions to reduce privacy risk to data subjects. This in turn affords data controllers the capacity to demonstrate accountability in the evaluation of privacy risk facilitated by their data processing to in turn, safeguard user privacy. There are a number of sub-principles related to the notion of accountability we consider for our proposed solution.

Data Protection by Design and Default

Article 25 of the GDPR stipulates data controllers should ”integrate necessary safeguards into the processing in order to meet the requirements of this regulation and protect the rights of data subjects” [69]. This operating principle of data protection by design and default strongly relates to the engineering paradigm of Privacy by Design [70]. The premise is to treat privacy concerns as first class entities in the conceptualisation and development of software systems. This principle encourages a holistic review of planned software systems and their impact on privacy requirements during the development life-cycle of software. One key component when demonstrating compliance with this principle is the requirement for user preferences on data processing activity to be respected. In the conflict analysis workflow, the definitions of privacy risk relate to scenarios where previously indicated preferences on data processing are undermined by subsequent data processing facilitated by divergent CMP instrumentations, or incompatible legal basis for data processing. This in turn facilitates the data protection by design and default principle of accountability, given this work affords analysts the capacity to model where, within a series of data processing activities, how the indicated preferences of data subjects may not be respected by other controllers or processors.

A second aspect of this principle concerns the selective involvement of external systems in data processing that exhibit appropriate data protection mechanisms. One of the aims of the conflict analysis workflow is to identify instances where data controllers are involved in dissemination activities with external controllers or processors that do not exhibit any legal basis. This in turn allows controllers to identify potential affiliates that may introduce unnecessary privacy risk to data subjects when the objective of such analysis is for data controllers to understand their own risk profiles for data subjects. The conflict analysis workflow again is positioned to provide such information to data controllers. Further, the refactoring analysis workflow provides controllers with insight on how to best safeguard the privacy needs of users by exploring a series of refactoring options on data processing activity to reduce observable privacy risk to data subjects.

Data Protection Impact Assessments

Article 35 of the GDPR states ”the controller shall, prior to the processing, carry out an assessment of the impact of the envisaged processing operations on the protection of personal data” [71]. Such analysis often involves the assessment of the data collection and processing methods involving both internal mechanisms on a data controller, and external data controllers or processors to illustrate perceivable risk to data subjects.

There are three main concerns when implementing a DPIA. First is the intention for business to understand how suited their business practices are in upholding data protection of data subjects. In this work through the conflict analysis workflow we focus on the scrutiny of external data processing activity involving external data processors and controllers under the perspective that incurred privacy harm to the data subject arises owed to a lack of meaningful choice or control mechanisms implemented by a pair of data processors or controllers. An additional concern relates to the opportunity for refinement of data processing activity on a business process model to better demonstrate compliance.

The refactoring analysis workflow is intended to provide a series of recommendations on the optimal way by which data processing activity can be refactored in order to maximise the reduction of observable privacy risk on a given data controller. This analysis can be used to understand how to refactor data processing of a single controller, or multiple controllers.

Additionally, data controllers can utilise DPIA as a means by which they can demonstrate regulatory compliance with GDPR. Typically, DPIA process is often an activity carried out on the behalf of a data processor by a Data Protection Officer either an appointed or contracted for the analysis. The solution framework in this thesis in turn can be utilised to assist DPOs in effectively scrutinising the external processing activity of data controllers. The three workflows correspond to the necessity to model, analyse and evaluate data processing on systems before concluding the DPIA report for the appropriate stakeholders.

Data Protection Officers

Article 39 of the GDPR legislation sets out the specific tasks to be undertaken by a Data Protection Officer either appointed, or contracted by a data controller [72]. They are positioned in the legislation as the main contact for organisations looking to conduct DPIA process, and to provide advice for data controllers on how to better demonstrate compliance. According to GDPR, a DPO can be appointed jointly between data controllers, or operate with a single controller. The DPO is the main stakeholder targeted in this work, with the DPIA process the targeted methodology. The solution framework aims to holistically operate as a decision support tool to assist DPOs make informed choices on how to advise data controllers in the refactoring of their data processing activities to reduce risk if such an objective is the goal of the analysis. As previously discussed in section 1, the main challenge we aim to address in this work to directly benefit DPOs is the capacity to objectively quantify risk and utilise this risk metric to evaluate and rank different refactoring outcomes to stakeholders involved in the decision making process for a data controller.

3.4 Stakeholder Usage Context

Voxmedia[1] is an example of a data controller that possesses a portfolio of multiple data controllers inclusive of The Verge[2] and New York Magazine [3]. In the event any controllers elicit or otherwise process data subject information on behalf of Voxmedia, according to GDPR such controllers within the portfolio of Voxmedia will be identified as data processors per Article 4 of GDPR [73]. This in turn means Voxmedia as the managing data controller is liable for the data processing activity involving processors identified on its portfolio and any external data processors involved externally on that portfolio. Therefore, it will be of interest to Voxmedia to scrutinise such data processing activity for the purpose of reducing privacy risk posed to data subjects, or to otherwise certify compliance with GDPR.

The workflows positioned in this thesis aim to cater to a usage context, such that a DPO is contracted for DPIA by a data controller responsible for the data processing activity on its portfolio of processors. This means that DPOs will have access to CMP platforms instrumented by the data processors on the controllers portfolio. The solution framework will take as input the URLs of each data processor contained on the portfolio of the managing controller, and the subsequent workflows will provide the DPO with output regarding the detected privacy risk resulting from CMP instrumentations, along with a series of optimal refactoring suggestions to reduce that risk value. Given the processors analysed on the portfolio of the controller are owned by the controller, the workflow accounts for the mutability of data processing activity that can be recommended by the DPO to individual data processors for the controller to improve their compliance with GDPR.

3.5 Summary

In this chapter we have briefly illustrated the solution framework positioned in this thesis, before discussing the relation with GDPR legislation. We discussed how the Fair Information Principle is supported due to the solutions capacity to identify both legal and non-existent legal basis for data processing. Next, we unpacked three sub-concerns of accountability. Data protection by design and default is catered to through the conflict and refactoring analysis workflows, illustrating where risk is introduced through external data processors. DPIA is the analysis methodology this work aims to support whilst DPOs are the intended benefactors to apply this work as a decision support tool in their analysis. Finally, we unpacked a use case involving data controllers and their interactions with the data processors they are liable for on their portfolios.

[1] `https://corp.voxmedia.com/`

[2] `https://www.theverge.com/`

[3] `https://nymag.com/`

Chapter 4

Modelling Privacy in Software

4.1 Introduction

In this work we define privacy as the capacity for users to regulate the manner by which the knowledge states of others evolve. This perception leverages off Communication Privacy Management Theory which (in brief) discusses the concept of perceived end user control over personal information through the establishment of information disclosure rules with actors they interact with [74] [75]. In an online setting, the interactions a data subject has with first parties online are privacy preserving in the event the knowledge acquisition of data processors aligns with end user expectations. Conflict occurs in the event there is a misalignment between the acquisition of knowledge and the expectations held by data subjects. We leverage on Epistemic Modal Logic in this work to model privacy concerns of software objects. This is because Epistemic logic allows us to formally represent the acquisition of knowledge. Given our philosophical definition of privacy concerns the regulation of knowledge, an approach facilitating the reasoning of knowledge makes Epistemic logic a compelling choice for our requirements. In this chapter we aim to address research question 1 by unpacking two formalisms. Firstly, we introduce our awareness model to computationally express the privacy concerns of data subjects. Secondly, we articulate a series of privacy conflict semantics.

We begin our discussion in section 4.2 with an overview of Epistemic modal logic and possible world semantics that support the awareness formalism discussed in section 4.3. The use of Epistemic modal logic allows us to formally express the acquisition of knowledge, which strongly aligns with our philosophical views on privacy. Next, in section 4.4 we articulate the context by which the memory state of an object will evolve and discuss the nature of tenability. We communicate how accessibility relations are used to determine the tenability of assertions for an object. Next, we unpack three different semantics to formally articulate conflict in section 4.5. We discuss explicit, implicit and uncertainty conflict formalisms.

We conclude this chapter in section 4.6 with an outlook on the advantages of this approach, before instantiating the formalisms in a case study in later sections of this thesis.

4.2 Supporting Logic

Knowledge models are generally implemented within Multi Agent Systems as a method to represent the local knowledge states of objects on a network for the purpose of evaluating actions of outcomes which can influence held objectives. In the literature, there are numerous agent based reasoning models supported with a variety of formalisms and semantics.

Modal Logic is a branch of logic for the formal expression and evaluation of the validity of propositions regarding logical possibility and necessity [76]. Propositions in Modal Logics are statements of fact concerning the environment that can be evaluated as being true or false. Modal logic is composed of a series of formalisms consisting of modal operators, syntax rules and logical axioms for the analysis of logical statements. In Modal Logic, there are two well studied modal operators to qualify logical propositions. The necessity operator ($\Box$) when combined with a proposition p generates a modal formula to expresses necessity. The modal formula expressing ”it is necessary that p” is illustrated in formula 4.1:

$$\Box p \tag{4.1}$$

Conversely, the modal operator $\diamond$ when combined with a proposition will generate a formula to express possibility. Formula 4.2 illustrates the modal formula expressing ”it is possible that p”.

$$\diamond p \tag{4.2}$$

In the literature, Modal Logic is discussed as a family of logics from different philosophical paradigms. Epistemic Logic is a branch of Modal Logic for reasoning about knowledge [77]. Whilst Modal Logic deals with reasoning about necessity and possibility, Epistemology deals with reasoning about knowledge and belief. Epistemic Logic (as a variant of Modal Logic) shares a lot of characteristics with Modal Logic, formalising a series of modal operators, syntax rules and axioms for the purpose of reasoning about statements regarding knowledge.

Epistemic Logic allows for the expression and reasoning about individual and collective levels of knowledge. In computing science, Epistemic Logic is commonly associated with agent based reasoning, where a single agent or a collection of agents can hold some attitude towards a given proposition commonly expressed as ψ [78] [79] [80]. The modal operator for knowledge in Epistemic Logic is K and when combined with a proposition ψ generates an epistemic formula regarding agent knowledge. An additional requirement in the expressive syntax of epistemic formulae is a subscript operator indicating the specific object that holds the attitude. The formula below illustrates an example statement "Joe knows ψ".

$$K_{Joe}\psi$$

The second operator in Epistemic Logic concerns the belief of an agent, represented as B. When this operator is combined with a proposition ψ the resulting formula expresses the belief an agent possesses. The formula representing the statement "Joe believes ψ" is illustrated below.

$$B_{Joe}\psi$$

The logical axioms are a series of well studied logical rules to evaluate the logical consequences of propositions. Each axiom exists as its own formulae used to deduce logical consequence, or can be combined with additional axioms for more complex logical evaluations [81]. Current research in schools of philosophy investigate the development of new and the refinement of existing axioms. Providing an exhaustive discussion on the different logical axioms is beyond the scope of this work, but we will discuss later in section how we leverage off some select axioms from Epistemic Logic for conflict detection.

Possible worlds are a common supporting semantic for various forms of Modal Logic [82]. Informally, a possible world serves to represent a way by which the truth or falsehood of a given proposition, or collection of propositions may be. As an example, the epistemic proposition "John is in his office" can be represented in two possible worlds, one world where it is the case John is in his office, and another where it is not the case John is in his office. Each world in this example expresses a mode, a possibility regarding the truth of the proposition being considered [83] [84]. It is acceptable for possible worlds to hold additional propositions; For instance, we could expand the example to include additional propositions regarding the current weather and traffic conditions, creating a collection of propositions that will in turn be represented in different possible states in the worlds that are created. For ease of illustration, we will constrain the example to a single proposition.

Formally, Kripke semantics are used to provide a structured approach to modelling possible worlds [85]. A Kripke structure is a tuple $< W, R, V >$ where W is an enumerated set of possible worlds such that $w_i \in W$. R is a set of binary relations on W and V is a valuation function. The purpose of the valuation function is to determine the subset of W such that each world within the subset satisfies the valuation function.

In possible world semantics, worlds are connected to one another through directed accessibility relations. How accessibility relations are formed, and how they are utilised is largely dependant on the perspective taken on what the accessibility relations indicate. Generally speaking, accessibility relations have different interpretations in different logical contexts.

From an epistemic perspective, indistinguishability relations are used to represent a collection of worlds that an agent cannot distinguish between based on the current level of knowledge the agent possesses [86]. An agent is said to exist within one of the worlds (commonly referred to as the actual world, or current world) which best reflects the current understanding the agent has regarding the propositions modelled in that world. Other worlds are said to be indistinguishable to the actual world when the knowledge available to the agent is insufficient to eliminate those worlds based on the provided truth and falsehoods regarding the propositions. Indistinguishable worlds from a current world are said to be accessible from the current world. The accessibility notation is as follows:

$$wR_aw'$$

Where the subscript a indicates which agent the relationship belongs to. As an example, consider the following two propositions p and q where $p =$"John is travelling" and $q =$"John is in a taxi". The possible world set W will look as follows:

$$W = \{w_1 = \{p, q\}, w_2 = \{p, \neg q\}, w_3 = \{\neg p, q\}, w_4 = \{\neg p, \neg q\}\}$$

In this example let's assume our agent has acquired knowledge which allows him to conclude John is indeed travelling (thusly, p). If the current world for the agent is w_2, the knowledge available to them only indicates that John is travelling. With this information, it is not known if John is travelling via taxi or not. This means w_1 and w_2 are indistinguishable from one another, as the agent does not have enough information to be able to eliminate either possibility regarding proposition q. As far as the agents understanding permits, both q and $\neg q$ are equally tenable, therefore we arrive at:

$$R = \{w1\ R_a\ w2, w2\ R_a\ w1\}$$

However, there is sufficient information to eliminate $w3$ and $w4$ when $w2$ is the current world, as it is known John is travelling. Therefore the proposition $\neg p$ does not make sense for the agent to consider. The resulting accessibility relations between $w1$ and $w2$ are mutual, where both worlds are accessible to each other.

From a Modal Logic perspective, determining accessibility relations involves the evaluation of modal formulae in a given possible world with respect to the other worlds in W. In this interpretation of accessibility, each possible world contains a modal formulae, evaluated according to the propositions existing within the possible worlds. As an example, consider the set of possible worlds below:

$$W = \{w_1 = \{p, q\}, w_2 = \{p, \neg q\}, w3 = \{\neg p, q\}, w4 = \{\neg p, \neg q\}\}$$

Let's assume the modal formula $\Box p$ is associated with $w1$. Accessibility relations in this instance will be determined by the worlds satisfying the modal formula. This is an example of how the developed axioms from Modal Logic can be used to illustrate what worlds are accessible from $w1$. System K is an axiom that operates as follows:

$$\Box A \to A$$

The axiom stipulates whatever is necessary is true. We can see how this axiom can be used in the above illustration of possible worlds along with the modal formula to determine which worlds satisfy the axiom. We can see in $w1$ both propositions are true, therefore the modal formula $\Box p$ is satisfied in $w1$ as p. The same applies for $w2$ but $w3$ and $w4$ do not satisfy $\Box p$ as $\neg p$, which defies the axiom K used for the evaluation. This will result in the following accessibility relations:

$$R = \{w1 \; R \; w1, w1 \; R \; w2\}$$

In practice, there can be multiple modal formulae associated with multiple worlds and as such, the resulting accessibility relations will reflect this. The example given here focuses on the relations between $w1$ and $w2$ for ease of illustration.

The capacity for Epistemic Logic to model an agents understanding and ignorance of propositions makes it a suitable choice for modelling privacy concerns under the interpretation of privacy being the regulation of knowledge evolution.

Functioning as a knowledge model, possible world semantics provides an approach to exhaustively model the different possible states the propositions may be in (wrt their truth values), and as such will constitute a possible world state-space regarding a proposition (or series of propositions) an agent is considering. The accessibility relations in turn will enable us to represent which epistemic possibilities an agent is considering. Leveraging on the concept of indistinguishability, an agent who is uncertain of some information regarding a data subject will in turn consider equally, the two epistemic possibilities p and $\neg p$ as being tenable to them which can be computed through the establishment of accessibility relations.

4.3 Software Awareness

In this work, we leverage on Epistemic Modal Logic and accessibility relations in possible world semantics as a logical tool to illustrate an agents inability (or capability) to distinguish between epistemic possibilities regarding a proposition [87] [88] [89]. This in turn is used to evaluate an awareness function to determine whether an agent is in a state of awareness or unawareness regarding the proposition we model as ψ.

An awareness instance is defined as a modal operator A which is associated with a principle object p and optional reference objects $r1$ and $r2$. Like Epistemic Logic, an awareness instance modal operator possesses a subscript containing one of the object operators $(p, r1, r2)$ to indicate awareness instance ownership. The awareness instance class represents an objects awareness state regarding a proposition ψ.

An unawareness instance conversely is defined as a modal operator $\neg A$ which is associated with a principle object p and optional reference objects $r1$ and $r2$. An unawareness instance also possesses a subscript containing one of the object operators to indicate ownership. The unawareness instance class represents an objects unawareness state regarding a proposition ψ.

Both awareness and unawareness instances are the result of an awareness function that operates on the possible worlds that comprise the memory of the principle object. The memory of an object p concerning a proposition ψ is formally represented as $M(\psi) = (W, p, R, I, w_c)$ where:

- W is a set of possible worlds each considering a unique viewpoint on ψ.
- p is the principle object whose memory is being modelled.
- $R \in 2^{su,s,r}, \forall su \in SU$ is a set of reference objects whose (un)awareness about ψ may be considered by the principal. Each subject, sender or recipient may assume the role of a principal and/or a reference object.
- $I \subset W \times W$ is the accessibility relation on W.
- w_c is the current world of p.

The current world w_c and the accessibility relations that exist in I are used to determine the resulting state of (un)awareness an object possesses regarding ψ. The simplistic scenario illustrates ψ as an atomic piece of information, such that the information being reasoned about cannot be expressed in a more granular manner. In our awareness model, atomic propositions are labelled f in distinction from composite propositions ψ involving the reasoning of the awareness states of other objects.

Considering the scenario where the principle object is considering its understanding of an atomic proposition f, the set of possible worlds will look as follows:

$$W = \{w_1 = \neg f, w_2 = f\}$$

Additionally, the accessibility relations on W are modelled as follows:

$$I = \{w_1 \; R_p \; w_1, w_2 \; R_p \; w_2, w_1 \; R_p \; w_2\}$$

The agents inability to distinguish between the proposition f is dependant on the current world w_c being w_1. In such a case, the principle considers the reflexive relationship at w_1, along with the asymmetric relation between w_1 and w_2. In this case, both propositions (on both possible worlds) are considered possible with equal tenability, therefore the resultant of this will be an unawareness instance classification, such that the principle is unaware of f [83]. This is represented in the following modal formula:

$$\neg A_p f$$

Conversely, an agents capability to distinguish the proposition f between w_1 and w_2 leverages on the current world w_c for p being w_2. In this case, according to the accessibility relation set I we can see that from the perspective of w_2, the principle only considers the reflexive relation $w_2 \; R_p \; w_2$ and as a consequence, is said to be able to distinguish between worlds w_1 and w_2 regarding f. This in turn results in the creation of an Awareness instance classification, where the principle is said to be aware of f. This is represented in the following modal formula:

$$A_p f$$

When ψ is disclosed between objects, the tenability of (un)awareness instance classifications will shift as a result of component interactions from a system design. We assume the objects possess limited rationality, such that their understanding can mutate gradually from a state of full unawareness to a state of full awareness of a proposition.

The complete memory bank of a principle consists of a series of awareness classifications which prescribe and constrain the nature of the principles reasoning. We define full unawareness regarding a proposition as a state where the principal holds unawareness instances regarding all of the awareness classifications. Conversely, a state of full awareness consists of an agent holding a series of awareness instances regarding all awareness classifications. Formally, we define the different awareness classifications as follows:

Awareness classification 1 (henceforth, $A1$), aims to represent the principals (un)awareness regarding an atomic proposition (therefore $\psi = f$). As this awareness classification deals with the principle only there are no reference objects, therefore $R = \emptyset$. For $A1$ we consider W to be a set of two possible worlds, illustrated in figure 4.1.

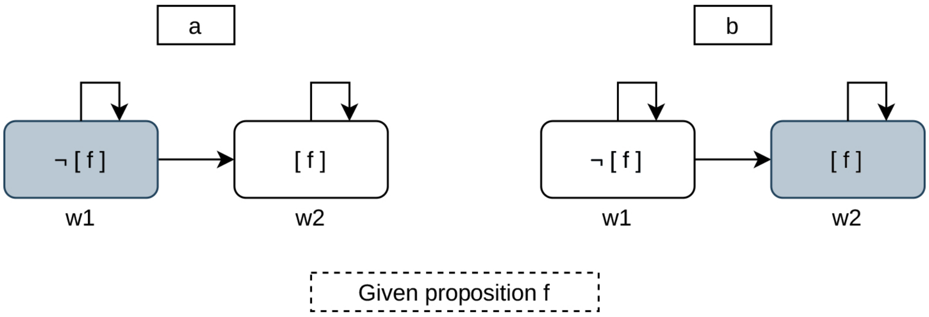


Figure 4.1: Possible world state-space for awareness classification 1

We can see in figure 4.1 that $W = \{w_1 = \{\neg[f]\}, w_2 = \{[f]\}\}$ where in $w1$ the proposition f is not considered to be possible and in w_2 the proposition f is considered to be possible. The square brackets in this representation serve to illustrate the proposition (whether atomic or composite) existing within the two possible worlds, one where it is not the case the proposition holds, and another where it is the case the proposition holds (as indicated by the negation prefix in figure 4.1). It is not the proposition itself that is being negated. The current world w_c for the principle object is illustrated as the shaded world. The accessibility relations between the worlds as $I = \{w_1 \; R_a \; w_1, w_1 \; R_a \; w_2, w_2 \; R_a \; w_2\}$ In scenario a) $w_c = w_1$. In a), the principle is unaware of f as a result of the accessibility relations connecting the current world to w_2 meaning the principle cannot distinguish f from both w_1 and w_2. Conversely, in b) we can see that when $w_p = w_2$ the principle only considers w_2 accessible and is therefore aware of f. When reasoning about f in $A1$, the memory bank of the principle object stores both the awareness and unawareness instances, therefore:

$$A1_p = \{\neg A_p f, A_p f\} \; and \, \forall a \in A1_p, a \in M_p$$

Awareness classification 2 (henceforth $A2$) allows the principle to consider its understanding of a single reference objects awareness or unawareness of f. In this classification the proposition is considered composite (thus ψ) as $A2$ allows a principle to reason about instances of $A1$, as opposed to atomic propositions. In $A2$ the reference object set R contains either the subject, sender or recipient of f such that consecutive object roles[1] cannot refer to the same object. For instance, the awareness statement A_pA_pf is not permitted.[2] The possible (un)awareness instances that compose the memory of the principle in $A2$ are generated as follows. Firstly, we take the potential reference object r and generate the possible (un)awareness instances wrt the atomic proposition f. Therefore, the generated $A1$ classifications with respect to reference object r will look as follows: $A1_r = \{\neg A_rf, A_rf\}$. The second step is to treat every instance of $A1_r$ as the proposition ψ (as this is the proposition p is either aware, or unaware of) and utilise the same heuristics to generate the possible world state-space, as was utilised for $A1$. This will result in the following possible worlds as illustrated in figure 4.2

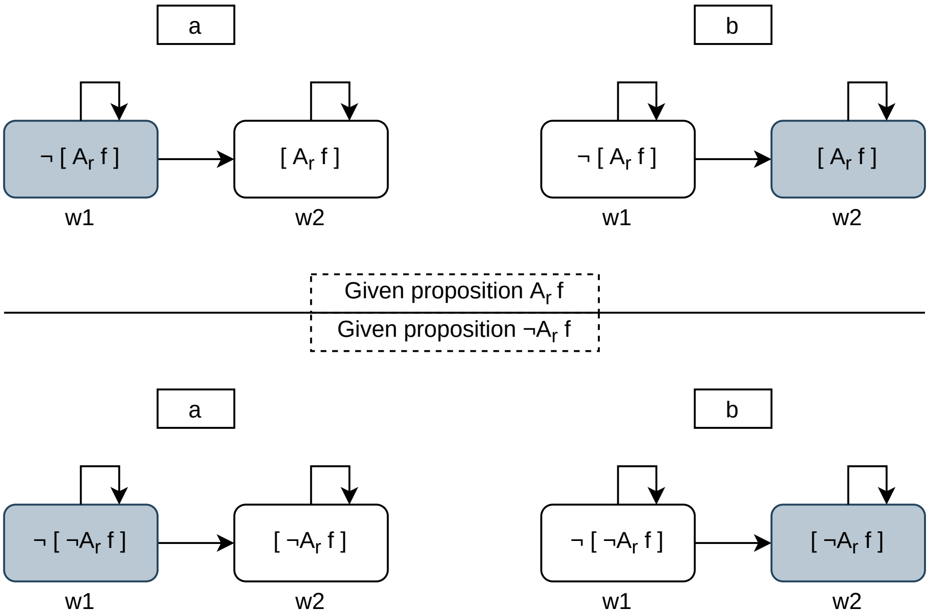


Figure 4.2: Possible world state-space for awareness classification 2

In figure 4.2 a), we can see an example current world w_c in the shaded world. In this instance, the resulting instance will be unawareness, as both w_1 and w_2 are accessible to the principle p. Conversely, the example b) illustrates the resulting awareness statement when $w_c = w_2$. Scenarios c) and d) illustrate unawareness and awareness respectively, where the proposition $\psi = \neg A_rf$. The memory banks of a principle object wrt instances of $A2$ will constitute of all (un)awareness instances regarding ψ, therefore $A_pA_rf, A_p\neg A_rf, \neg A_pA_rf, \neg A_p\neg A_rf$ will exist in M_p, therefore:

[1] Additional discussion on role modelling which is out of scope in this work can be found in [90]

[2] Certain axioms in Epistemic Modal Logic discuss the concept of reflexive knowledge, however for our purposes this is not a necessary consideration as such representations are redundant.

$$A2_p = \{\neg A_p A_r f, A_p A_r f, \neg A_p \neg A_r \neg A_r f\} \; and \, \forall a \in A2_p, a \in M_p$$

Awareness classification 3 (henceforth $A3$) further enables a principle object to consider the (un)awareness a reference object may hold regarding the principal. In $A3$, the proposition ψ is again composite, where ψ is equivalent to instances of $A2$. $A3$ instances contain a reference object and the principal object where the reference object may be the subject, sender or recipient on the condition that the principal object and the reference object r do not hold the same roles. The full set of $A3$ instances are generated utilising similar heuristics as for $A2$. Firstly, we generate all possible instances of $A2$ for the reference object, thusly:

$$A2_r = \{A_r A_p f, A_r \neg A_p f, \neg A_r A_p f, \neg A_r \neg A_p f\}$$

Having arrived at $A2_r$ the next step is to iterate over all the instances, treat them as ψ and generate the possible world state-space for each instance of ψ in $A2_r$. We will arrive at a set of possible worlds along with a set of accessibility relations as shown in figure 4.3.

We can see in figure 4.3 the 4 possible world sets for each instance of ψ. Again the same accessibility rules are applied to indicate whether the principle is in a state of unawareness or a state of awareness. As an example in a), the current world of the principle $w_c = w_1$, as such, both worlds are considered accessible to the principle because of $I_p = \{w_1 \; R_p \; w_1, w_1 \; R_p \; w_2, w_2 \; R_p \; w_2\}$ and thusly, is said to be unaware of the proposition as a result of indistinguishability rules. Again the memory banks of the principle object will contain all possible (un)awareness instances for ψ for $A3_p$, therefore the following is added to M_p:

$$\begin{aligned} A3_p = \{&\neg A_p \neg A_r A_p f, A_p \neg A_r A_p f, \neg A_p A_r A_p f, \\ &A_p A_r A_p f, \neg A_p \neg A_r \neg A_p f, A_p \neg A_r \neg A_p f, \\ &A_p \neg A_r \neg A_p f, A_p A_r \neg A_p f\} \; and \, \forall a \in A3_p, a \in M_p \end{aligned}$$

Finally, awareness classification 4 (henceforth $A4$) allows for the principle object to reason about the references objects (un)awareness state regarding a second reference objects (un)awareness state, thusly the proposition ψ for the principle . The two reference objects in $A4$ instances are an ordered pair (r_1, r_2). For this final class, the instances of $A4$ will contain propositions equivalent to instances of $A2$. The full set of $A4$ instances are generated by first

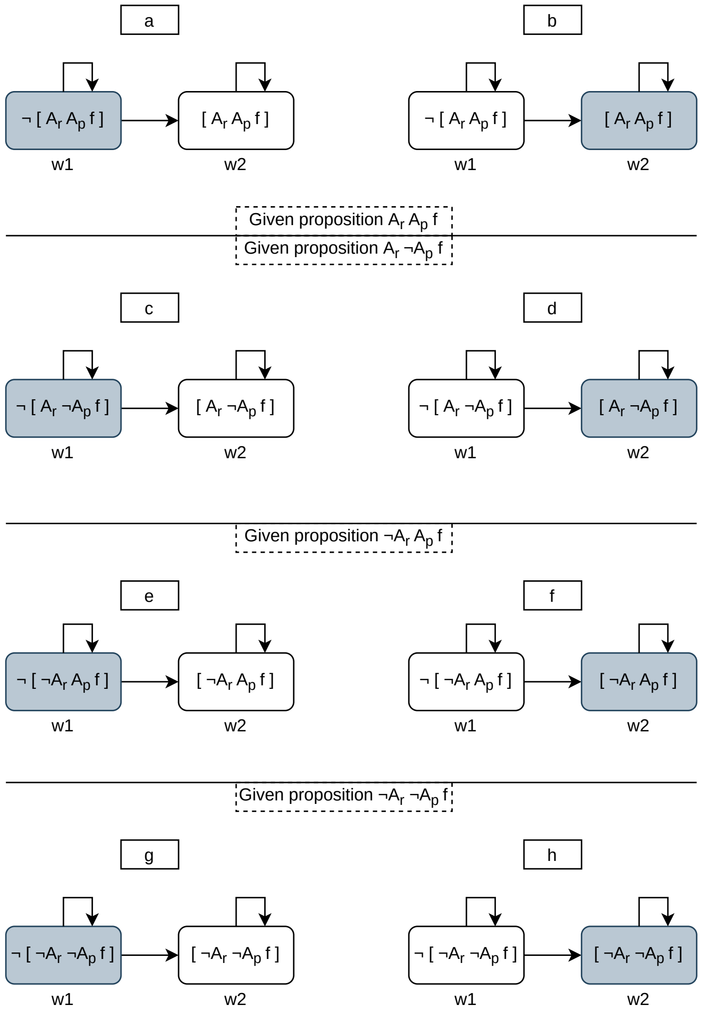


Figure 4.3: Possible world state-space for awareness classification 3

generating the $A2$ instances for $r1$ and $r2$ respectively. This will yield two sets $A2_{r_1}$ and $A2_{r_2}$ such that:

$$A2_{r_1} = \{\neg A_{r_1} A_{r_2} f, A_{r_1} A_{r_2} f, \neg A_{r_1} \neg A_{r_2} f, A_{r_1} \neg A_{r_2} f\}$$
$$A2_{r_2} = \{\neg A_{r_2} A_{r1} f, A_{r_2} A_{r1} f, \neg A_{r_2} \neg A_{r1} f, A_{r_2} \neg A_{r1} f\}$$

Again, by employing the same heuristics as before with $A3$ instances, we iterate over each element in both sets and create a possible world for each element within the set. The resulting possible world state-space is shown in figure 4.4

All possible resulting instances of $A4_p$ are finally added to the memory banks of M_p:

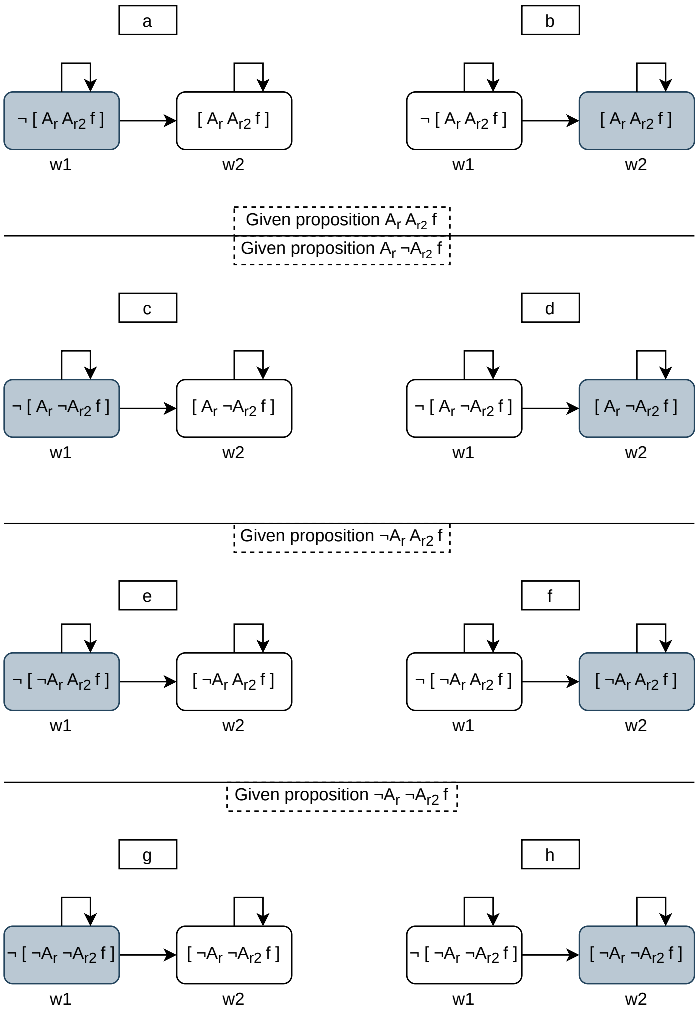

a
b
$\neg [A_r A_{r2} f]$
$[A_r A_{r2} f]$
$\neg [A_r A_{r2} f]$
$[A_r A_{r2} f]$
w1
w2
w1
w2
Given proposition $A_r A_{r2} f$
Given proposition $A_r \neg A_{r2} f$
c
d
$\neg [A_r \neg A_{r2} f]$
$[A_r \neg A_{r2} f]$
$\neg [A_r \neg A_{r2} f]$
$[A_r \neg A_{r2} f]$
w1
w2
w1
w2
Given proposition $\neg A_r A_{r2} f$
e
f
$\neg [\neg A_r A_{r2} f]$
$[\neg A_r A_{r2} f]$
$\neg [\neg A_r A_{r2} f]$
$[\neg A_r A_{r2} f]$
w1
w2
w1
w2
Given proposition $\neg A_r \neg A_{r2} f$
g
h
$\neg [\neg A_r \neg A_{r2} f]$
$[\neg A_r \neg A_{r2} f]$
$\neg [\neg A_r \neg A_{r2} f]$
$[\neg A_r \neg A_{r2} f]$
w1
w2
w1
w2

Figure 4.4: Possible world state-space for awareness classification 4

$$\begin{aligned} A4_p = \{&\neg A_p \neg A_{r_1} A_{r_2} f, A_p \neg A_{r_1} A_{r_2} f, \neg A_p \neg A_{r_1} A_{r_2} f, \\ &A_p A_{r_1} A_{r_2} f, \neg A_p \neg A_{r_1} \neg A_{r_2} f, \neg A_p A_{r_1} \neg A_{r_2} f, \\ &A_p A_{r_1} \neg A_{r_2} f, \neg A_p \neg A_{r_2} A_{r_1} f, A_p \neg A_{r_2} A_{r_1} f, \\ &\neg_p \neg A_{r_2} A_{r_1} f, A_p A_{r_2} A_{r_1} f, \neg A_p \neg A_{r_2} \neg A_{r_1} f, \\ &\neg A_p A_{r_2} \neg A_{r_1} f, A_p A_{r_2} \neg A_{r_1} f\} \; and \; \forall a \in A4_p, a \in M_p \end{aligned}$$

Similar rules regarding the current world w_c and the resulting accessibility relations are leveraged on to determine which (un)awareness instances are said to be tenable in the memory of an object.

As information disclosure regarding ψ occurs, the memory banks of the principal object will change, as the tenability of different awareness classifications will change. An (un)awareness instance is said to be untenable to an object in the event that the object has acquired sufficient information to rule out the possibility of the instance class, as a result of the shifting current world.

4.4 Memory Transforms

The complete memory state-space of an object will consist of all 42 possible (un)awareness instances from awareness classifications $A1, A2, A3$ and $A4$ as illustrated in table A.4. When interactions occur between an information subject, sender and recipient, the principle and reference objects will assume the roles of subject, sender and recipient as discussed in [90]

The complete instantiated memory banks of an information subject, sender and recipient are illustrated in Appendix A.4. We can see that there are three tables, each illustrating the state-space from the perspective of the subject (M_{su}), sender (M_s) and recipient (M_r). For example, labels 3-6 under M_{su} detail the (un)awareness instances of class $A2$ concerning the subject as the principle object p, and the sender as the reference object r'. Additionally, labels 7-10 detail the (un)awareness instances of class $A2$ concerning the subject as the principle p and the recipient as the reference object r'.

When the roles of subject and sender are not unique and thusly apply to the same object, this leads to the phenomena of positive or negative introspection, where an object considers whether it is (un)aware of its awareness (positive introspection) or unawareness (negative introspection) of a proposition. Such instances of knowledge reasoning are commonly discussed in epistemology and manifest in the awareness model in the event the roles of su and s are not unique.

As information disclosure regarding ψ occurs, the tenability of both the awareness and unawareness instances in the memory banks of each of these objects will change. An (un)awareness instance is said to be untenable for an object in the event that, as a result of information disclosure, it no longer makes logical sense for the object to consider that it is (un)aware of the proposition. For example, if a proposition ψ is disclosed to a recipient object r' then it is intuitive for r' to reason that it is aware of ψ.

As information disclosure occurs, the current world w_c will change which in turn determines which awareness and unawareness instances are considered tenable for that object. As previously discussed, an agent is fully unaware in the event that for all awareness classifications, the current world $w_c = w_1$ as this will result in all the (un)awareness instances being tenable for that object. Conversely, full awareness occurs when the current world $w_c = w_2$ for all awareness classifications.

The tenability of a subset of the instance classes in the memory of an object will result in partial awareness of disclosed information. Formally, we define partial awareness as an awareness differential as follows:

$$ADiff(p, A_i) = \frac{\Omega - 1}{\lambda - 1} + \delta$$

Where:

- Ω is the number of tenable instances of class A_i with the same reference object(s) in M_p
- λ is the number of instances of class A_i with the same reference objects(s) in M_p
- δ is the number of tenable unawareness instances of class A_i with the same reference object(s) in M_p

At full unawareness for a given instance classification A_i, the value of $ADiff = 1$. As the principal object gains awareness (ie. the number of tenable awareness instances increases and the number of untenable unawareness instances increases) the value of $ADiff(p, A_i)$ will decrease. At full unawareness wrt the instance classification A_i, the value of $Adiff = 0$. Measuring the awareness differential over all awareness class instances is thusly illustrated as follows:

$$\sum_{i=1}^{4} ADiff(p, A_i) = \frac{\Omega - 1}{\lambda - 1} + \delta$$

4.5 Conflict Formalism

The different awareness classifications discussed in 4.3 allows an analyst to unambiguously articulate the resulting knowledge acquisition of objects when reviewing the implementation of a software design. The nature of the designed software will impact on the manner by which privacy concerns evolve. For example, privacy consent mechanisms offered to data subjects are the default mechanism businesses leverage on to enact data subject control. When subjects interact with such consent mechanisms, the nature of their interactions will directly influence their privacy expectations. Awareness classifications 1-4 allow stakeholders to model the privacy expectations of a data subject, with reference to other objects in such interaction contexts.

Informally, we state that an assertion (α) represents a statement of awareness or unawareness a principal object p (where p can be the data subject or data processor) should have about a proposition f. In the context of this thesis, the proposition f relates to attributes associated with the data subject such as IP address or location data that can be tracked or otherwise measured by data processors online. We can map a proposition to one of these attributes. Therefore $f = su.loc$ represents the proposition mapping to the location data attribute of the data subject.

For a data subject, the tenability of (un)awareness instances from different awareness classifications change depending on the interactions they have with subject control mechanisms provided by data processors. These changes represent from the perspective of the data subject, the expected knowledge gain of data processors. An assertion in turn, is a statement of (un)awareness mapping to one of the tenable (un)awareness instances within the memory store of an object. For example, a data subject may wish to have Facebook be aware of their location and thusly opt into Facebook location tracking. This in turn will inform the creation of the following assertion and make the following tenable for the data subject:

$$A_{su}A_{Facebook}su.location$$

Where A is the modal operator of awareness. The expression is an awareness construct representing the awareness state from the perspective of a principal object p concerning the awareness of a reference object r. Here, $p = su$, $f = su.location$ and $r1 = Facebook$. This formally represents the data subjects expectation Facebook will acquire an understanding of the location attribute of the subject su.

Conversely, it may be the desire of a data subject to keep their location data secret from Facebook and in turn will opt out of location tracking. This will inform the creation of the following assertion and make the following unawareness instance tenable for the data subject:

$$A_{su}\neg A_{Facebook}su.location$$

Where the optional negation operator $\neg A$ indicates an unawareness construct. The assertion $A_{su}A_{Facebook}su.location$ and conversely, $A_{su}\neg A_{Facebook}su.location$ are examples of assertions from awareness classification 2 discussed in section 4.3. The attribute $su.location$ can be represented generally as the operator f. Where f maps directly to a single measurable attribute of a data subject. Equation 4.3 illustrates the general representation of an (un)awareness assertion:

$$(\neg)A_pf \tag{4.3}$$

Where the optional negation prefix indicates an unawareness operator (negating the awareness operator). The awareness classifications $A1, A2, A3$ and $A4$ discussed in section 3 allows us to represent higher order assertions that are more complex than the example in equation 4.3.

$$(\neg)A_p(\neg)A_{r1}f \tag{4.4}$$

$$(\neg)A_p(\neg)A_{r1}(\neg)A_pf \tag{4.5}$$

$$(\neg)A_p(\neg)A_{r1}(\neg)A_{r2}f \tag{4.6}$$

Formula 4.4, 4.5 and 4.6 all present a general representation of assertions that fall within classification $A2, A3$ and $A4$ respectively. These assertions are considered composite, specifically they reference multiple distinct principles (such that p, $r1$ and $r2$ delineate distinct objects). $A1$ is considered atomic, where it only references a single principle object p and does not reference any additional objects in the expression. Each assertion is modelled from the perspective of the principle p. Composite assertions can be simplified in their representation as $A_p\psi$, where ψ represents the additional (un)awareness constructs the principle is to reason about.

We can express more complex, higher order assertions by instantiating formulas 4.4, 4.5 and 4.6. For example, in the event a data subject is presented with subject control mechanisms on the first party data processor Whathifi and opts into location tracking by Facebook through this mechanism, the following assertions are generated:

(1) $A_{su}A_{Facebook}su.loc$,
(2) $A_{whathifi}A_{Facebook}su.loc$,
(3) $A_{whathifi}A_{su}A_{Facebook}su.loc$, and
(4) $A_{su}A_{whathifi}A_{Facebook}su.loc$.

Modeling privacy expectations as assertions enables stakeholders to compare multiple assertions to determine whether the resulting tenable (un)awareness constructs can exist within the memory stores of their respective principle subjects concurrently. This allows an analyst to make an assessment on whether the privacy expectations of data subjects will align or conflict according to usage contexts being analysed.

For example, in the event the interactions a data subject has with a consent management platform generate assertions that can all concurrently be tenable within the memory stores of their principle objects, there are no misaligned privacy expectations. However, conflict occurs in the event interactions with subject control mechanisms generate assertions that cannot concurrently be tenable in the memory stores of their respective principle objects. In such an event, the interactions a data subject has leads to a scenario where it is impossible for a system instrumentation to satisfy their expectations which in turn impacts their privacy concerns.

Consider the following example where we have a data subject interacting with a CMP presented by Whathifi. With this mechanism they deny consent to location tracking by Fandom, this informs the creation of the following assertion:

$$A_{su} \neg A_{Fandom} su.loc \tag{4.7}$$

This is an awareness construct that indicates the subject will have the expectation that the third party $Fandom$ will be unaware of the location attribute of the subject. However, in the event their interactions with the CMP on Techradar required the creation of the following assertion:

$$A_{su} A_{Fandom} su.loc \tag{4.8}$$

This will be problematic. These two assertions cannot both be tenable in the memory stores of the principle. Their mutual exclusivity means there is a conflict which serves to indicate a misalignment of user privacy expectations resulting from a given usage context. We will unpack these different usage contexts and their relation to conflicts in section 6.

Modelling the memory store M_p of a principle p with awareness classifications $A1, A2, A3$ and $A4$ allows us to pair awareness and unawareness constructs between awareness classifications to consider how the desired knowledge gain of a principle object may lead to incompatible memory states within M_p. In this work we articulate 3 distinct ways by which privacy conflict can occur. These are explicit, implicit and uncertainty conflicts resulting from the combination of unawareness and awareness constructs from different awareness classifications.

4.5.1 Explicit Conflict

Explicit conflict occurs in the event that a given pair of assertions (α, α') exhibit the following properties:

1. The principal object p should be identical.
2. The modal operator of p in α should be inverted in α'.
3. The proposition ψ must be identical in both α and α'.

Explicit conflict is a phenomena that involves the aggregation of incompatible assertions from the same awareness classification, thus $(\alpha, \alpha' \in A_n)$. The possible state-space of the memory of a principle M_p is illustrated in table A.4. Here, we illustrate each generalised possible (un)awareness instance that composes M_p for each awareness classification. For each awareness instance from a given classification A_i there exists a conflicting unawareness instance within the same classification.

For example, if a data subject opts into Admedia location tracking, a resulting assertion $\alpha = A_{Admedia} su.loc$ will conflict in the event they opt out of Admedia location tracking through another first party data processor resulting in $\alpha' = \neg A_{Admedia} su.loc$. The pair of assertions here satisfies the 3 conditions above as:

1. $p = Admedia$
2. $A_{Admedia}$ in α is inverted as $\neg A_{Admedia}$ in α'
3. $\psi = f = su.loc$ (the proposition is atomic, as it does not relate to a higher order consideration)

This aligns with label 1 in table A.1 which illustrates all possible explicit conflicts from M_p.

In awareness classification 1 ($A1$) the proposition $\psi = f$, as the principle p is only considering its understanding of a subjects attribute. We can see that row 1 in table A.1 is an example of explicit conflict occurring between a pair of assertions that both exist within $A1$. There are 2 possible conflict combinations from this classification involving only a principle and an atomic proposition f.

In awareness classification 2 ($A2$) the proposition becomes ψ as it models the (un)awareness state of a reference object. Rows 3-6 in table A.1 illustrate all possible instances of explicit conflict that can occur when introducing a reference object $r1$ in adherence to the 3 explicit conflict pre-requisites. There are a total of 4 conflicting assertion states from this classification. As an example, row 3 illustrates $\alpha = \{A_p A_{r1} f\}$ and $\alpha = \{\neg A_p A_{r1} f\}$. Here, we can see $\psi = A_{r1} f$, A_p is negated as $\neg A_p$ and the principle p is the same object reference.

In awareness classification 3 ($A3$) the proposition is another higher order consideration involving a reference object $r1$ and the principal object p rows 7 to 14 in table A.1 illustrate all possible instances of conflict from a pair of assertions satisfying the explicit pre-requisites. There are a total of 8 possible conflict scenarios from classification $A3$. As an example, row 9 illustrates $\alpha = A_p A_{r1} \neg A_p f$ and $\alpha' = \neg A_p A_{r1} \neg A_p f$ which are both reflexive awareness considerations. We can see again $\psi = A_{r1} \neg A_p f$, A_p is inverted as a negation between α, α' and p is the same object reference.

The final awareness classification 4 ($A4$) the proposition is another higher order consideration involving two distinct reference objects in sequence $r1$ and $r2$. Rows 15-22 in table

A.1 illustrate all possible conflicting pairs that satisfy the pre-requisites for explicit conflict. There are again 8 possible mutually exclusive combinations that can be made under classification $A4$. As a final example, row 19 illustrates the pair $\alpha = A_p \neg A_{r1} A_{r2} f$ and $\alpha' = \neg A_p \neg A_{r1} A_{r2} f$. We can see here that $\psi = \neg A_{r1} A_{r2} f$, the modal operator of p is inverted as a negation and p refers to the same object.

4.5.2 Implicit Conflict

Implicit conflict leverages off analysis of the constituents of a given assertion. In the event an assertion is composite, the proposition ψ will model the awareness of additional objects. It is necessary these constituents in turn are verifiable in the memory stores of their respective principle objects in order for a given assertion containing ψ to be tenable in the memory of M_p. For example, the assertion $\alpha = A_{Bob} A_{ifixit} A_{Facebook} Bob.IP$ is tenable in the memory of M_{Bob} in the event the constituents (which in turn can be treated as assertions) are tenable as follows:

(1) True $\leftarrow$ *tenable*$(A_{Facebook}$*Bob.IP*$, M_{Facebook})$
(2) True $\leftarrow$ *tenable*$(A_{ifixit} A_{Facebook} Bob.IP, M_{ifixit})$
(3) True $\leftarrow$ *tenable*$(A_{Bob} A_{ifixit} A_{Facebook} Bob.IP, M_{Bob})$

Each composite assertion therefore, will recursively yield a collection of constituent assertions C_α to be tenable in the memory of their respective principle objects. Implicit conflict occurs in the event one of the constituent assertions $\alpha_i \in C_\alpha$, conflicts explicitly with another assertion α', or one of the constituents within C'_α.

Taking the previous example $\alpha = A_{Bob} A_{ifixit} A_{Facebook} Bob.IP$ the constituents C_α will be as follows:

1. $\alpha_1 = A_{Facebook}$*Bob.IP*

2. $\alpha_2 = A_{ifixit} A_{Facebook} Bob.IP$

3. $\alpha_3 = A_{Bob} A_{ifixit} A_{Facebook} Bob.IP$

Assuming the assertion $\alpha' = \neg A_{ifixit} A_{Facebook}$*Bob.*$IP$ the conditions for explicit conflict are satisfied between α' and $\alpha_2 \in C_\alpha$. Consequently, α and α' conflict implicitly.

As a second example, consider $\alpha' = A_{ifixit} \neg A_{Facebook}$*Bob.*$IP$. Here we do not see conditions for explicit conflict between α' and an element of the constituent set of α. However, taking the constituents of α' where $\alpha'_1 \in C'_\alpha = \neg A_{Facebook}$*Bob.IP* we see there is explicit conflict observable between $\alpha_1 \in C_\alpha$ and $\alpha_1' \in C_\alpha'$. Again, this means there is a conflict between the constituents of α and α' hence, this assertion pair conflicts implicitly.

Table A.2 illustrates how implicit conflict occurs between a pairing of assertions α and α'. In this illustration, column 3 exhibits the complete set of constituents associated with the assertion presented in column 2. It is between the (un)awareness instances in columns 3 and 4 that we observe explicit conflict, which in turn means we observe implicit conflict between the assertions in columns 2 and 4.

For $A1$, it is not possible to observe implicit conflict between an assertion pair (α, a') such that $\alpha, a' \in A1$. This is because the propositions for assertions within $A1$ are atomic, and therefore we cannot elicit constituents from them.

Rows A2-1 to A2-4 illustrate the possible cases of implicit conflict in $A2$. A given assertion α from $A2$ will yield a a constituent collection C_α as illustrated below.

$$a \in A2 \rightarrow C_\alpha = \{(\neg)A_{r1}f\} \tag{4.9}$$

Taking A2-2 from table A.2 as an example, $\alpha = A_p \neg A_{r1} f$ will yield the constituent collection $C_\alpha = \neg A_{r1} f$. With $\alpha' = A_{r1} f$ we can see α' explicitly conflicts with the constituent element in C_α. Therefore, α and α' implicitly conflict.

Rows A3-1 to A3-8 illustrate the possible cases of implicit conflict for instances of $A3$. As the proposition ψ in classification 3 concerns 2 objects, the constituents can be elicited recursively, providing us with 2 member elements in C_α as follows:

$$a \in A3 \rightarrow C_\alpha = \{(\neg)A_{r1}(\neg)A_p f, (\neg)A_p f\} \tag{4.10}$$

Taking A3-7 as an example, here we have an assertion pair α, α' such that $\alpha = A_p A_r A_p f$. This awareness instance will provide us the following constituents $\alpha_1 = A_r A_p f, \alpha_2 = A_p f$. There are 2 possible instances of explicit conflict. In the event $\alpha' = \neg A_{r1} A_p f$ or $\neg A_p f$, either of these instances will explicitly conflict with one of the member constituents. Therefore, in both instances, α and α' implicitly conflict.

Additional consideration is paid to the constituent collection when we pair assertions where one stems from $A4$. Similar to $A3$, the constituent set we get from an assertion $\alpha \in A4$ is as follows:

$$a \in A4 \rightarrow C_\alpha = \{(\neg)A_{r1}(\neg)A_{r2} f, (\neg)A_{r2} f\} \tag{4.11}$$

The collection C_α contains 2 (un)awareness instances, one a member of $A1$ and another a member of $A2$. Given the requirement that constituent members $\alpha_i \in C_\alpha$ should be tenable in their respective M_p state-space, the constituent element $(\neg)A_{r1}(\neg)A_{r2}f$ allows for the principle to make transitive inference on the tenability of $(\neg)A_{r2}f$ in M_{r2}. This is unique to awareness class 4, as with class 3 the constituent set does not allow a principle to make such transitive inference on awareness tenability as it is a cyclic construct of a principles awareness state. Therefore we can expand on the previous constituent list to include:

(1) True $\leftarrow$ *tenable*$(A_{r2}f, M_{r2})$
(2) True $\leftarrow$ *tenable*$(A_{r1}A_{r2}f, M_{r1})$
(3) True $\leftarrow$ *tenable*$(A_pA_{r1}A_{r2}f, M_p)$
(4) True $\leftarrow$ *tenable*$(A_pA_{r2}f, M_p)$

Rows A4-1 to A4-8 illustrate the resulting combinations of assertions that lead to implicit conflict from $A4$, such that α yields a constituent set and α' explicitly conflicts with one of the constituent elements. To take A4-1 as an example. We have $A_pA_{r1}A_{r2}f$ which will yield $C_p = A_{r1}A_{r2}f, A_{r2}f$. In addition to this we have the inference element $A_pA_{r2}f$ which is added to the constituent collection. There are possible instances of α' which may explicitly conflict with one of the member elements of C_α.

Another dimension of implicit conflict analysis involves the analysis of the constituent elements from both assertions. In the event we have assertions α, α' where both assertions are not members of $A1$, both assertions will yield constituent collections C_α, C'_α respectively. In the event one constituent member $\alpha_i \in C_\alpha$ explicitly conflicts with another constituent member $\alpha' \in C'_\alpha$ then this is another manner by which α, α' conflict implicitly, as there exists mutual exclusivity between the constituent elements that cannot be concurrently tenable in the memory stores of their respective principle objects.

Table A.3 illustrates all possible instances of implicit conflict between two assertions where both assertions will yield constituent elements. Again, awareness classification $A1$ has been omitted from this table as it is not possible to acquire constituent elements from an atomic assertion.

Rows A2-1 to A2-4 illustrate the possible conflicting combinations of assertions from $A2$. In column 2 a listed assertion α along with its constituents C_α within column 3. Likewise, a proposed assertion α' and its constituents C'_α within columns 5 and 4 respectively. Taking A2-1 as an example, $\alpha = A_pA_{r1}f, a' = A_p\neg A_{r1}f$ implicitly conflict with one another because the constituent element $\alpha_1 = A_{r1}f$ explicitly conflicts with the second constituent element $\alpha'_1 = \neg A_{r1}f$.

Rows A3-1 to A3-8 illustrate conflicting combinations of assertions α, a' such that $\alpha \in A3$. The constituent set C_α in turn allows for greater possibilities when modelling conflicting assertion pairs. Taking A3-1 as an example, the provided assertion $\alpha = A_pA_{r1}A_pf$ will yield the constituent set $C_\alpha = A_{r1}A_pf, A_pf$. This in turn means potential conflicting constituents in C'_α are $\neg A_{r1}A_p$ or $\neg A_pf$. Such constituents may result from the paired assertion α' being $(\neg)A_p\neg A_{r1}A_p$ or $(\neg)A_p(\neg)A_{r1}\neg A_p$. The negation statements within parenthesis denote an optional negation which has no bearing on the conflict semantics.

Rows A4-1 to A4-8 illustrate conflicting combinations of assertions α, a' such that $\alpha \in A4$. Taking A4-1 as an example, the constituent set $C_\alpha = A_{r1}A_{r2}f, A_{r2}, A_pA_{r2}f$ resulting from $\alpha = A_pA_{r1}A_{r2}f$. This in turn means the following members from a conflicting constituent set $C'_\alpha = \neg A_{r1}A_{r2}f, \neg A_{r2}f, \neg A_pA_{r2}f$ will conflict explicitly. Therefore, any assertion facilitating the elicitation of any members from C'_α will implicitly conflict with α. Hence $\alpha' = (\neg)A_p\neg A_{r1}A_{r2}f$ or $(\neg)A_p(\neg)A_{r1}\neg A_{r2}f$ will implicitly conflict with α.

4.5.3 Uncertainty Conflict

Uncertainty conflict concerns the exclusive analysis of the memory state of a given principle M_p. A principle is considered to hold uncertainty, in the event a pair of assertions a, a' implicitly conflict with one another such that:

(1) True ← $\textit{tenable}(a, M_p)$
(2) True ← $\textit{tenable}(a', M_p)$

Uncertainty conflict concerns itself with higher order reasoning and as such the following semantic rules need to be observed:

1. The principal object p should be identical.
2. The modal operator of p should be identical.
3. The modal operator for the reference object in a should be inverted in a'

Uncertainty conflict occurs in the event a pair of mutually exclusive higher order assertions are both equally tenable in the memory of a principal. This occurs from a lack of information provided by the environment to allow the tenability of one of the conflicting assertions to be revoked. For example, Bob is said to be uncertain about whether $Yahoo$ is aware or unaware of their location data in the event the following assertions are tenable in their memory.

(1) True ← $\textit{tenable}(A_{\textit{Bob}}A_{\textit{Yahoo}}\textit{Bob}.loc, M_p)$
(2) True ← $\textit{tenable}(A_{\textit{Bob}}\neg A_{\textit{Yahoo}}\textit{Bob}.loc, M_p)$

For $A1$ uncertainty conflicts are not applicable given they are concerns resulting from higher order reasoning. Since $A1$ concerns with atomic assertions, a principle cannot reason about a reference object. The tenability concerns for $A2$ are shown below:

(1) True $\leftarrow$ *tenable*$(A_pA_{r1}f, M_p)$
(2) True $\leftarrow$ *tenable*$(A_p\neg A_{r1}f, M_p)$

Likewise, uncertainty conflicts can be applicable to $A3$ in the event the following assertions are tenable in M_P

(1) True $\leftarrow$ *tenable*$(A_pA_{r1}A_pf, M_p)$
(2) True $\leftarrow$ *tenable*$(A_p\neg A_{r1}A_pf, M_p)$

This indicates the principle p is uncertain of whether the reference object holds awareness or unawareness regarding the awareness state of the principal. The negation of the 3rd modal operator in the reflexive expression is omitted as it does not make sense a principal object would be uncertain of its own awareness state. Finally we can illustrate how uncertainty conflict can be facilitated with assertion instances from $A4$:

(1) True $\leftarrow$ *tenable*$(A_pA_{r1}A_{r2}f, M_p)$
(2) True $\leftarrow$ *tenable*$(A_p\neg A_{r1}A_{r2}f, M_p)$
or
(1) True $\leftarrow$ *tenable*$(A_pA_{r1}A_{r2}f, M_p)$
(2) True $\leftarrow$ *tenable*$(A_pA_{r1}\neg A_{r2}f, M_p)$

In the first combination, the principle is said to be uncertain of the awareness state of the first reference object. The second combination illustrates the principle is uncertain of the awareness state of the second reference object. Table A.5 and A.6 illustrates the possible combinations for uncertainty conflict for a given principle p. Again, with $A1$ we cannot arrive at uncertainty conflict as this phenomena refers to the modelling of a subjects uncertainty regarding the knowledge state of a reference object. In table A.5 rows A2-1 to A2-4 illustrate the paired assertions α, α' resulting in uncertainty conflict for $A2$. Taking A2-1 as an example, we can see $\alpha = A_pA_{r1}f$ whilst $\alpha' = A_p\neg A_{r1}f$, for these two assertions to remain tenable in M_p indicates the subjects uncertainty concerning the (un)awareness state of $r1$. Rows A3-8 indicate the same principle but for assertions of $A3$. Taking A3-1 as an example, the assertion $\alpha = A_pA_{r1}A_pf$ when tenable with $A_p\neg A_{r1}A_pf$ indicates the subject holds uncertainty regarding the (un)awareness state of $r1$ in a cyclic representation of the awareness state of p. Finally, rows A4-1 to A4-8 illustrate the combinations of assertions for $A4$ resulting in uncertainty conflict with su. Taking A4-1 as an example, $A_pA_{r1}A_{r2}f$ paired with $A_p\neg A_{r1}A_{r2}f$ indicates the subject holds uncertainty regarding the awareness state of $r1$.

Modelling privacy from the perspective of awareness and unawareness enables us to align our work with philosophical views on privacy being a boundary management process on what information users are to be aware, and unaware of. Modelling conflicts with awareness enables us to identify granule, contradicting views on how users wish information be disclosed while maintaining a separation of concerns from domain specific properties. The rules for explicit, implicit and uncertainty conflicts enable us to identify conflicting privacy requirements by virtue of incompatible statements of awareness between software objects.

4.6 Summary

In this chapter we introduced our software awareness model, which is a computational model of knowledge we use in this work to represent knowledge states of objects interacting on a network. We state the notion of privacy in software systems relates to the management of knowledge gain in an interaction network. We leverage on this definition as motivation for the discussion of an awareness model to articulate information gain on a network as a metric to be used for privacy analysis.

We continue the discussion of our awareness model with an introduction to some of the formal notations and semantics commonly used in the construction of knowledge systems in computing science. We discuss the formalisms of Epistemic Modal Logic which are commonly used to represent held knowledge agents have about the knowledge states of other agents. We leverage on such semantics in our work as such approaches align well with our problem scenario.

We introduce possible world semantics in the construction of our awareness model to enable software agents to reason about their local understanding of information. Possible worlds are utilised in conjunction with accessibility relations to determine which epistemic possibilities are considered tenable to an object.

We introduce four different awareness classification formalisms concerning the reasoning of different tiers of higher order reasoning. Together, these formalisms allow us to represent the knowledge state space of interacting objects on a network. We then discuss the manner by which the tenability of awareness and unawareness statements change as interactions occur. We describe the concept of tenability and how the possible world semantics are leveraged on to determine the tenability of both awareness and unawareness instances in the state space of an object.

Chapter 5

Case Study: Web Interactions

5.1 Introduction

When interacting with online services, it is often the responsibility of end users to make appropriate disclosure decisions to reduce the perceived risk to their privacy concerns. For instance, a first party data processor may only provide notice to the subject data processing activity occurs involving third party data processors. A first party data processor collects or generates data about the data subject directly, whilst a third party data processor processes data sourced from first parties or other third parties. In such instances, privacy policies are leveraged to provide full transparency (and therefore shift responsibility of managing privacy concerns) to the data subject under the assumption they have read, understood and agree to the terms laid out in the policy.

Studies have concluded the opaque language of privacy policies renders them ineffective. Furthermore, the sheer number of policies a data subject would be expected to read in the event a first party connected to several third parties renders the task of understanding the consequence of their interactions infeasible. CMPs are often employed to reduce the burden placed on data subjects through technical mechanisms for data subjects to review and determine whether they agree to specific legal basis for data processing. For instance, data processing activity can be enacted under contractual necessity, such that for the provision of basic services, data processing of subject attributes must occur. Other lawful basis involves the provision of informed consent, where users are provided options regarding which third parties they wish to provide their consent to for data processing to occur.

The challenge with CMPs is the capacity for data subjects to perceive or enact control over the data processing activities is largely constrained by the nature of their implementation. For instance, if a data subject rejects consent to data processing by a third party when visiting a first party, such control can be enacted by the data subject because data processing occurs under the legal basis of consent. However, if they were to subsequently interact with a first party where the legal basis is contractual necessity, their expectations will be undermined in the event data processing occurs involving the same third party they previously denied consent towards. The combination of optional and mandatory data processing activity results in misaligned expectations, increasing risk of privacy harm. The fact data processors may provide full, partial or no transparency to data subjects, means data subjects when interacting with first parties, may enact a series of data processing activities which do not align with their privacy expectations.

Several contemporary privacy regulations (such as the European GDPR) compel stakeholders to demonstrate adherence to the paradigm of Privacy by Design where the mitigation of privacy risk is considered in the design of software systems. Therefore, to demonstrate compliance it is important for analysts to understand the nature of these data processing activities. To this end, in this section we aim to address RQ2 by providing analysts with a methodology to elicit information flow network representations corresponding to observable data processing activities between first and third parties online. We articulate three categories of data processing activity (mandatory, optional and covert) and unpack a case study applying the methodology to demonstrate the approach in assisting stakeholders model the data processing activities between first and third party data processors.

We begin this chapter with a discussion of the workflow we follow in section 5.2 before illustrating our three step process. In section 5.2.2 we communicate the elicitation method and introduce the key workflow steps involving the Selenium toolkit used in the study. Section 5.2.3 illustrates the process followed to validate the network that we elicit from the Selenium workflow. The output from sections 5.2.2, 5.2.3 are used to identify data flow categories detailed in section 5.2.4. The results of the study are presented in section 5.2.4 and we conclude with an outlook on the application of the results in section 5.3.

5.2 Method Overview

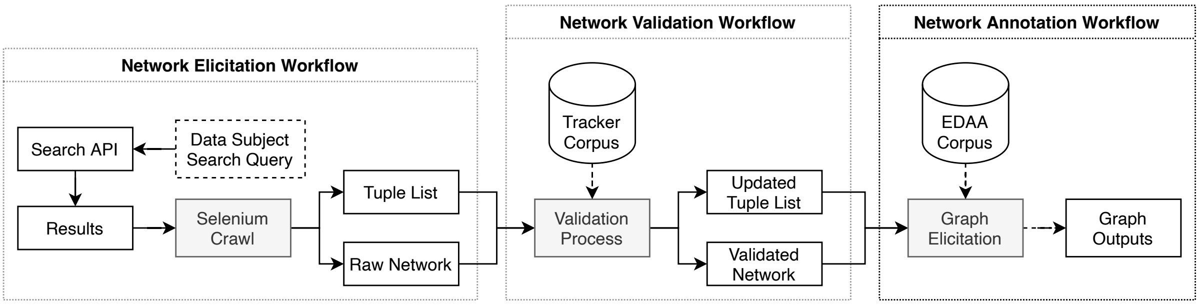


Figure 5.1: An overview of the methodology followed for network elicitation and data flow categorisation.

The methodology articulated in figure 5.1 allows an analyst to conduct analysis of the data processing activities between a series of first parties and their associated third parties. The approach can be articulated as a three step process where the first step concerns the elicitation of network data. We position the methodology as a workflow to perform privacy risk assessment during Data Protection Impact Procedure (DPIA). In step 1 we leverage on a constructed query to elicit a series of first party URLs returned from the Custom Search API. This step can be enacted by analysts, or can be substituted with URL information they possess.

To distinguish between different data processing categories, it is first necessary to measure ground truth of data processing activity. This involves capturing requests made to external third party domains when data subjects interact with first parties. However, it is often the case such requests are executed in scripts by the browser, rendering traditional scraping workflows ineffective at acquiring representative data. We mitigate this challenge in the elicitation workflow by leveraging on Selenium. Selenium is an autonomous web testing framework providing the web driver toolkit. This toolkit allows developers to configure and manage autonomous browser instances for the purpose of emulating user interactions. The distinct advantage of Selenium is its capacity to execute JavaScript and thusly, render dynamic HTML elements imperceivable by other scraping tools. With Selenium, we can point the autonomous browser to a target URL and acquire a dataset on the external resource requests made to third parties. The output of this step N will be a k, v JSON structure representing the connections between first and third parties (where first parties are mapped to k and third parties are mapped to a collection v) observable in the crawl. This is labelled as 'Raw Network' in figure 5.1. In addition, the output 'Tuple list' will be a collection T such that $t_i = f, o, v \in T$ contains information pertaining to the observable behaviour of a first party. The first party is labelled f, o indicates the data processing behaviour observed on f whilst v is a collection of third parties acquired through consent elicitation modules provided to the data subject. Both datasets are input into step 2 which concerns the validation and abstraction of the elicited network.

Step 2 involves a 2-step validation and abstraction process. The objective of this workflow is to reduce noise on the elicited dataset by removing third parties performing benign functions and are not involved in tracking data subjects online. We leverage on an external dataset of known third party trackers and perform a cross examination of the third parties elicited from step 1 of the methodology. For consistency, we perform the same check on the 'Tuple List' data. The output of this approach will be two datasets. The 'Validated Network' dataset N' will be an updated JSON k, v structure where the third party elements within v are verifiable third parties from the Selenium crawl. The 'Updated Tuple List' collection T' will be updated such that $v \mapsto t_i \in T$ will be updated in the same manner. These two datasets will be input into the final step to elicit information flow graph models.

Step 3 involves the creation of a network information flow graph illustrating the distinct interaction behaviours of first parties. This involves associating annotated behaviour of a first party in T' with the validated ground truth interactions in N' to generate network edges. For instance, a first party may connect to multiple third parties but the data processing behaviour involving these third parties may be distinct. We leverage on the observable behaviour of a first party in T', along with the connections observed in 'Validated Network' to delineate whether the first party enacts mandatory, optional or covert behaviour when connecting to a specific third party. Thusly, we parameterise generated network edges with this observed behavior in the creation of an information flow graph. Furthermore, we can also paramaterise the behaviour of third parties when creating network edges and we leverage off both the Selenium workflow, along with the 'EDAA corpus' to distinguish between mandatory, optional and covert behaviours of a third party. The output from this process can be used by stakeholders to perform analysis on usage contexts introducing privacy risk to data subjects.

5.2.1 Data Elicitation

The commodification of user data that supports targeted advertising models occurs frequently within web settings. The privacy risk incurred by data subjects when interacting with multiple data controllers in a given online web session can arise due to the nature of instrumented CMPs as discussed in section 1. This means the expected privacy loss as understood by the data subject may not accurately reflect the actual privacy loss occurring owed to the data processing activities between processors. The workflow to model data processing activity requires an input dataset, specifically a collection of data controller URLs. The focus of the case study was to model a resulting data processing network which is representative of data subject interactions with data controllers in a typical online session. The principle is to demonstrate inherit privacy risk data subjects can be exposed to when interacting with web services in a typical online sitting.

To acquire an input dataset representative of a typical web browsing session, we elicited the input dataset through the construction of a custom search parameter to be provided to the Google Custom Search API. The given parameter will control the returned data controllers to be input into the solution framework, but does not impact the methodology followed to analyse privacy risk on the data controllers within the input dataset. The objective of smart advertising models is to primarily assist advertising partners maximise their return on advertising expenditure by serving relevant ads to data subjects on products and services [91]. To this end, we opted to focus the search on consumer headphones. Consumer reports suggest headphone purchases from users are often infrequent and follow substantial investigation to inform purchasing decisions. Given the high likelihood for users to elicit information about headphones in an online context, it is intuitive to consider data subjects will likely interact with several data controllers for the purpose of investigating headphone products. Secondly the criteria should identify a popular brand associated with the chosen product. We focused on the Beats brand, being one of the most recognisable headphone brands with higher average Google trend statistics than its competitors.

The constructed query $q =$ ”beats headphone reviews” will yield a dataset of controllers from the API illustrating a typical search result presented to data subjects [1] when they opt to read reviews on a potential future purchase. Data controllers related to e-commerce are often cited as having higher levels of third party tracking in privacy literature [92]. Therefore this case study investigates the degree of privacy risk a data subject will be exposed to in the event they look to exhibit conservative behaviours on data processing controls offered on CMPs they will interact with online under the context of reading product reviews online.

5.2.2 Network Elicitation

For the elicitation we implemented a Python Selenium instance of Google Chrome configured with the default security and privacy settings. In the study we do not assume that the data subject will be utilising external script blocking technology to circumvent data processing activity. To identify external third parties we configured the Chrome browser with a single extension Lightbeam. Lightbeam is an external research project [93] that aims to provide end users transparency on which third parties are associated with a first party online. User browsing activity is often tracked through the instrumentation of technologies such as cookies that are requested by and fulfilled in the browser when executing JavaScript from the first party. The role of LightBeam in the Selenium crawl is to identify these requests when the browser visits a first party. We configured Lightbeam as an autonomous external .crx addon to the Selenium browser which allows the addon to operate with the autonomous Selenium workflow.

The Selenium workflow is enacted on the collection of first party URLs acquired from the Custom Search API. First we create a Chrome browser instance and load the external Lightbeam .crx addon. Next, we iterate through the URLs in sequence and create a new request to the URL through the Selenium browser. Each request made occurs in the same session instance. When visiting a first party with the Selenium browser, the first party will enact one of the following behaviours:

1. Provide only notice of data processing activity to the data subject.
2. Provide notice and consent options via the implementation of consent elicitation modules.

[1] we assume that a data subject is not signed into any affiliated accounts with the search engine and the results represent a ’fresh’ search on the Google search engine.

3. Does not provide any notice or consent options to the data subject.

The objective of the Selenium implementation in this step is to distinguish between these behaviours for the purpose of labelling the observed behaviour on a first party. To achieve this we inspect the rendered HTML acquired from the first party when requests are fulfilled in the browser. Consent management solutions instrumented in house or by external entities such as OneTrust can be identified within the HTML of first parties through prescribed attributes associated with the tag elements of the page source.

The Selenium browser will simulate data subject interactions with the first party. The purpose of which, is to firstly identify whether the first party provides notice of data processing to the data subject. Secondly, we wish to identify named third parties as presented to the data subject, in the event the data subject is provided subject control mechanisms to express their data processing preferences. To this end we implement a series of rules R such that each collection $r_i \in R$ contains an ordered series of XPath elements for the browser to check for. Each collection is representative of the series of steps a data subject should enact in order to identify the listed vendors on the consent elicitation modules. The collection R allows Selenium to distinguish between different consent elicitation technologies, and delineate the different levels of transparency exhibited towards the data subject.

For instance, if a first party f provides subject control mechanisms in the form of a privacy consent framework, this can be identified by examining the HTML source to identify elements matching rules in R. These include links to submit data processing preferences to third parties. Conversely, if only notices are provided to data subjects, again, the series of identifiable HTML elements can be associated with a ruleset in R. Finally, if there is nothing provided to end users, then no rules within R can be matched. We implement Selenium to autonomously determine whether such elements of interest exist on a first party, to determine how data processing occurs on that first party. The objective is to generate a tuple $t = (f, o, v)$ for each first party such that:

- f represents the first party that is being visited.
- o represents the nature of the interaction f has with the data subject.
- v represents the list of vendors that are identifiable from the consent elicitation module.

The procedure taken to elicit t is illustrated in algorithm 1. The browser bins rules that can be matched against the observable XPath elements within the HTML of the first party. At line 3 the $r_i[index]$ check will return true in the event the XPath element contained at the index offset can be identified within the HTML. For example, the OneTrust consent elicitation

Algorithm 1: Recursive approach to elicit the tuple (f, o, v).

```
$Acquire\ index$
Input  : index value
Output: Tuple labelling the first party $(f, o, v)$
for $r_i \in R$ do
    if $true \longleftarrow r_i[index]$ then
        $\theta \cup r_i$
if $\theta$ is Empty then
    return $(f, Covert, \{\})$
else
    if true $\leftarrow same(\theta[index])$ & $true \leftarrow scrape(\theta[index])$ then
        $o \leftarrow vendorScrape()$
        return $(f, Optional, o)$
    if true $\leftarrow same(\theta[index])$ & $true \leftarrow exit(\theta[index])$ then
        return $(f, Mandatory, \{\})$
    else
        execute($\theta[index]$)
        return Acquire(index+1)

```

module provides a link to allow data subjects to view third parties for the purpose of opting-out. Here, the XPath element associated with the consent elicitation module link is one of the first possible elements a data subject can interact with if data processing activity operates under the lawful basis of informed consent. If the element within the ruleset r_i at the offset is contained within the source, it is added to θ.

There are three possible outcomes. In the event no XPath elements contained in R can be identified in the HTML, no ruleset can be binned and θ will be an empty set. In this instance, we return the tuple representation $t = (f, Covert, \{\})$. Here, f points to the first party, $o = Covert$ and $v = \{\}$. Here, the analysis concludes the first party does not operate data processing transparency as indicated by o. Finally v is returned as an empty set.

If θ is not empty, we have matched some of the rules. If we do not have a consensus on what XPath elements have been matched (i.e. the matched XPath element is not unanimous across the rules), we take a majority to determine which element to work with. We use sentinel values within the rulebases to determine when Selenium should look to scrape the vendor lists provided to data subjects. If the selected element is a sentinel value, we take the result of the scrape function. This function acquires a list of third party data processors a data subject can indicate their data processing preferences to. The returned tuple in this outcome is $t = (f, Optional, o)$ such that f is the first party, o indicates the first party provides consent modules to the data subject and v is a non empty set of vendors identified in the vendorScrape function. If the element is not the sentinel value, we perform an interaction on the Selenium browser via the execute function. This allows Selenium to emulate user interaction with the first party and will update the index value before calling the procedure again. In the final else clause, we increment the value of index and consider the next available XPath command in the rulesets in a recursive analysis.

If a ruleset has been exhausted as indicated in the exit function on line 11 we return the tuple $t = (f, Mandatory, \{\})$ where f points to the first party, o indicates data processing transparency is provided to the data subject through notices, but full control is not provided. v is an empty set as no vendor information is identifiable.

The Selenium crawl will generate a tuple t_i per process 1 for each first party returned from the Custom Search API query. This Selenium workflow will output a collection $T = \{t_1 ... t_n\}$ labelled 'Tuple List' in figure 5.1 labelling the behaviours of first parties with the data subject and provide a collection of optional vendors if such data is provided to a subject. The Selenium workflow is programmable such that the browser instance can end the simulated interaction in one of two ways:

1. The browser can accept all data processing occurring when interacting with f
2. The browser can reject all data processing occurring when interacting with f

When users reject or accept all data processing the resulting information flow networks are expected to differ. This workflow enables analysts to acquire an understanding of the different data processing behaviours between a first party and third parties under the two different interaction contexts noted above. When visiting URLs from the Custom Search query, Lightbeam will monitor the outbound requests to an external third party and will associate the identifiable third parties as a JSON k, v data structure such that k points to the first party and v is a collection of all the associated third parties. All requests made in the Selenium workflow occur on the same browser instance, therefore Lighbeam cumulatively collects data on third parties involved in the processing of user data when visiting first parties online. When the Selenium workflow finishes interacting with a first party by accepting or rejecting all data processing, the resulting interactions involving third parties can be monitored by Lightbeam. This data will serve as ground truth interactions between first and third parties when data subjects interact with first parties online. For the study, we configured Selenium to terminate interactions with a first party by accepting data processing activity if the option was available.

5.2.3 Network Validation/Abstraction

The challenge with Lightbeam concerns the requirement to distinguish third parties involved in the tracking of data subjects from those performing benign functions. Lightbeam does not distinguish third parties natively, which introduces noise into the output dataset from the elicitation process. It is necessary to validate third party data processors as trackers prior to performing conflict analysis on the network. To achieve this we leverage on a public corpus labelled in figure 5.1 as ’Tracker Corpus’. The dataset provided by [94] is a publicly available record of a large scale study involving the analysis of 3.5 billion web pages as part of the 2012 common crawl project. The aim of the study was to identify third-party resources in HTML pages and the associated resource provides a dataset including 1375 labelled third parties with the following (truncated) schema:

domain registration_org ... company

Where ’domain’ is the second level domain, this column in the dataset provides us with a URL to navigate to the third party directly. ’Registration_org’ is the named corporate entity responsible the domain entry. Finally, ’company’ is a second column for corporate identification in the event the ’Registration_org’ field is empty. This dataset allows analysts to cross reference the ’Raw Network’ output from Lightbeam to determine which third party data processors from the observations can be verified as trackers. The procedure to validate third party trackers is articulated in algorithm 2:

Algorithm 2: Approach to validate a third party from the Lightbeam data structure.

```
Validate $D$
Input  : JSON k,v collection $D$
Output: JSON k,v collection $D'$
for $k, v \in D$ do
    if $true \longleftarrow firstParty(k)$ then
        let $k' = k$
        let $v' = \{\}$
        for $tp \in v$ do
            if true $\leftarrow success(tp)$ then
                let $reg \leftarrow whois(tp)$
                if $true \leftarrow contained(reg, schema, registrant)$ then
                    $v \cup id$
            else
                let $URL \leftarrow whois(tp)$
                if $true \leftarrow contained(URL, schema, domain)$ then
                    $v \cup id$
            $D' \cup (k', v')$
return $D'$
```

The data acquired from Lightbeam is a structured JSON object. Each first party within the key value pair correspond to a collection of third parties. In our analysis, we first identify the k, v entries within the JSON object associated with first parties. The function firstParty returns true in the event k contains a corresponding flag element indicating its first party status. For each first party, we check the third party elements, to see whether the third party exists within one of the three column sections illustrated in the schema. If we identify the third party, we label it as a verified tracker and add it to an updated value list, which will replace the original unverified third party list. The objective is to replace the original third party collection with an updated collection where the elements can be verified by cross referencing the public corpus of known data trackers. However, there are two challenges to this approach:

1. Multiple unique tracker URLs may point to a common second level domain (SLD).
2. Third party trackers may leverage on aliases with unique URLs.

These challenges introduce ambiguity into the analysis, therefore it is necessary to abstract the resulting data-set from Lightbeam to verify third parties. To perform the abstraction, we leverage on an external whois lookup API to identify registrant information associated with the URLs of each labelled third party URL from the Lightbeam dataset. The output of the whois lookup will be used to cross-examine the dataset schema provided by [94]. There are two possible outcomes:

1. The lookup is successful (on line 7) and identifies registrant information associated with a tracker URL. We take the registrant information and perform the cross-examination with the 'registrant' entry in the schema provided by [94]. The whois function on line 8 will return the registrant information reg and the contained function per line 9 will return true in the event reg can be identified in the supplied schema attribute $registrant$ in the associated database $Schema$.
2. The lookup on line 7 is not successful but provides us with the URL of the third party. In this instance, we take the URL and perform the cross examination per line 12. The function contained on line 13 returns true in the event URL can be observed in the $domain$ schema attribute of the external database $Schema$.

The output of the procedure will be a second JSON structure, where each key value pair corresponds to a first party, third party collection relation. Analysts can work with either the full registrant information associated with a tracker URL or the URL itself if the registrant information is not available to validate third parties and in turn reduce the noise from the Lightbeam output.

This approach addresses the challenges for the following reasons:

1. Multiple tracker URLs on the same second level domain will be aggregated in the analysis because registrant information can associate a single registrant to the different URLs. If registrant data is not usable, raw URLs can be cross referenced with the registrant name provided in the external dataset.

2. CDNs with different URLs under the same named registrant can be aggregated. Again in the event registrant lookup fails, the URLs (if matched) can be used to lookup registrant information from the external database

The process followed to update the vendor lists for each tuple $t_i \in T$ is illustrated in procedure 3:

Algorithm 3: Approach to validate third parties from the tuple collection T.

```
Input  : Tuple collection $T$
Output: Tuple collection $T'$
let $T' = \emptyset$
for $t_i \in T$ do
    let $v' = \emptyset$
    for $tp \in v$ do
        if $true \leftarrow contained(tp, schema, company)$ then
            $v' \cup tp$
        else
            if true $\leftarrow success(tp)$ then
                let $reg \leftarrow whois(tp)$
                if $true \leftarrow contained(reg, schema, registrant)$ then
                    output
    $(f, o, v') \cup T'$
return $T'$
```

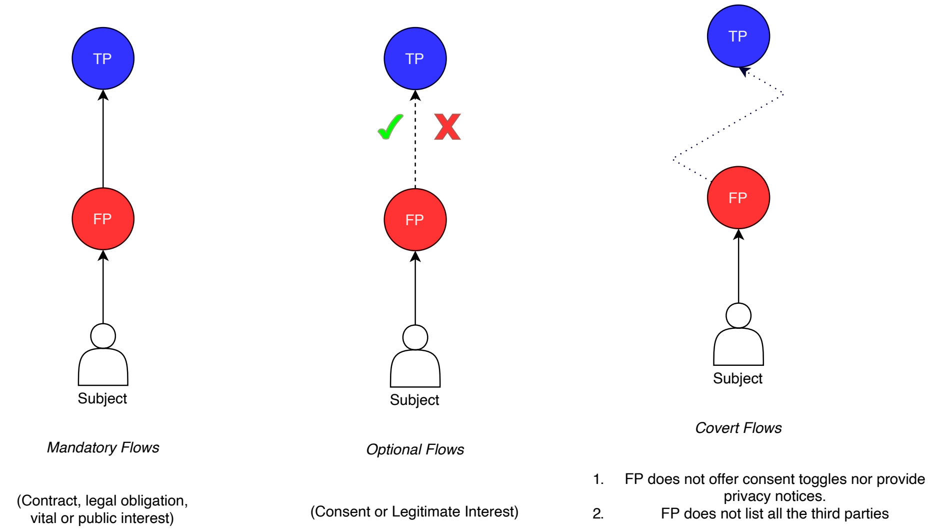
For each tuple $t_i \in T$ we iterate through the vendor elements within v. Each vendor presented to the data subject is more human readable compared to the output acquired by Lightbeam. In response to this we attempt to match the vendor name with the company schema attribute in the external database. If successful, we verify tracker status and append the value to an updated collection v'. If we are not able to match the schema, we look to acquire the result of the whois lookup, as the API provides us with the capacity to match partial information if provided. If successful, we take the registrant information and match that data in the registrant schema. If successful the vendor is added to v'. The resulting tuple (f, o, v') is added to T'. The values f, o do not change from previous. This continues for each tuple $t_i \in T$.

5.2.4 Network Annotation

The final step involves the generation and annotation of a directed graph G such that each edge $e \in G$ represents data processing activity between first and third parties modelled as vertices. G is constructed from the N' data structure labelled as 'Validated Network' on figure 5.1. Each key in the JSON structure connects to a series of third parties and thusly can be represented as an edge $E(V_1, V_2)$ where $V_1 = k$ (the first party) and $V_2 = tp_i \in v$ (a third party member element of v).

When creating an edge $E(V_1, V_2)$ the objective is to parameterise V_1 and V_2 with elicited data processing behaviour. For instance, in the event a data subject were to visit a first party (V_1) they may not be provided consent elicitation modules allowing the data subject to opt out of data processing involving a third party V_2. However, if the data subject were to visit the third party, such opt out mechanisms may be provided. The data processing behaviours of V_1 and V_2 are not identical, but influence the manner by which user information is disseminated across an information flow network.

Parameterising V_1 and V_2 requires understanding their data processing behaviours. Paramaterising V_1 involves cross referencing the observed behaviour as noted on the respective tuple structure $t_i = \{f, o, v\} \in T$ such that $f \mapsto V_1$. There were three possible observations from the Selenium crawl characterising the data processing behaviour of a first party as illustrated in figure 5.2.


TP
FP
Subject
Mandatory Flows
(Contract, legal obligation, vital or public interest)
TP
✓
X
FP
Subject
Optional Flows
(Consent or Legitimate Interest)
TP
FP
Subject
Covert Flows
1. FP does not offer consent toggles nor provide privacy notices.
2. FP does not list all the third parties

(a) Mandatory flow illustration

(b) Optional flow illustration

(c) Covert flow illustration

Figure 5.2: An illustration of mandatory, optional and covert information flows involving a subject, first party (FP) and third party (TP)

Where FP is the first party, TP is the third party and $subject$ is the data subject.

Mandatory Flows

Mandatory flows occur in the event FP provides notice to su informing them of data processing of their attributes but do not provide them with any technical means by which they can express or otherwise object to the data processing activity. Alternatively, where privacy consent toggles are provided to su but no specific opt out option is provided for a specific TP, the information flow is considered mandatory as su cannot exert any control over the dissemination of their attributes. We assume mandatory flows occur under the legal requirement of contractual necessity for the provision of services. We parameterise V_1 in an edge $e = (V_1, V_2)$ having mandatory behaviour in the event the following holds:

$$t[o] = Mandatory$$

Here, t indicates Selenium workflow detected notice of data processing towards the data subject but did not identify mechanisms to opt out of data processing. Therefore, interactions between V_1 and third parties are considered mandatory, operating under the legal basis of contractual necessity.

Parameterising V_2 involves annotating third party data processor behaviour. We leverage on Selenium to determine whether subject control mechanisms are available to the data subject. A third party exhibits mandatory behaviour in the event data subjects are provided notice regarding third party behavioural tracking online, but are not provided a means by which they can opt out or otherwise reject such data processing activity.

Optional Flows

Optional flows occur when FP provides a consent toggle to su listing TP. If su can express preference on the data processing activities involving TP then FP exhibits optional data processing behaviour under the lawful basis of consent or legitimate interest, where su has the right to choose whether or not such data processing occurs. Formally, we parameterise V_1 having optional behaviour in the event the following holds:

$$t[o] = Optional \text{ and } V_2 \in v$$

Here, t indicates Selenium identified mechanisms allowing data subjects to exert control over data processing. If V_2 is contained in the optional vendors list provided by V_1 then V_1 is parameterised as exhibiting optional behaviour.

To parameterise V_2 we leverage on the EDAA corpus as illustrated in figure 5.1. The European Digital Advertising Alliance (EDAA) is a regulatory initiative to promote consumer trust in online advertisement delivery models. The aim of the EDAA consent management platform is to provide a single data subject control mechanism for data subjects for the expression of consent preferences. Third parties affiliated with EDAA are audited and expected to respect the consent options expressed by data subjects. A key distinction to make is the EDAA allows data subjects to present their consent preferences to a third party without the involvement of a first party web service. Interactions therefore occur between the subject and the third party directly. A third party being listed as an accredited associate of the EDAA framework allows us to verify the third party will allow data subjects to express their consent options to data processing directly. If $V2$ is contained within the EDAA corpus we label the behaviour of $V2$ as optional.

Covert Flows

Covert flows occur in the event FP does not provide data processing notifications to su. Consequently, su attributes are collected and processed between first and third parties without the knowledge of the data subject. Furthermore, covert flows occur in the event that the privacy consent toggles offered by first parties are incomplete. An incomplete consent module does not make all third party processors involved in data processing known to the data subject. In both instances, we consider the information flow to be covert in nature. We parameterise V_1 having covert behaviour in the event one of the following holds:

1. $t[o] = None$
2. $t[o] = Optional$ and $V_2 \notin v$

If condition 1 holds, t indicates Selenium was unable to identify any HTML elements indicating the provision of data processing notices or data subject control mechanisms. This means the connections a first party has to third party data processors are covert in nature as the data subject is not made explicitly aware of data processing activity.

If condition 2 holds, t indicates Selenium detected data processing notice and consent elicitation modules. Therefore, the collection of vendors contained in $t[v]$ where $t \in T'$ is cross referenced to consider whether the target third party V_2 is contained within v. If it is not, the interactions between the first and third parties occur irrespective of user preference and are labelled covert, as the data subject is not made aware of these interactions. Parameterising V_2 involves asserting there are no detectable consent mechanisms provided to data subjects by the third party directly.

Procedure

The procedure followed to construct an information flow graph G is illustrated in algorithm 4. This process creates a series of network edges with vertices paramaterised in correspondence with the information dissemination behaviour of first parties. Modelling the connections under different data processing conditions allows analysts to consider what third parties are recipients of data under which legal basis for data processing. This allows for the identification of data processing activity where there is no legal basis for data processing, or the inter-dependant nature of data processing can result in misaligned privacy expectations from end users.

Algorithm 4: Approach to elicit a network edge.

```
1  **for** $k, v, t_i \in T$ **do**
2      **for** $tp_i \in v$ **do**
3          **if** *true* $\leftarrow active(tp_i)$ **then**
4              $param = behaviour(tp_i)$
5              $tp_i \mapsto V_2[param]$
6          **else**
7              $tp_i \mapsto V_2[Inert]$
8          **if** $Covert \longleftarrow t_i[o]$ *or (Optional* $\longleftarrow t_i[o]$ *and* $tp_i \notin T(v)$*)* **then**
9              $k \mapsto V_1[covert]$
10         **if** $Mandatory \longleftarrow t_i[o]$ **then**
11             $k \mapsto V_1[mandatory]$
12         **if** *Optional* $\longleftarrow t_i[o]$ *and* $V_2 \in T(v)$ **then**
13             $k \mapsto V_1[Optional]$
14     $E(V_1, V_2)$
15     $G \cup E$
16 **return** $G$
```

We iterate over each first party k, the associated third parties v and corresponding tuple element $t_i \in T$. For each third party $tp_i \in t_i$ we check whether the third party is active per line 3. Active third parties are reachable by Selenium and can thusly interact with data subjects. If active, a third party may exhibit mandatory, optional or covert data processing behaviours. The function 'behaviour' returns 'param' mapping to either mandatory, optional or covert labels which in turn paramaterise V_2. If the third party is inert, we parmaterise V_2 per line 6. Inert third parties we assume default to the associated privacy dissemination behaviours of associated first party data processors.

With V_2 generated, we generate and parameterise V_1. We check whether the associated tuple structure $t_i[o]$ is optional. If so, we check whether the third party is not a member element of the optional vendors ($T(v)$). In such an instance, we parameterise V_1 as being covert per line 10. Otherwise, we check if $t_i[o]$ is mandatory. If so, V_1 is parameterised as mandatory. Finally, if $t_i[o]$ indicates the first party exhibited optional behaviour and the third party is contained in the optional vendors list for the first party, V_1 is paramaterised as optional per line 13. With both V_1 and V_2 instantiated we create the edge $E(V_1, V_2)$ and add it to G.

Taking 51 first parties from the API search, our initial step in the network elicitation process returned a network of 4205 third parties. Applying the validation procedure in turn reduced this to 139 data processors. Of these entities, 37% were first parties and 63% were third parties. The network elicitation result is shown in figure 5.3 a).

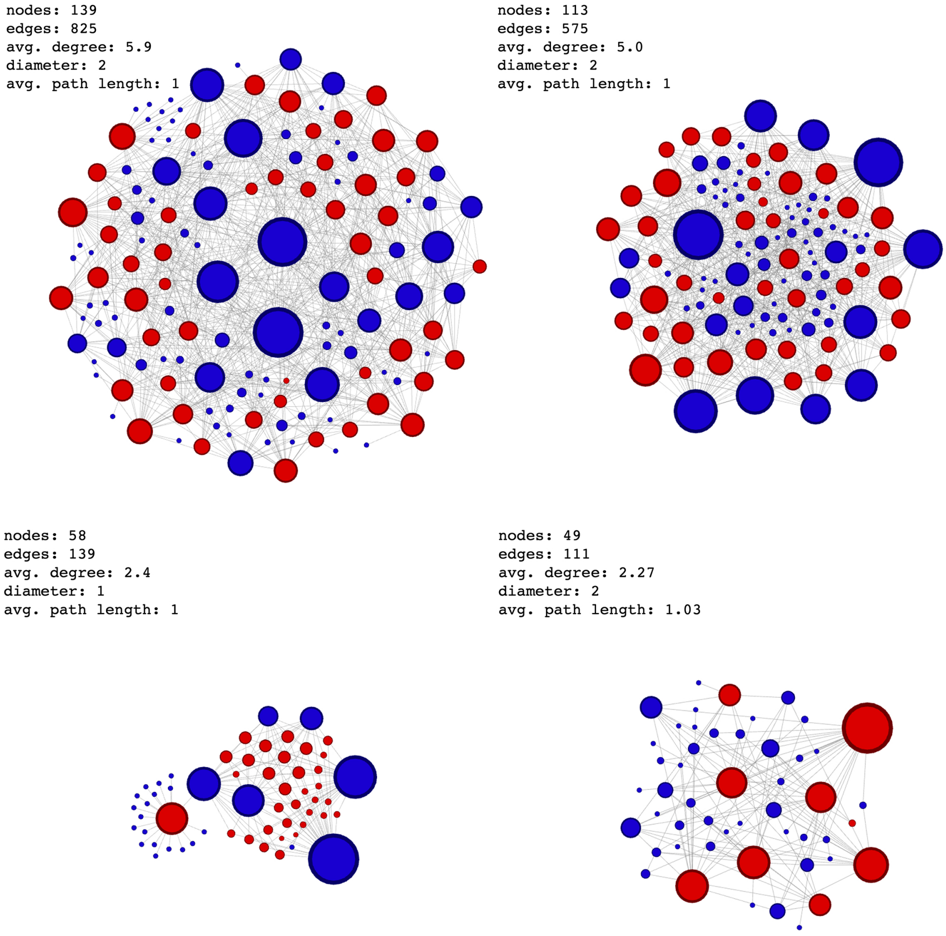


Figure 5.3: Web-service data flow graphs involving first parties (red nodes) and second/third parties (blue nodes)

The colour of the nodes indicates the nature of the node. First parties are represented in red, third parties in blue. The size of a node is an illustration of its relative centrality. In total, there were 825 data flows between first and third parties.

The network illustration in figure 5.3 a) illustrates the graph encapsulating all information flow types. We delineate networks comprised exclusively of mandatory, optional and covert flows in figures 5.3 b), c) and d) respectively. 81% of data processors from figure 5.3 a) implemented mandatory flows as their data sharing principle, which is illustrated in b). 41% and 35% of data processors relied on optional and covert flows as their data sharing principles shown in c) and d) respectively.

The dominant form of data processing omits the elicitation of consent from subjects prior to the processing of information. Such activity involves the presentation of notice to end users that data processing activity takes place, but does not present subjects the capacity to reject or otherwise object to the data processing activity.

The covert information-flow graph in figure 5.3 d) indicates 2 key things. Firstly it indicates to us privacy conflicts can be multi-faceted. This is because data pertaining to the subject may still be disseminated irrespective of whether a subject rejects or accepts data processing activity. Secondly it indicates a challenge with consent elicitation mechanisms. We observed scenarios where the third party lists presented to data subjects were incomplete. In such instances it will not be possible for subjects to exert control over the data processing involving external entities as such information is omitted from the consent module. In such an instance, our method was able to identify subsequent information flow occurring to such third parities as covert flow.

Initial data from the network elicitation suggests expected privacy loss may be different from actual privacy loss subjects experience when visiting first parties online. The statistics indicate a significant amount of data subject tracking may occur online, irrespective of whether such data processing activity is subsidised by any legal processing requirement, which exemplifies the problem of misaligned privacy expectations of data subjects. The challenge however, is the difficulty for end users to measure their privacy expectations and evaluate them against actual privacy loss occurring when they interact with first parties. Because of this, it is important for system designers to be able to understand how such misalignment's can manifest as privacy conflicts in the design of such systems.

5.3 Summary

In this section we have introduced a three step methodology allowing analysts to construct an information flow network representation of the data processing activity between different entities along with their specific legal basis for data processing. We unpacked three distinct basis for data processing, and articulated how we can identify such basis when performing an autonomous web crawl of first parties.

This methodology and workflow will help analysts better align with the principle of privacy by design in the review of how privacy consent modules are instrumented in an interdependent setting. The approach allows stakeholders to acquire insight on instances where data processing occurs without appropriate legal basis, thereby offering insight on potential incidents of non compliance with data protection regulations such as GDPR. This is an important concern if the objective of analysis is to better demonstrate compliance with privacy legislation. The capacity to highlight aspects of a software design leading to non-compliance is an emergent contribution of the work.

An additional outlook on the approach involves the refactoring of software systems. When the objective of the analysis is to refactor the design of software to mitigate privacy risks faced by data subjects, this work can be expanded on to further consider how privacy risk manifests given the instrumentation choices made by first and third parties. The next chapter of this thesis leverages on the information flow network model output from this case study to perform privacy conflict analysis on the elicited model.

Chapter 6

Case Study: Conflict Analysis

6.1 Introduction

The previous chapter detailed a method to elicit an information flow network representing different observable data processing behaviours between first and third parties. Distinct behaviours correspond to specific instrumentation choices for data CMPs. For example, optional data flows are characterised by the instrumentation of consent elicitation modules for the purpose of capturing data subject preferences on how their data should be disseminated. Mandatory and covert data flows correspond to mechanisms providing, or withholding notice to subjects respectively.

The manner by which users interact with CMPs will inform their privacy requirements. However, satisfying user privacy concerns is not always a straight forward consideration for system engineers as the instrumentation of privacy consent modules is often inter-dependant. As such, when data subjects interact with first parties connected by a common third party, the resulting consent preferences (whether explicit or implied) are aggregated by third parties involved in the dissemination of user data.

Assume we have TP, FP_1, FP_2 where TP represents a third party data processor and FP_n represents a distinct first party data processor. Conflict can be observed in the event a subject su grants consent to TP when visiting FP_1 but denies consent to TP when visiting FP_2. This mutual exclusivity means preference aggregation will be difficult as the conflicting preferences of su cannot be reconciled by TP.

Furthermore, the nature of the CMPs informs privacy expectations. For example, covert and mandatory data flows operate on 'implied consent' where it is assumed data subjects will agree to data processing activities when they visit a first party web service. Consider a third party and first party (TP, FP respectively) where a subject su denies consent to TP directly, but visiting FP is either forced to accept data processing involving TP, or the data processing activity occurs covertly.

Data subject expectations and data processing activity do not align in such a scenario. Further, the capacity for the subject to reason about their privacy in mandatory and covert data flow scenarios is significantly reduced. Mandatory interactions necessitate reading numerous privacy policies, whilst covert interactions occur in secret. Additionally, in optional data flow scenarios subjects can indicate interdependent preferences that are mutually exclusive, or misalign with the consequences of prior (or future) interactions with other data processors. Therefore the expected level of privacy loss when utilising web services online may not reflect the actual privacy loss experienced by a data subject.

There is a need therefore, for analysts to computationally measure the knowledge gain when a data subject were to interact with data processors. To this end, we aim to address RQ3 in this section by introducing a method for analysts to scrutinise the interactions between a data subject, first and third party data processors. We introduce privacy anti-patterns, to articulate both a usage context, along with the privacy assertions from this usage context. We leverage on the awareness formalism from section 4 to illustrate how an interaction context will lead to misaligned privacy expectations by formally expressing the logical consequence of the interactions. This in turn can be used by analysts, to objectively measure the efficacy of remedial actions taken to reduce privacy risk. This allows stakeholders to demonstrably support data subjects by measuring these objective properties of an information flow network to spearhead their analysis to reduce privacy risk.

6.2 Privacy Anti-Patterns

Privacy anti-patterns encapsulate two concerns. Firstly, they articulate a specific usage context, facilitated by the instrumentation of subject control mechanisms between a first and third party. Secondly, they express a logical consequence of the usage context. Formally, this is articulated as an assertion which maps to the modified tenability of an objects awareness. Assertions validate the existence of conflict by cross referencing incompatible memory states in the memory stores of their software objects. Each anti-pattern is distinguished by a combination of usage context and formal awareness semantics indicating the logical consequence of such interactions. Each context is unique, however the logical consequences are not. Meaning, two unique anti-patterns may have the same (formalised) consequence.

Identifying Anti-Patterns

Anti-patterns are a result of observations made on the constructed information flow network G from section 5. We analysed possible interaction contexts permissible by the corresponding data processing behaviour of first and third parties. Each data processing principle results in the modification of (un)awareness tenability for both data processors and subjects. Data processors acquire knowledge as a result of data processing activity, whilst data subjects formulate expectations from their interactions with subject control mechanisms. Given an interaction context involving a first and third party, we consider how misaligned privacy expectations between the data subject and processors can be facilitated by the data sharing principles of F and T.

The procedure to identify anti-patterns is illustrated in algorithm 5:

Algorithm 5: Process followed to identify conflict patterns on a given information flow edge.

```
$\underline{Analyse}$
Input  : Annotated edge $E(V_1, V_2)$
Output: Pattern Collection $P$
let $V_{FP} = \{\}$
let $V_{TP} = \{\}$
let $P = \{\}$
for $V \in V_1, V_2$ do
    if true $\leftarrow optional(V) and\ not\ Inert(V)$ then
        $\Delta \leftarrow Accept(V)$
        $getDb(V) \cup \epsilon(optional, accept, \Delta)$
        $\Delta' \leftarrow Reject(V)$
        $getDb(V) \cup \epsilon(optional, reject, \Delta')$
    else
        $\Delta \leftarrow Execute(V)$
        $getDb(V) \cup \epsilon(\beta, -, \Delta)$
for $i \in V_{FP}$ do
    for $j \in V_{TP}$ do
        if $Conflict(V_{FP}[i], V_{TP}[j])$ then
            $desc \mapsto context(V_{FP}[i], V_{TP}[j])$
            $p \mapsto (desc, V_{FP}[i][\Delta], V_{TP}[j][\Delta])$
            $P \cup p$
return $P$
```

This procedure operates on an edge $E(V_1, V_2) \in G$ such that V_1 and V_2 represent a first and third party respectively. We start with identifying the first and third parties within the edge. Next, we look to inspect the parameterised behaviour of V_1 and V_2. This is achieved by inspecting the annotations on the constructed edge as described in process 4.

If the data subject is presented with a consent elicitation module owed to optional data processing principles, the subject can either accept or reject data processing. Their decision will inform the privacy assertions held by their software object. It is not known at design time which outcome (and thusly which logical consequence) will be observed. Therefore, we consider both outcomes. For instance, if an edge connects a first party offering consent modules to a third party exhibiting mandatory data processing there are two possible outcomes:

1. Data subject assertions are generated when the subject accepts data processing on the first party reflecting their privacy expectations. Assertions are generated by the third party as a result of mandatory data processing behaviour reflecting their knowledge acquisition.
2. Data subject assertions are generated when the subject rejects data processing on the first party reflecting their privacy expectations. Assertions are generated by the third party as a result of mandatory data processing behaviour reflecting their knowledge acquisition.

If both the first and third party offered consent elicitation modules there would be four possible outcomes concerning the distinct consent combinations. We first check to see whether the identified behaviour of a data processor (V) is optional and whether the data processor is inert per line 6. Inert data processors default to the lawful basis of data processing of their associated first party and do not afford us the capacity to elicit logical consequences of data subject interactions. If both checks on line 6 are true, we elicit the privacy assertions from the data flows resulting from the data subject opting into data processing. The function Accept(V) returns a collection of privacy assertions Δ where each assertion $\alpha_i \in \Delta$ maps to a given Awareness classification A_n. The assertions returned in Δ correspond to the newly tenable (un)awareness constructs in the memory stores of the object and are detailed in table 6.1.

These assertions are added to a tuple ϵ which takes three parameters β, γ and Δ, the behaviour of V, the (optional) interaction context and the assertion collections respectively. In line 8 β and γ are parameterised as 'optional' and 'accept' due to data flow being optional, with Δ mapping to the output of the interaction function. ϵ is added to the set V_{FP} or V_{TP} depending on whether it is a first or third party data processor. On line 9 we perform the operation again, except we elicit assertions returned when the data subject rejects data processing on the consent modules offered by V. The collection Δ' represents the newly tenable (un)awareness instances in the memory stores of objects when the user rejects processing, which is distinct from Δ. We create a tuple ϵ' such that $\beta =$'optional' and $\gamma =$'reject' and $\Delta = \Delta'$ before adding it to either V_{FP} or V_{TP} per the 'getDB' function.

If the behaviour of V is not optional, we acquire the logical consequence of the resulting data flows when the subject interacts with V. For instance, covert data flows will generate less awareness assertions in the memory store of the data subject compared to mandatory data flow, owed to the decrease in transparency. We acquire the generated assertions (Δ) from the 'Execute' function and generate another tuple ϵ. The parameter β is dependant on the data processing behaviour of V being covert or mandatory. The second parameter γ is not evaluated as neither behavioural context is optional.

Once we populate the two assertion collections V_{FP} and V_{TP}, we perform a pairwise comparison within these collections to check for conflict. The conflict function takes as input a pair of assertions, and will return true in the event the assertions generated conflict per the semantic rules discussed in section 4.5.1. If the function returns true, we generate a natural language description of a use case involving a data subject, $V1$ and $V2$ resulting in conflict being detected. We encapsulate the description, and conflicting assertions within a pattern object p accepting three parameters $desc, \Delta, \Delta'$ where $desc$ is the natural language description of the conflict and Δ, Δ' represent the assertion collections from the respective ϵ tuple structure. We add the pattern to a collection P which will be returned. The output of this approach is a conflict pattern collection illustrating how a data subject can interact with a first and third party to generate privacy conflict and therefore increase privacy risk.

The interdependence of data processing activity means interactions a data subject has with multiple first parties can generate privacy assertions leading to misaligned privacy expectations. We articulate a second procedure in 6 for the systematic generation of privacy conflicts on an information flow network as opposed to a single edge within the information flow network.

Algorithm 6: Approach to identify conflict patterns on an information flow network graph G.

```
Input  : Information flow network model $G$
Output: Pattern Collection $P$
1 let $P = \{\}$
2 for $E(V_1, V_2), E'(V_1', V_2') \in G$ do
3     if $V_1 \neq V_1'$ and $V_2 = V_2'$ then
4         $P \cup Analyse(V_1, V_2)$
5     return $P$
```

The procedure takes as input an information flow network model G and outputs a series of privacy anti-patterns associated with the observable interactions on the edges within G. We start by taking a pairwise comparison on each edge E, E' as seen in line 2. Each edge is associated with a source and target vertice V_1, V_2 respectively. On line 2 we check if V_1 and V_2 are distinct and whether V_1' is the same third party as V_2'. If true, we have two first parties sharing a common third party. The objective is to consider those privacy anti-patterns involving a data subject interacting with V_1 and V_2 independent of the third party. To achieve this, we take V_1 and V_2 and input them into procedure 5. The output from this process in line 4 will be joined with P. We perform this check for each possible edge comparison within G and output P as a result.

The output of procedures 5 and 6 enables stakeholders to both identify and categorize conflicts on the information flow model. Being able to understand the usage context associated with privacy conflicts will assist in developing appropriate mitigation approaches to the design of software systems.

Conflict Mapping

The assertions generated during procedures 5 and 6 reflect the expectation of the data subject and the knowledge gain of data processors. Formally, these are expressed as newly tenable (un)awareness instances in the awareness memory stores of their software object M_p. A data subject is said to remain uncertain regarding the consequences of their interactions with data processors if the interaction context results in tenable (un)awareness instances aligning with the uncertainty conflict semantics discussed in section 4.5.3. For instance, with an initial starting memory state M_{su}, a data subject engaging the first party and accepting data processing activity through consent mechanisms will affect M_p such that the following awareness instance[1] becomes tenable for the data subject:

$$A_{su} A_T su.f$$

This illustrates awareness the data subject has over the awareness the third party T has regarding their attribute $su.f$. The assertion is a logical consequence of the interactions the data subject has with the consent modules offered by the first party. Therefore it is un-intuitive to consider the following assertion tenable:

$$A_{su} \neg A_T su.f$$

As the data subject has acquired an understanding of the awareness state of the third party T after accepting data processing on F. If the first party did not enact data processing transparency and instead data processing was covert, the subject will not stand to gain an understanding of the knowledge acquisition of the third party. Therefore, both assertion $A_{su} A_T su.f$ and assertion $A_{su} \neg A_T su.f$ will be tenable as the subject did not acquire information to discount $A_{su} \neg A_T su.f$ being tenable. per uncertainty conflict, this indicates the data subject will remain uncertain over the awareness the third party data processor possesses regarding their data attributes. This uncertainty conflict is formally expressed when both $A_{su} A_T su.f$ and $A_{su} \neg A_T su.f$ remain tenable in M_p.

Each interaction scenario involving a first and third party will result in changes to the tenability of (un)awareness instances which affect M_p. Table 6.1 illustrates the assertions returned

[1] This assertion is not an exhaustive example, the example presented here was selected for illustrative clarity. A full breakdown is illustrated in table 6.1.

Table 6.1: M_p awareness mutations that occur when enacting data processing activities

Mandatory(F)	**Optional Accept**(F)	**Optional Reject**(F)	**Covert**(F)
$A_F su.f$	$A_F su.f$	$A_F su.f$	$A_F su.f$
$A_F A_T su.f$	$A_F A_T su.f$	$A_F \neg A_T su.f$	$A_F A_T su.f$
$A_{su} A_F su.f$	$A_{su} A_F su.f$	$A_{su} A_F su.f$	
$A_{su} A_T su.f$	$A_{su} A_T su.f$	$A_{su} \neg A_T su.f$	
$A_{su} A_F A_T su.f$	$A_{su} A_F A_T su.f$	$A_{su} A_F \neg A_T su.f$	
$A_T su.f$	$A_{su} A_T su.f$	$\neg A_T su.f$	
$A_F A_{su} A_T su.f$	$A_F A_{su} A_T su.f$	$A_F A_{su} \neg A_T su.f$	
Mandatory(T)	**Optional Accept**(T)	**Optional Reject**(T)	**Covert**(T)
$A_T su.f$	$A_T su.f$	$\neg A_T su.f$	$A_T su.f$
$A_{su} A_T su.f$	$A_{su} A_T su.f$	$A_{su} \neg A_T su.f$	

by the functions 'Accept', 'Reject' and 'Execute' from procedures 5 and 6 representing newly tenable (un)awareness instances in the memory stores of their respective objects. Explicit and implicit conflict arise in a given anti-pattern in the event the usage contexts results in tenable assertions matching associated semantics for explicit and implicit conflict (sections 4.5.1 and 4.5.2 respectively). Uncertainty conflict arises in the event the lack of tenability changes in the memory store of a principal results in the tenability of assertions that match the criteria discussed in section 4.5.3.

Table 6.2: Processor states that generate privacy conflicts

Anti-pattern	Data sharing principle						
	Optional		Mandatory		Covert		Inert
	F	T	F	T	F	T	T
1	X$_r$	X$_a$					
2	(X$_r$, X$_a$)						X
3		(X$_r$/X$_a$)			X		
4					X	X	
5	X$_a$	X$_r$					
6	(X$_r$/X$_a$)					X	
7	X$_r$			X			
8		X$_r$	X				
9				X	X		
10			X			X	
11					X		X
12/13	X$_r$		X$_{F'}$				X

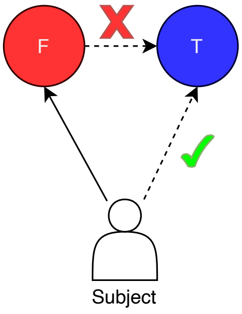
F- First party, T- Third party, X- active state
X$_r$- subject reject processing state, X$_a$ -subject accept processing state
F'- First party such that $F \neq F'$, X$_{F'}$ -subject accept processing state

Table 6.2 presents an overview of the anti-patterns and how these anti-patterns are mapped to data sharing principles of a first party F and a third party T. Each row indicates the usage scenarios involving a data subject, with the columns indicating the possible data sharing principles enacted by the first or third party (leading to optional, mandatory or covert information flows). Each cell indicates whether a given first and third party implement mechanisms aligning with these principles, and whether or not the subject accepts or rejects data processing in the event of optional data flows. The final column is used to indicate whether the third party is inert. Each anti-pattern is articulated with a visual representation, formal definition and associated privacy assertions resulting from the interactions data subjects will have with the instrumented privacy consent features.

6.2.1 Anti-Pattern Breakdown

Anti-Pattern 1

Anti-pattern 1 is characterised as an interaction scenario involving a data subject su, first party F and third party F such that the interactions between F and T are facilitated as optional flows. The interaction scenario is illustrated in figure 6.1.



Privacy Anti-Pattern 1

Figure 6.1: Privacy Anti-Pattern 1

In this pattern subject consent is the sole legal basis for data processing and as such, both T and F make available appropriate mechanisms to su. Conflict occurs per row 1 in table 6.2 where su rejects data processing by T on the mechanisms available on F, but accepts processing by T on the mechanisms offered by T directly. A natural language definition of this anti-pattern is as follows:

Description: Given a first party F listing a third party T on its consent toggle. If T provides consent toggles, the subject su rejects data processing by T on the consent toggle provided by F, but accepts on T directly.

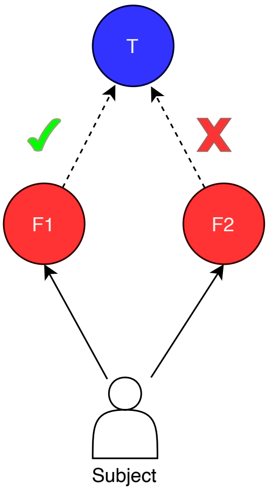
The following awareness constructs become tenable in this interaction scenario facilitating privacy conflict:

Privacy Conflict:
True $\leftarrow \mathit{\Delta}(A_{su}A_F\neg A_T su.f, A_{su}A_T su.f)$
True $\leftarrow \mathit{\Delta}(A_F\neg A_T su.f, A_{su}A_T su.f)$
True $\leftarrow \mathit{\Delta}(\neg A_T su.f, A_T su.f)$

The function $\Delta(a, a')$ returns True if the assertion a conflicts with a' explicitly or implicitly, otherwise it returns False.

Privacy Anti-Pattern 2

Anti-Pattern 2 is characterised as an interaction scenario involving the data subject su, a first party F and a second first party F' each connected to a common third party T. The interaction scenario is illustrated in figure 6.2.



Privacy Anti-Pattern 2

Figure 6.2: Privacy Anti-Pattern 2

Here, again consent is the sole basis for data processing. Therefore both F and F' provide consent toggles for su to express their preferences. In this scenario, conflict occurs per row 2 in table 6.2 such that su rejects data processing by T on the F but will reject data processing by T on the privacy consent toggle offered by F'. A natural language definition of this anti-pattern is as follows:

Context: given the first parties F and F' listing a third party T on their consent toggles. The subject su rejects data processing by T on the consent toggle provided by F', but accepts on F.

The following awareness constructs become tenable in this interaction scenario which facilitate the privacy conflict:

Privacy Conflict:

$$\begin{aligned}
\text{True} &\leftarrow \Delta(A_{su}A_F\neg A_T su.f, A_{su}A_{F'}A_T su.f)\\
\text{True} &\leftarrow \Delta(A_F\neg A_T su.f, A_{F'}A_T su.f)\\
\text{True} &\leftarrow \Delta(\neg A_T su.f, A_T su.f)\\
\text{True} &\leftarrow \Delta(A_F A_{su}\neg A_T su.f, A_{F'}A_{su}A_T su.f)\\
\text{True} &\leftarrow \Delta(A_{su}\neg A_T su.f, A_{su}A_T su.f)
\end{aligned}$$

Privacy Anti-Pattern 3

Anti-Pattern 3 is characterised as an interaction scenario involving the subject su, a first party F, third party T and is visualised in figure 6.3

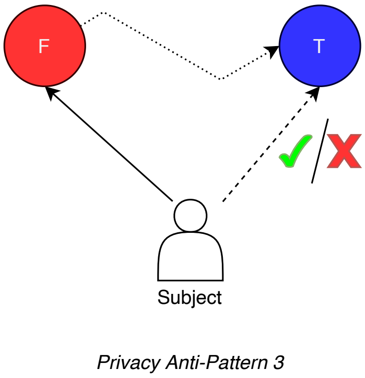


Figure 6.3: Privacy Anti-Pattern 3

This interaction scenario involves the implementation of optional flows between T and su, and covert flows between F and T. In this scenario, we state T provides consent mechanisms to su directly, as a member of EDAA, thus it is assumed su preferences will be respected. However, F does not operate any form of data processing transparency and discloses attributes pertaining to f to T when su interacts with F. Therefore, there is a covert data flow connecting F and T. Conflict arises in this scenario per row 3 in table 6.2 due to the uncertainty conflict regarding the inability for su to distinguish what F and T know or do not know about su. Irrespective of whether su accepts or rejects data processing on T, the covert activities of F will generate uncertainty. A natural language definition of this anti-pattern is as follows:

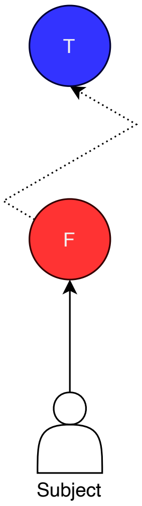
Context: Given the subject su, first party F and third party T. If su accepts or rejects data processing on consent mechanisms offered by T and subsequently interacts with F. As F does not support data processing transparency, a covert data flow occurs from F to T. The following assertions become tenable in this scenario:

`Privacy Conflict:` $$\begin{aligned} \text{True} &\leftarrow \lambda(A_{su}A_F su.f, A_{su}\neg A_F su.f) \\ \text{True} &\leftarrow \lambda(A_{su}A_F A_T su.f, A_{su}\neg A_F A_T su.f) \\ \text{True} &\leftarrow \lambda(A_{su}A_F A_T su.f, A_{su}A_F\neg A_T su.f) \\ \text{True} &\leftarrow \lambda(A_{su}A_T A_F su.f, A_{su}\neg A_T A_F su.f) \\ \text{True} &\leftarrow \lambda(A_{su}A_T A_F su.f, A_{su}A_T\neg A_F su.f) \end{aligned}$$

The function $\lambda(a, a')$ returns True in the event the tenability of a and a' generates uncertainty in M_p, otherwise it returns False.

Privacy Anti-Pattern 4

Anti-Pattern 4 is an interaction scenario involving a subject su, first party F and third party T. The interaction scenario is visualised in figure 6.4



Privacy Anti-Pattern 4

Figure 6.4: Privacy Anti-Pattern 4

In this scenario, neither F or T follow any data sharing principle and will act in a covert manner with su when su interacts with F online. In this scenario, it is assumed T is inert, where it is not reachable by su and in turn, will defer to the covert dissemination approach of F. Conflict arises in this scenario per row 11 in table 6.2 due to the uncertainty conflict regarding the inability for su to distinguish what F and T know or do not know about su. A more formal description is presented as follows:

`Context:` Given a subject su, a first party F not supporting transparency of data processing, and an inert third party T. Interactions between su and F will result in covert data flow from F to T. The privacy assertions which facilitate conflict are as follows:

Privacy Conflict: $\text{True} \leftarrow \lambda(A_{su}A_F su.f, A_{su}\neg A_F su.f)$

$\text{True} \leftarrow \lambda(A_{su}A_F A_T su.f, A_{su}\neg A_F A_T su.f)$

$\text{True} \leftarrow \lambda(A_{su}A_F A_T su.f, A_{su}A_F\neg A_T su.f)$

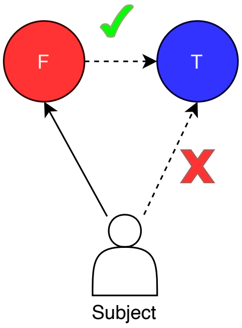
$\text{True} \leftarrow \lambda(A_{su}A_T su.f, A_{su}\neg A_T su.f)$

$\text{True} \leftarrow \lambda(A_{su}A_T A_F su.f, A_{su}\neg A_T A_F su.f)$

$\text{True} \leftarrow \lambda(A_{su}A_T A_F su.f, A_{su}A_T\neg A_F su.f)$

Privacy Anti-Pattern 5

Anti-Pattern 5 is an interaction scenario involving a subject su, first party F and third party T. A visual representation of this scenario is presented in figure 6.5.



Privacy Anti-Pattern 5

Figure 6.5: Privacy Anti-Pattern 5

This scenario involves the implementation of optional flows exclusively between F and T. In this scenario, the subject accepts data processing by T on the mechanisms made available by F, but rejects data processing directly on the mechanisms offered by T. Conflict per row 5 in table 6.2 arises due to the misaligned privacy expectations of users, and is the opposite case to anti-pattern 1 as su accepts data processing on F but rejects on T. A scenario description is as follows:

Context: Given a first party F listing the third party T on its consent toggle. su accepts data processing by T on F but rejects on mechanisms directly offered on T. The tenable assertions facilitating conflict are as follows:

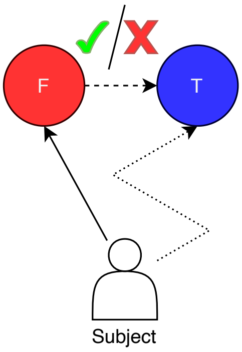
Privacy Conflict: $\text{True} \leftarrow \Delta(A_{su}A_F A_T su.f, A_{su}\neg A_T su.f)$

$\text{True} \leftarrow \Delta(A_F A_T su.f, A_{su}\neg A_T su.f)$

$\text{True} \leftarrow \Delta(A_T su.f, \neg A_T su.f)$

Privacy Anti-Pattern 6

Anti-Pattern 6 is a scenario involving a subject su, first party F and third party T illustrating the challenge of incomplete privacy consent mechanisms. The interaction is visualised in figure 6.6



Privacy Anti-Pattern 6

Figure 6.6: Privacy Anti-Pattern 6

In this scenario su interacts with T directly to deny consent to data processing on T. Subsequently, su interacts with F where F either does not list T on its consent options, or does not provide consent options to su. In either case, the data processing will be covert in nature, as su is not made aware of the data processing activity. Conflict arises per row 6 in table 6.2 due to the uncertainty conflict regarding the inability for su to distinguish what F and T know or do not know about su. Irrespective of whether su accepts or rejects data processing on F, the covert data processing on T creates uncertainty. A natural language definition of this anti-pattern is as follows:

`Context:` Given the subject su and third party T. su accepts or rejects data processing on consent mechanisms directly offered on T. F does not support transparency of data processing. Hence, su interaction with F results in covert data flow from F to T. The tenable assertions (irrespective of which interaction scenario) facilitating conflict are as follows:

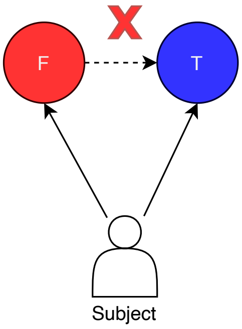
`Privacy Conflict:` $\mathsf{True} \leftarrow \lambda(A_F A_T su.f, A_{su} \neg A_T su.f)$

$\mathsf{True} \leftarrow \lambda(A_{su} A_T A_F su.f, A_{su} \neg A_T A_F su.f)$

$\mathsf{True} \leftarrow \lambda(A_{su} A_T A_F su.f, A_{su} A_T \neg A_F su.f)$

Privacy Anti-Pattern 7

Anti-Pattern 7 is an interaction scenario involving the subject su, first party F and third party T and is illustrated in figure 6.7.



Privacy Anti-Pattern 7

Figure 6.7: Privacy Anti-Pattern 7

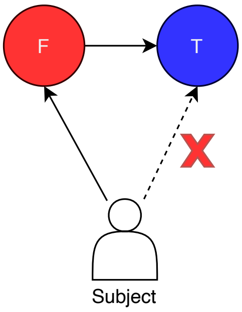
This interaction involves the incompatible instrumentation of mandatory and optional data flows. Here, F provides consent mechanisms to su, which allows su to opt out of data processing occurring on T. However, T does not provide opt out mechanisms to su. These instrumentations are fundamentally incompatible with one another. Conflict per row 7 in table 6.2 arises from misaligned privacy expectations similar to anti-pattern 1 as su rejects processing on F but is forced to accept on T. A description of this scenario is as follows:

`Context`: Given a first party F listing the third party T on its consent toggle. The subject su rejects data processing by T on F, whereas T provides data processing transparency to su it does not allow su to opt-out of processing. The tenable assertions of this scenario are as follows:

`Privacy Conflict`: Same as anti-pattern 1.

Privacy Anti-Pattern 8

Anti-Pattern 8 is an interaction scenario involving the subject su, first party F and third party T and is illustrated in figure 6.8.



Privacy Anti-Pattern 8

Figure 6.8: Privacy Anti-Pattern 8

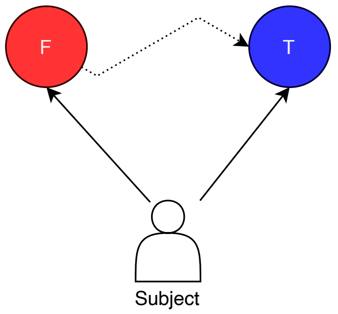
This scenario involves the instrumentation of mandatory and optional data flows in an incompatible manner. Here, su interacts with the consent mechanisms offered by T to opt out of the data processing activities performed by T. However, when su interacts with F, they do not provide any consent elicitation methods, despite providing transparency notice to su. Conflict occurs per row 8 in table 6.2 where su rejects data processing by T on the mechanisms available on T, but is forced to accept processing by T when visiting F. This is similar to anti-pattern 7.

`Context:` Given the subject su and third party T. Then, su rejects data processing on consent mechanisms directly offered on T. The first party F only presents its privacy policies to support transparency of data processing without enabling su to opt-out. The tenable assertions generating conflict are as follows:

`Privacy Conflict:`
$$\begin{aligned}\text{True} &\leftarrow \Delta(\neg A_T su.f, A_T su.f)\\ \text{True} &\leftarrow \Delta(A_{su}\neg A_T su.f, A_{su}A_T su.f)\\ \text{True} &\leftarrow \Delta(\neg A_T su.f, A_T su.f)\end{aligned}$$

Privacy Anti-Pattern 9

Anti-Pattern 9 is an interaction scenario involving a subject su, first party F and third party T and is illustrated in figure 6.9:



Privacy Anti-Pattern 9

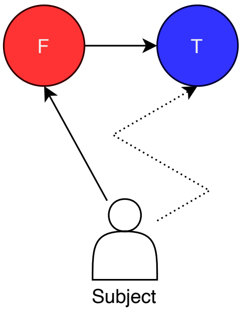
Figure 6.9: Privacy Anti-Pattern 9

This scenario involves the use of mandatory and covert data processing. Here, su is presented with data processing information from T without the option to opt-out of this data processing activity. In addition, F does not support data processing transparency, and when su interacts with F, covert information disclosure occurs between F and T. Conflict arises in this scenario per row 9 in table 6.2 due to the uncertainty conflict regarding the inability for su to distinguish what F and T know or do not know about su. A more formal description is presented as follows:

`Context:` Given the subject su and third party T. Then T provides data processing transparency but no option to opt-out to su. The first party F does not support transparency of data processing. Hence, when su interacts with F covert data flow occurs from F to T. The assertions generated from this interaction scenario are as follows:
`Privacy Conflict:` Same as anti-pattern 4.

Privacy Anti-Pattern 10

Anti-Pattern 10 is an interaction scenario involving the subject su, first party F and third party T and is illustrated in figure 6.10.



Privacy Anti-Pattern 10

Figure 6.10: Privacy Anti-Pattern 10

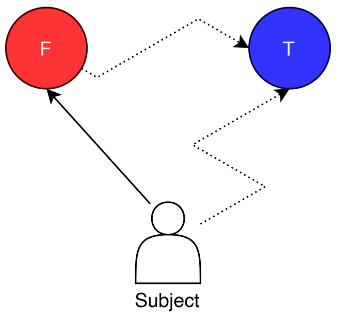
In this scenario su when interacting with F will be forced to accept data processing by T on F due to privacy notices not affording su the capability to opt-out. Likewise, T in this scenario does not follow data sharing principles and interacts in a covert manner with su. Therefore, conflict arises in this scenario per row 10 in table 6.2 due to the uncertainty conflict regarding the inability for su to distinguish what F and T know or do not know about su. A more formal description is presented as follows:

`Context:` Given the subject su and first party F. Then F provides data processing transparency without allowing su to opt-out. The third party T does not support transparency of data processing. Hence, interactions between su and F result in covert data flow from F to T. Tenable assertions facilitating conflict are as follows:

`Privacy Conflict:` $\text{True} \leftarrow \lambda(A_{su}\neg A_T A_F su.f, A_{su}A_T A_F su.f)$
$\text{True} \leftarrow \lambda(A_{su}A_T\neg A_F su.f, A_{su}A_T A_F su.f)$

Privacy Anti-Pattern 11

Anti-Pattern 11 is an interaction scenario involving the subject su, first party F and third party T illustrated in figure 6.11.



Privacy Anti-Pattern 11

Figure 6.11: Privacy Anti-Pattern 11

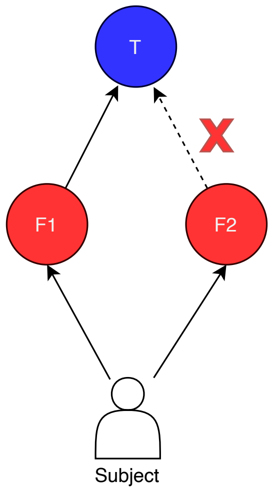
This scenario involves the implementation of covert flows exclusively between F and T. When su visits F, neither F nor T support transparency of data processing activity. Therefore covert data flows occur between F to T. Conflict per row 4 in table 6.2 arises in this interaction scenario because of the covert data processing occurring on both F and T. Therefore su will be unable to distinguish what F and T know or do not know about su. A description of this scenario is as follows:

`Context`: Given the subject su, third party T and first party F. Both T and F do not support transparency of data processing. Therefore, su interacts with F resulting in covert data flow from F to T. The conflicting tenable assertions are as follows:
`Privacy Conflict`: Same as anti-pattern 4.

Privacy Anti-Pattern 12

Anti-Pattern 12 is an interaction scenario involving the subject su, first party F and third party T illustrated in figure 6.12.

In this scenario, $F1$ operates on contractual necessity, and as such exhibits mandatory data processing behaviour with su, providing notice of data processing activities but without the option for su to reject data processing activity. Conversely, $F2$ operates on the lawful basis of informed consent or legitimate interest, such that su is presented with consent elicitation options over data processing activity. T is inert in this scenario and defaults to the data processing principles of $F1$ and $F2$. Conflict occurs due to the misaligned expectations su possesses regarding the knowledge state of T. A more formal description is as follows:



Privacy Anti-Pattern 12

Figure 6.12: Privacy Anti-Pattern 12

`Context:` Given a subject su, two distinct first parties $F1, F2$ operating mandatory and optional data processing principles respectively and an inert third party T. Interactions between su, $F1$ and $F2$ create uncertainty conflict in the event su were to reject data processing on the consent modules offered by $F1$. The tenable assertions facilitating conflict are as follows:

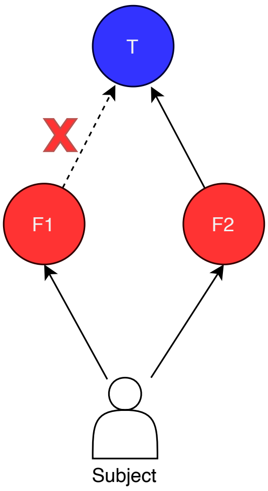
`Privacy Conflict:` Same as anti-pattern 8.

Privacy Anti-Pattern 13

Anti-Pattern 13 is an interaction scenario involving the subject su, first party F and third party T illustrated in figure 6.13.

In this scenario, $F1$ enacts a data sharing principle of informed consent and provides su with the capacity to opt out of data processing activity in T. Conversely, $F2$ enacts the data sharing principle of contractual necessity, and therefore provides su with information processing transparency but no technical means to opt out of data processing activity. T is inert in this scenario and defaults to the data processing principles of $F1$ and $F2$. Conflict occurs due to the misaligned expectations su possesses regarding the knowledge state of T. This anti-pattern is the opposite of anti-pattern 12. A more formal description is as follows:

`Context:` Given a subject su, two distinct first parties $F1, F2$ operating optional and mandatory data processing principles respectively and an inert third party T. Interactions between su, $F1$ and $F2$ will create uncertainty conflict in the event su were to reject data processing on the consent modules offered by $F1$. The tenable assertions that facilitate conflict are as follows:



Privacy Anti-Pattern 13

Figure 6.13: Privacy Anti-Pattern 13

`Privacy Conflict`: Same as anti-pattern 8.

6.3 Privacy Risk

Affiliated third parties may instrument their consent mechanisms independently of the first party service. Therefore it is important for a stakeholder to understand the decision taken by affiliated data processors as their choices can have an impact on the resulting privacy risk faced by users when interacting with any implemented first party consent mechanisms. The best case scenario is a series of interactions with third parties not generating any privacy conflict. Conversely, the worst case scenario involves all interactions with third parties generating conflict. It is important for analysts to understand where such conflicts manifest on an information flow network if the goal of analysis is the reduction of privacy risk.

The capacity to map the privacy awareness formalism to a series of observable anti patterns allows stakeholders to identify privacy conflicts at design time. Quantifying the conflicts on a given first party involves a metric to quantify conflicts. We introduce privacy risk to articulate the potential privacy harm faced by a data subject as a result of misaligned privacy expectations when interacting with first parties.

Privacy risk is defined as a measurement of the exhibited privacy anti-patterns present when interactions occur between a first and third party. In the event a first party F interacts with n third parties we elicit a sub-graph S from figure 5.3 a) such that $S = \{F, T_1, ...T_n\}$ where T_i represents a distinct third party associated with F. Assuming further data processing activity occurs from a given element $T_i \in S$ we can expand S to include additional data processors.

Further, S can be expanded to include additional third party actors interacting with T_i for the purpose of further data dissemination. Furthermore, additional first parties can be included in S such that all first parties share a common third party element with F.

This elicited sub graph allows a stakeholder to perform a targeted analysis involving those third parties and first parties connected to any system under review, or a broader analysis involving external data processors if desired. A data processor $s_i \in S$ contributes to risk if it facilitates a given anti-pattern AP, specifically if s_i maps to T, F or F' in any of the anti-patterns. The following equation is used to calculate privacy risk facilitated by AP.

$$PrivacyRisk_{AP} = \frac{|\forall s_i \in S : \mathsf{True} \leftarrow \mathit{facilitates}(s_i, AP)|}{|S|} \tag{6.1}$$

The risk value is a ratio of the number of data processors within the sub-graph facilitating at least one privacy conflict anti-pattern. The function facilitates(s_i, AP) returns True in the event the data processor s_i matches one of the entities within the associated anti-pattern AP. Figure 6.14 illustrates the associated risk values across the 51 first parties from the Selenium study.

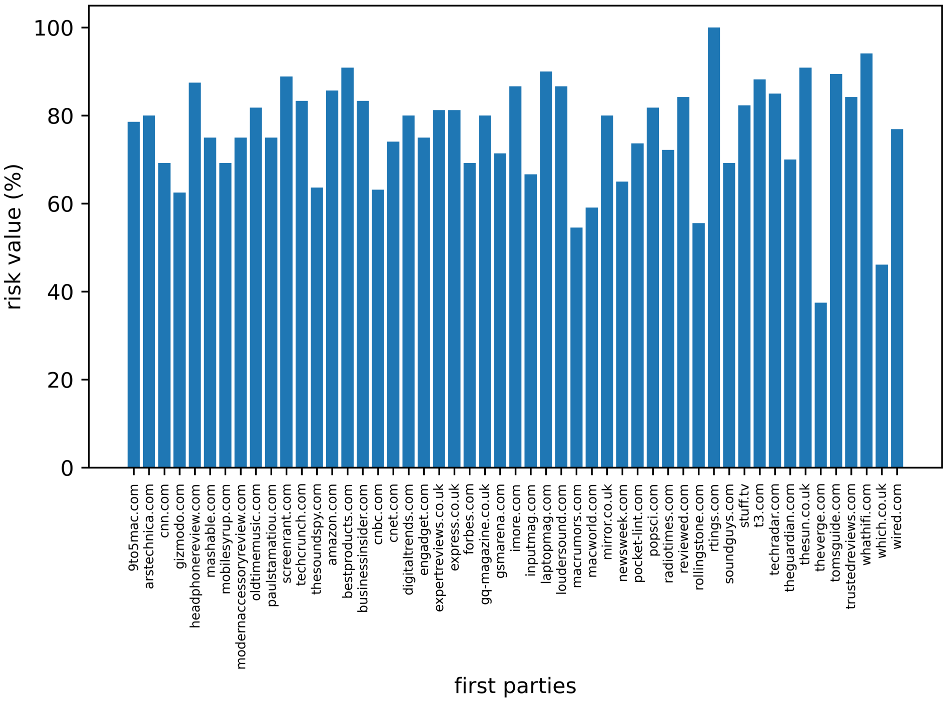


Figure 6.14: Breakdown of risk values for the different first parties

Each bar presents the associated percentile value of risk incurred by the first party. Such values are computed by applying equation 6.1 to each associated third party. This provides insight on the ratio of third parties introducing privacy conflict on an associated first party.

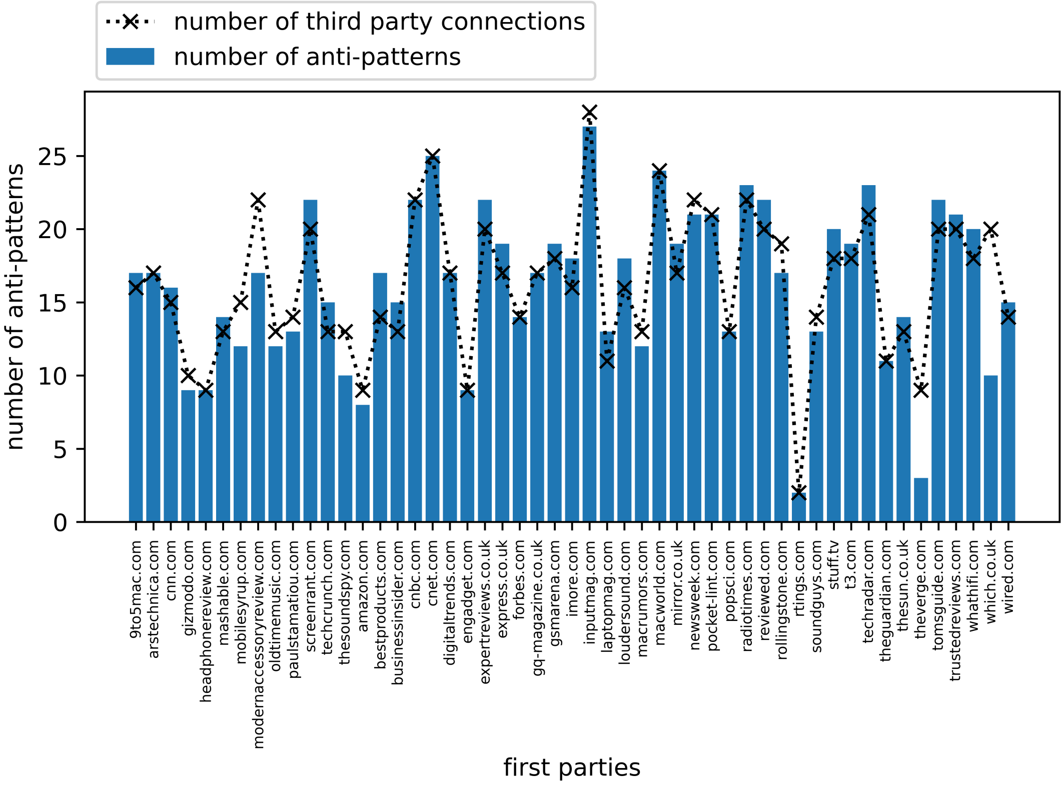


Figure 6.15: Distribution of anti-patterns associated with different first party data processors

The average risk value of 76% suggests the majority of third party data processors facilitate anti-patterns. We note only 2 of the analysed first parties sit below 50% privacy risk, with the minimum observed value being 38%.

From the perspective of design time review, stakeholders can utilise such information to understand which third parties introduce conflict to a first party data processor. This information in turn can be used to inform mitigation efforts if the objective of such analysis is to reduce the level of perceived privacy risk. Additional insight can be acquired by cross referencing the risk values with the statistical information seen in figure 6.15.

The line graph in figure 6.15 illustrates the total number of third party data processors associated with a first party. Each bar represents the cumulative number of anti-patterns facilitated by third parties to provide insight on the impact third parties have on the risk incurred by a data subject when interacting with a first party online.

One immediate observation concerns the total number of connected third parties with the associated privacy risk in figure 6.14. For instance, rtings has two third parties in figure 6.15 but yields the largest risk value in figure 6.14. Conversely, inputmag has the largest number of associated third parties but has a lower risk score of 66%. Bi-variate analysis of these metrics did not determine significance, with a Pearson value of -0.82 and 2 tailed significance value of 0.568. This suggests the total number of third parties does not serve

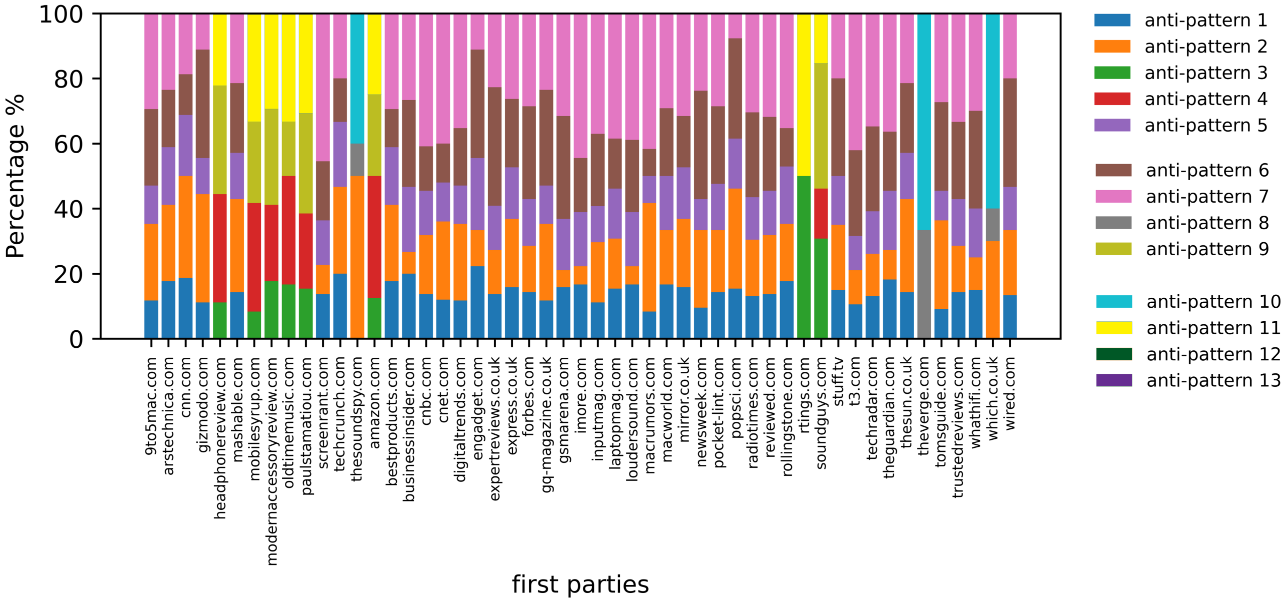


Figure 6.16: Distribution of privacy risks generated by different first-party data processors

as a strong indicator of the inherit risk a data subject may face when interacting with first parties.

The total number of anti-patterns is an interesting observation. The number of anti-patterns is equivalent to the number of third parties for both rtings and endgadget. Interestingly, the risk values for these two first parties differ substantially. Bi-variate analysis indicates a significant relation between the number of anti-patterns and increased risk values. The Pearson score being 0.313 with a significance value of 0.25. We noted 11 instances where the total number of anti-patterns exceeded the number of third parties, confirming multiple anti-patterns are applicable to third parties. In such instances, the average risk value was recorded at 87.2%, an increase over the data set average of 76.4%.

The analysis indicates the risk values observed in figure 6.15 are impacted buy the total number of observed anti-patterns. Given formula 6.1 this is expected. However despite the total number of third party connections relating to an increased number of observed anti-patters, the correlation analysis indicates this does not immediately impact the risk values.

For instance, taking the top 10 first parties exhibiting the highest risk values, and the highest anti-pattern values respectively, we observe only 3 common elements between these two datasets. This suggests an analyst cannot solely rely on the total number of anti-patterns to inform their decision making when mitigating the privacy risk of data subjects.

Further insight can be acquired when we consider the distribution of privacy anti-patterns on a network representation. Figure 6.16 illustrates the distributions observable on all 51 first parties. The bars correspond to a percentile and visually illustrate the prevalence of different anti-patterns introduced by subject control instrumentation choices made by both a first and third party.

One of the most prevalent anti-patterns is anti-pattern 7. The prevalence of this anti-pattern tells us if subjects were to interact with first parties presented in the top pages from a google query $q =$"*beats headphone reviews*" the privacy conflict they are likely to encounter can be characterised as a misalignment of expectations concerning the first and third party data processors. The distributions of total third party values and associated anti-patterns demonstrates the impact of the choices made by first parties when instrumenting their consent mechanisms involving third parties and exemplifies the challenge of interdependent consent mechanisms.

The instrumentation choices made by a first party correspond to the list of third parties involved in the data flows. In the event the analyst is tasked with conducting privacy risk assessment on the choices made by a first party, the breakdown of privacy anti-patterns associated with the third parties provides insight on two key details:

1. Which third party introduces conflict to the first party.

2. What design choices facilitate the conflict involving the first and third party.

The approach further caters to the use case an analyst is responsible for conducting privacy risk assessment on various first parties they are responsible for. The approach affords analysts the capacity to investigate the privacy risk faced by data subjects when interacting with multiple first parties where the outcome of these interactions are aggregated by a common third party. From the data presented in figure 6.16, pattern 2 is the most prominent. Therefore an analyst can conclude the instrumentation of consent modules between two distinct first parties (they may be responsible for) is a design choice more conducive to privacy risk.

Determining the causality between the instrumentation of a software system and the inherit privacy risks introduced by the system is an important consideration in the event the analyst is responsible for investigating potential refactoring to the design of the software pursuant to the mitigation of privacy risk to users. The quantification of privacy risk in turn can be used as an evaluation metric by analysts to consider the efficacy of refactoring such as the removal of third party data processors which introduce or exacerbate an anti-pattern.

6.4 Summary

In this section we have proposed an approach to privacy risk analysis to reconcile design and runtime considerations enabling analysts to investigate how the design choices of a software system, when coupled with runtime usage contexts can facilitate privacy conflict. We articulate a methodology which takes an information flow representation of a software system, and identifies the logical consequence of interactions between a data subject, first and third parties when data flows are enacted. The principle of the method is to consider how the memory of objects interacting on the network will evolve as a result of the interactions. Conflict occurs in the event the evolution of such objects memory stores (i.e. the tenability of (un)awareness instances) happens in a way that is incompatible.

We unpacked a process to identify conflicts by considering how the tenability of (un)awareness instances occur when different data processing activities are instigated. Different data processing activities result in different tenable assertions, which in turn we leverage on to identify different conflicts. We assign a natural language description to explain the usage context to construct a pattern that uniquely articulates a use case resulting in conflicting tenable assertions. Understanding the relation between the instrumentation choices made by a first and third party and the applicable conflicts observable on a system representation in turn allows analysts to identify aspects of the system instrumentation at design time which facilitate privacy conflict at runtime. This in turn allows analysts to consider how specific refactoring of the information flow model can mitigate the privacy risk faced by users. In the next section we will unpack different approaches to conflict refactoring.

Chapter 7

Refactoring Conflict

7.1 Refactoring Risk

Privacy risk is a measurement of the degree of privacy harm a data subject may incur when interacting with a first party resulting from the inconsistencies between the data subjects privacy expectations and the resulting tenable awareness instances of data processors. A key insight provided to analysts from section 6 is the identification of specific third parties facilitating anti-patterns. These third parties in turn contribute to privacy risk associated with a first party data processor. The reduction of privacy risk in turn will lead to a reduction of privacy harm by reducing instances of misaligned privacy expectations involving the data subject. In this work we define a refactoring/mitigation option as the systematic removal of third parties from a first party that facilitate a target privacy anti-pattern. A naive approach to reducing privacy risk will subsequently involve the complete removal of third parties facilitating all observable anti-patterns from a first party. However, such an approach can have an unacceptable negative impact for the first party. Firstly, the technical overhead involved in reconfiguring their data processing models can have a significant time requirement. Secondly, it may not be economically feasible for a first party to simply remove those third parties.

There is a need for analysts to quantify the benefit refactoring a privacy anti-pattern will have to the data subject. Likewise analysts also need to be provided with an approach to quantify the cost of refactoring a privacy anti-pattern for a first party. In this chapter we aim to address research question 4 by first proposing metrics that can be used to model the benefit provided to a data subject and the cost incurred by a data processor when a privacy anti-pattern is refactored. Secondly, we discuss a series of decision making strategies to arrive at a recommendation on which privacy anti-pattern to refactor. The principle of this work is to consider which privacy anti-pattern is the optimal choice subject to preference biases from the analyst.

We begin this chapter by discussing our approach to quantifying the utility for first parties. We leverage on centrality values from the information flow network to consider the impact refactoring activity will have when the third parties facilitating an anti-pattern are removed from a first party. We then discuss an approach to model the utility for data subjects. We then unpack two key refactoring decision strategies. We discuss strategies involving a single decision criterion and articulate the results from our information flow network model. We then unpack an approach to modelling the decision as a multi objective problem and leverage off economic theory to achieve this. We discuss the output of this approach on the information flow network before illustrating the advantages and disadvantages of both approaches. We conclude this chapter with a final approach to decision making allowing for the expression of granular biases and unpack the recommendations across various bias values.

7.2 Quantification of Metrics

In order to quantify the benefit of refactoring anti-patterns on one or multiple first parties, there is a need to articulate metrics to measure the level of benefit to both data subjects and processors. The information flow network representation from section 5 when combined with the anti-pattern descriptions in section 6 provides an analyst with three observable properties:

1. Centrality levels: This value indicates the number of external connections a third party has which facilitates a given anti-pattern.
2. Awareness levels: The awareness levels represent the total number of awareness transforms in the memory stores of principles associated with a given privacy anti-pattern.
3. Prevalence levels: This value indicates the frequency by which a given anti-pattern is facilitated by a third party associated with the first party.

Table 7.1 illustrates a breakdown of these quantifiable properties between the third party data processors and the first party data processor ’inputmag.com’ which yields the highest levels of privacy risk per figure 6.14.

Column 1 represents the third parties connected to the first party whilst column 2 indicates the specific tainting anti-pattern associated with the third party. For clarity, a ’tainting anti-pattern’ is an anti-pattern facilitated between the third and first party. It is not assumed a single third party will only facilitate a single anti-pattern. Column 3 indicates the measured centrality values for the third parties, whilst column 4 illustrates the measurable centrality values when performing various refactoring actions with different anti-patterns. The updated centrality values for the third parties individually are also presented in the respective

Table 7.1: Mitigating privacy risk associated with 'Inputmag.com' by removing third party data processors

Third Party	Tainting anti-pattern	Original centrality	Centrality after refactoring anti-patterns				
			1	2	5	6	7
Google	1,5	51	-	51	-	51	51
Index Exchange	6	38	38	38	38	-	38
Verizon Media	6	33	33	33	33	-	33
Adobe	1,5	27	-	27	-	27	27
Chartbeat	6	10	10	10	10	-	10
Nielsen Co.	1,5	12	-	12	-	12	12
Book.net	6	29	29	29	29	-	29
GoDaddy	7	15	15	15	15	15	-
Criteo SA	6	21	21	21	21	-	21
Smartadserver	7	10	10	10	10	10	-
Amazon	6	24	24	24	24	-	24
Wetter	7	19	19	19	19	19	-
Twitter	7	20	20	20	20	20	-
Skimbit	2	12	12	-	12	12	12
2mdn	2	42	40	-	40	40	40
Crwdcntrl	2	20	20	-	20	20	20
Total observed centrality loss		0	90	74	90	155	64
Prevalence of anti-pattern			3	3	3	6	4
Awareness levels			12	18	9	24	16

columns when anti-patterns are refactored. Third parties removed in the event an anti-pattern is refactored are illustrated with null values '-' in the table as appropriate.

The objective of this table is to illustrate the cumulative values for centrality, prevalence and awareness associated with a particular anti-pattern. This provides analysts with insight to measure the benefit of a refactoring strategy for both data processors and data subjects. We begin by unpacking the rationale behind the metric to evaluate the utility of anti-pattern refactoring for data processors.

First parties have full control over how they instrument their web services and how such services behave with data subjects. The choice to instrument frameworks or specifically, the inclusion of third party scripts into their source code represents a degree of perceived value such third parties provide to the first party. We can leverage on the resulting information flow graph to acquire insight on this level of perceived value. Centrality is a property of an information flow network model underpinning numerous business aspects that contribute to the inherit value a third party possesses to a first party. For instance, third parties can perform numerous functional services such as the procurement of user traffic. External marketing/publishing services exist to increase user traffic on first party web services, the capacity for such third parties to effectively profile users contributing to traffic resulting in an extended customer base lies with their reach which can be elicited with centrality [95]. For e-commerce, third parties can assist in the delivery of targeted advertising which in turn can result in increased sales through the first party platform. Google is an example of an advertisement partner offering such services to first parties within the e-commerce sector. Again such benefit can be estimated based on centrality values [92] [96]. This in turn, allows us to estimate a business value one particular third party may have for a first party in fulfilment of their business objectives. Therefore to quantify the impact of refactoring third parties exhibiting a particular anti-pattern, analysts can examine the centrality levels of those third parties to be removed from the first party. The negative utility incurred by the first party is relatable to the cumulative centrality value of pruned third parties.

The second measurable property of an anti-pattern relates to the levels of prevalence observable on the information flow network. We discussed previously the naive approach to refactoring anti-patterns involves the complete removal of third parties facilitating any anti-pattern on a first party. Such an approach has the capacity to eliminate perceivable privacy risk for a first party, but is not a feasible solution. A suitable alternative is to consider which anti-pattern has the highest levels of prevalence and base refactoring off this as to maximise the reduction of privacy risk whilst minimising the engineering effort required to achieve this.

The challenge with prevalence levels is the assumption each anti-pattern has the same level of impact on the privacy of a data subject. The nature of the interactions a data subject has with a data processor will have an impact on not only the knowledge the data processors acquire pertaining to the attributes of a data subject, but also their capacity to reason about the knowledge states of the data subject, and other data processors. The levels of transparency surrounding the attributes of a data subject can be modelled not only as the understanding a data processor has over the attribute, but also their understanding of the knowledge states of other processors regarding the attribute.

Table 6.1 articulates the specific awareness transformations, which in turn characterise the evolution of a data processors knowledge. Taking anti-pattern 1 as an example. If a data subject were to reject data processing on the first party, the data processor stands to acquire an understanding of the data subject attributes, the unawareness of the third party along with the subjects understanding of the unawareness state of the third party. Formally, this is captured as the following assertions:

$$A_F su.f \tag{7.1}$$

$$A_F \neg A_T su.f \tag{7.2}$$

$$A_F A_{su} \neg A_T su.f \tag{7.3}$$

The data processor learns of the data subject su attributes f as such information exchange is necessary for the business function of the first party F, (item 6.1). The unawareness of the third party data processor T is a consequence of su denying consent to data processing activity involving T (item 6.2). As this preference is indicated on consent modules provided by F, they in turn are made aware of this requirement. Further, the awareness state of su concerning T becomes tenable for F (item 6.3). Additionally, if su were to accept data processing directly on T, then the following awareness transforms occur for T.

$$A_T su.f \tag{7.4}$$

The first party and third party data processors account for 4 awareness transforms in this interaction context (items 6.1-4). The awareness transforms aim to provide analysts with a more holistic understanding of the information data processors stand to acquire when interacting with data subjects in the different anti-pattern contexts. A second anti-pattern (AP2) from table 7.1 has a prevalence level of 3. This anti-pattern involves a data subject interacting with two distinct first party data processors interconnected by a common third party. In the event the data subject su accepts data processing on the first party F, the following awareness transforms take place:

$$A_F su.f \tag{7.5}$$

$$A_F A_T su.f \tag{7.6}$$

$$A_F A_{su} A_T su.f \tag{7.7}$$

The resulting awareness transforms for F are similar in structure to the transforms detailed in items 6.1-3. This time, the awareness transforms detail the understanding F acquires over the resulting awareness T possesses over the data attributes of the subject. The transform in 6.5 occurs as a result of the data exchange between su and F. Transform 6.6 occurs as a result of the acceptance preference being enacted explicitly through the consent mechanisms provided by F. Finally transform 6.7 occurs as a result of the first party being able to infer the awareness state of the data subject owed to the explicit interactions with F. Likewise, when the data subject rejects processing on F' the following awareness transformations occur:

$$A_{F'}su.f \tag{7.8}$$

$$A_{F'}\neg A_T su.f \tag{7.9}$$

$$A_{F'}A_{su}\neg A_T su.f \tag{7.10}$$

These transforms are the result of the explicit interactions had between the privacy consent modules offered by F' and su. Again the awareness of F' occurs because of necessity for business function (item 6.8). The unawareness of T is something F' can reason over as the consent modules used to reject consent are made available by F' (item 6.9). Finally, F' is able to reason about the understanding the subject owed to the consequences of the interaction su has with their consent modules (item 6.10). Finally, both of these actions will have the following awareness transformations on the third party T:

$$A_T su.f \tag{7.11}$$

$$\neg A_T su.f \tag{7.12}$$

In table 7.1 anti-pattern 1 and anti-pattern 2 have the same levels of prevalence in the information flow sub-network (i.e. the connections to inputmag.com) but examination of the awareness transforms allows an analyst to elicit distinct levels of information transparency intrinsically tied to the understanding of the first and third party data processors. Anti-pattern 1 provides us with 4 awareness transforms increasing information transparency to the data processors, whilst anti-pattern 2 provides 6 awareness transforms.

We exclude awareness transformations involving the data subject or the tenability of unawareness in this analysis. This is because when looking to quantify the reduced transparency benefiting a data subject during anti-pattern refactoring, the level of understanding tied to data processors will indicate the level of attainable secrecy, as this is inversely proportional to the levels of transparency withheld from data processors.

Refactoring a third party facilitating a given anti-pattern will mean the third party is no longer involved in the data processing activities of the first party. This will lead to a reduction of information transparency levels as represented by awareness transforms no longer occurring. This benefits the data subject as the reduction in awareness transforms owed to the omission of data processing activity reduces the potential for misaligned privacy expectations. Quantifying the benefit to the data subject can be achieved by considering the degree of removed information transparency from data processors. By examining the awareness transforms associated with a privacy anti-pattern, in addition to the prevalence of such an anti-pattern,

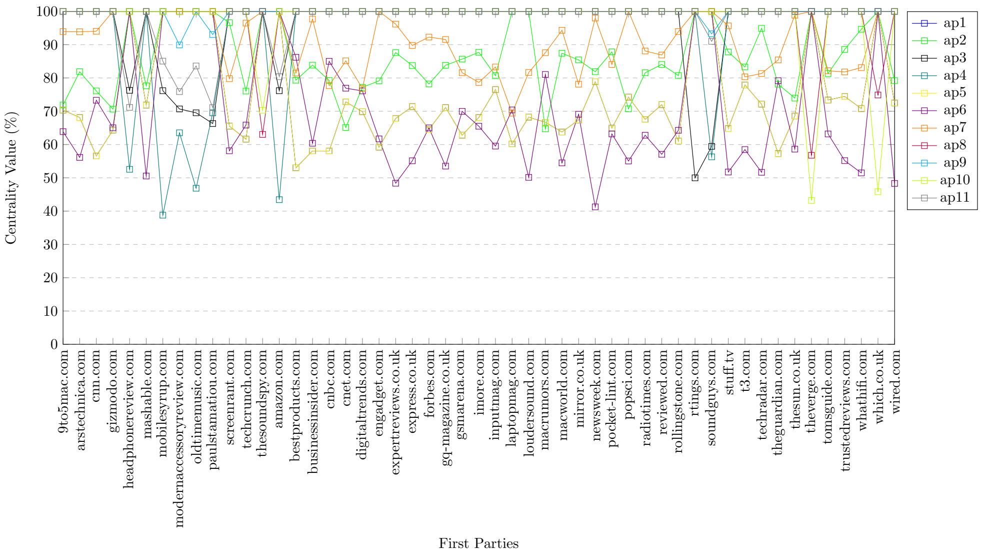


Figure 7.1: Anti-pattern centrality values for all first parties

an analyst can quantify the utility of a refactoring action for a data subject by considering the number of distinct awareness constructs no longer tenable in the memory stores of data processors.

In the event an analyst wished to leverage on this information in order to make a determination on how to select an anti-pattern to enact refactoring, they have the choice of leveraging on the centrality values or the awareness values associated with the anti-pattern. In the event an analyst wished to refactor anti patterns based on their awareness on the information flow model we can conclude anti-pattern 6 will be the target for refactoring with the largest observable awareness value of 24. This appears to be the best outcome for the data subject as it reflects the largest degree of transparency removed when the anti-pattern is refactored. Conversely, if the analyst wished to base their refactoring decision on the total centrality loss, then anti-pattern 7 will be the recommended outcome, as this yields the lowest centrality loss and as such, represents the best possible outcome for the first party.

Removing the anti-pattern with the highest level of awareness also leads to the highest level of centrality loss. Likewise, selecting an anti-pattern impacting the centrality values the least does not in turn remove the anti-pattern exhibiting the largest awareness levels. This suggests a tension between the two metrics. To confirm an inverse relationship, we analysed the observable centrality values for the anti-patterns across all first parties. Figures 7.1 and 7.2 illustrate a breakdown of the centrality and prevalence metrics respectively.

The columns in both figures represent the values associated with the first parties on the crawl. On figure 7.1 each series illustrates the associated centrality percentile observable with the anti-pattern. This indicates the remaining percentage from the original centrality values when the third parties facilitating the anti-pattern are removed. Generally, the trends associated with the centrality loss in this illustration correlate with the patterns observable in table

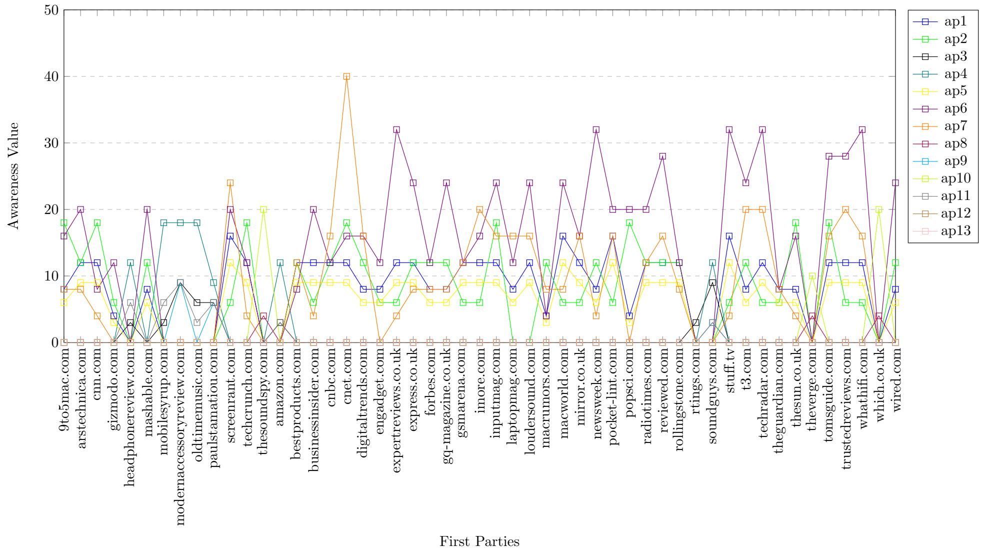


Figure 7.2: Anti-pattern awareness values for all first parties

7.1. Specifically the first party ’inputmag.com’ yields a refactoring centrality value for anti-pattern 7 of 83% whilst the average for the dataset is 74%. Conversely, the worst centrality value is associated with anti-pattern 6 with a value of 58% whilst the dataset average for anti-pattern 6 is 71%. The column values in figure 7.2 indicate the total transparency/awareness value for third parties facilitating the anti-patterns. Here anti-pattern 6 appears to on average yield larger awareness values of 15.

Generally, anti-pattern 6 yields the larger awareness values in the dataset but also yields the largest percentile loss to a first party. Pearson testing confirms a significant inverse relation, meaning efforts to provide utility to the first party will in turn negatively affect the utility for the data subject. This tension is not problematic in the event the analyst wishes to investigate optimal refactoring strategies benefiting either the data subject or the data processor exclusively. However, it is intuitive to consider analysts will want to investigate such refactoring options whilst concurrently safeguarding the business function of the impacted first party and the benefits to the data subject. It is not immediately obvious what the optimal choice is for an analyst trying to balance these two competing objectives. To investigate a solution to this challenge, we leverage on economic theory to represent our decision making problem as a multi objective problem.

7.3 Multi Objective Analysis

Various real world problems involve investigating optimal approaches to satisfying multiple competing objectives. Such scenarios are often referred to as multi objective optimisation problems. Typically, the outcome of a multi objective optimisation problem will not simultaneously satisfy all given objectives, but rather identify the best trade-offs given competing objectives. These optimal trade-offs are commonly referred as the Pareto-optimal set, Pareto set/front or feasible region [97] and are typically required when a single solution is not identifiable [98].

Instead, a set of Pareto efficient solutions are returned and in most cases, a decision maker (typically a domain expert) selects the most preferred option. A possible solution is considered Pareto efficient if there exists no other solution that increases the satisfaction of one objective without decreasing the satisfaction of the other [99]. In multi objective settings there is typically a fitness function determining the score, or ”fitness” of a possible solution with respect to one or more objectives.

In our problem scenario there is an inherit tension between the awareness levels associated with a privacy anti-pattern and the centrality values associated with the third parties facilitating the anti-pattern. When looking to review refactoring options subject to these conflicting requirements, the need for a stakeholder to make their decision subject to a series of trade-offs makes this well suited as an optimisation problem.

A multi objective problem concerns the reduction of a solution space Π where each element $x \in \Pi$ represents one possible solution. In our scenario, a solution space is populated with privacy anti-patterns representing possible refactoring options an analyst has with a software model. A given anti-pattern (solution) will provide values that can be mapped to an objective space concerning both awareness levels and centrality loss.

A multi objective problem is characterised by a series of objectives representing the inherit tensions constraining the solution space. The assumed default objective of an analyst will be the maximisation of centrality (after refactoring) to a first party and the maximisation of awareness loss for the end user. We represent such an objective o formally as a tuple (γ, m) with γ representing MIN/MAX function, m indicates the measurable property to apply γ to.

Identifying the Pareto set for a given first party involves scoring the different anti-patterns applicable to a first party forming Π with respect to their prescribed awareness levels and the centrality values. To elicit a collection of optimal solutions given two competing objectives o_1 and o_2 we leverage on a modification of the next generation non dominated sorting algorithm (NGSA-II) [100]. This procedure allows us to elicit a set of Pareto efficient solutions F from an unsorted set of solutions Π, given a pair of objectives o_1 and o_2.

The approach compares each solution within the solution space Π, to determine whether a solution dominates another (i.e. $\alpha \prec \alpha'$). In the event a solution is dominated (i.e. performs worse on both objectives), we will not add it to the Pareto front F. The application of Pareto efficiency allows analysts to acquire a Pareto front F, where each solution within F is itself considered Pareto efficient. This allows analysts to rule out sub-optimal anti-patterns when considering how to best refactor consent mechanisms given the competing objectives.

Algorithm 7: Calculation of Pareto Frontier

```
Set of unsorted solutions $\Pi$,
Objective space $O = \{o_1, o_2\}$ Pareto efficient set $F$
foreach $\alpha \in \Pi$ do
    $n_\alpha = 0$
    foreach $\alpha' \in \Pi$ do
        if $\alpha' \prec \alpha$ then
            $n_\alpha = n_\alpha + 1$
    if $n_\alpha = 0$ then
        $F = F \cup \{\alpha\}$
return $F$
```

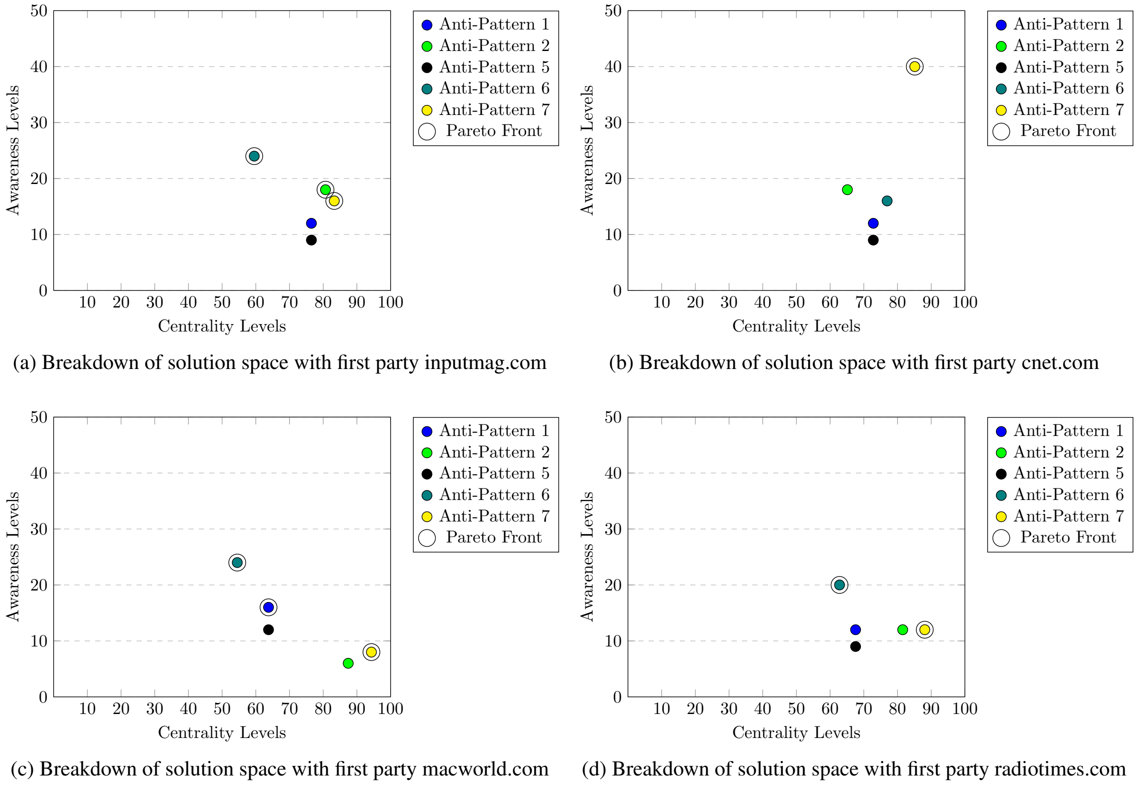
Figure 7.3 illustrates a series of multi objective models for the top 4 first parties with the highest risk values per figure 6.14. Each sub-figure depicts the solution space composed of a series of anti-patterns, along with the identified feasible region. For all four examples the solution space is composed of anti-patterns 1, 2, 5, 6 and 7. Each possible solution within the solution space is scored on an objective space, where $o1 =$ maximise centrality and $o2 =$ maximise awareness. For all solution spaces, those anti-patterns providing larger values for both $o1$ and $o2$ are objectively better candidates for selection.

The objective is to leverage on procedure 7 to reduce the solution space to a single collection of Pareto efficient options. This allows analysts to spearhead their decision making in the event bias information towards either objective is not known or is not intended to be provided.

Sub-figure a) presents the solution space for the first party 'inputmag.com' with the Pareto optimal membership indicated with an enclosing circle. Anti-patterns 2, 6 and 7 form the Pareto front whilst anti-patterns 5 and 1 are dominated and thusly not recommended, reducing the solution space by 40%. Sub-figure b) illustrates the output from the first party 'cnet.com'. This time, we see the Pareto front only contains anti-pattern 7, reducing the solution space by 80%. Sub-figure c) illustrates the output from the first party 'macworld.com' where the elicited Pareto front is again three elements. Anti-patterns 2 and 5 are not considered feasible solutions with anti-pattern 6, 5 and 7 forming the Pareto set. Finally sub-figure d) illustrates the outcome for the first party 'radiotimes.com'. The solution space is the same but the Pareto front is only two elements in size, with anti-patterns 7 and 6 being recommended to the analyst.

An interesting observation is each set of Pareto optimal anti-patterns is unique for each illustrated problem. This suggests each problem can be uniquely characterised and the amount of effort required by an analyst to effectively analyse multiple first and third parties may be significant in the event a non trivial analysis is required. The multi objective method can help mitigate the complexity of reducing the solution space of available anti-patterns.

Figure 7.4 illustrates the elicited Pareto fronts for each first party across all analysed first



(a) Breakdown of solution space with first party inputmag.com

(b) Breakdown of solution space with first party cnet.com

(c) Breakdown of solution space with first party macworld.com

(d) Breakdown of solution space with first party radiotimes.com

Figure 7.3: Average scores for conflicting privacy assertions (experiment class A)

party data processors. The objective of this analysis is to provide information to an analyst at a glance, of what anti-patterns are recommended refactoring options. Each column represents the options for a given first party, whilst the stacked bars illustrate the breakdown of the different anti-patterns composing the Pareto front for the associated first party. The number of recommended options differs across the dataset. Anti-patterns 7 and 2 are the most commonly recommended refactoring options, appearing in 35 and 34 of the returned Pareto sets, comprising 69% and 67% of the analysed first parties respectively. Conversely, this data also provides us with information on the anti-patterns not commonly recommended as refactoring options. Anti-pattern 8 existed within the Pareto efficient solutions only twice, accounting for 5% of the first parties, whilst anti-pattern 9 is a recommended outcome in 3 of the 51 first party Pareto sets at 6%. An important consideration when contemplating the frequency of recommendations in figure 7.4 is the ratio between an ani-pattern occurring on the solution space and being a member of the Pareto efficient collection.

Table 7.2 presents a breakdown of this data. Anti-patterns 7 and 2 despite being the most recommended in the multi objective analysis, are not the most prominent anti-patterns detectable on the dataset. Anti-patterns 1, 5 and 6 are the most common anti-patterns composing solution spaces at 40 instances, whilst their inclusion ratios are not as high. Anti-pattern 1 in particular, despite being one of the most common anti-patterns only appears in the Pareto set 4 times. Anti-pattern 5 is not a member of any Pareto set whilst anti-pattern 6

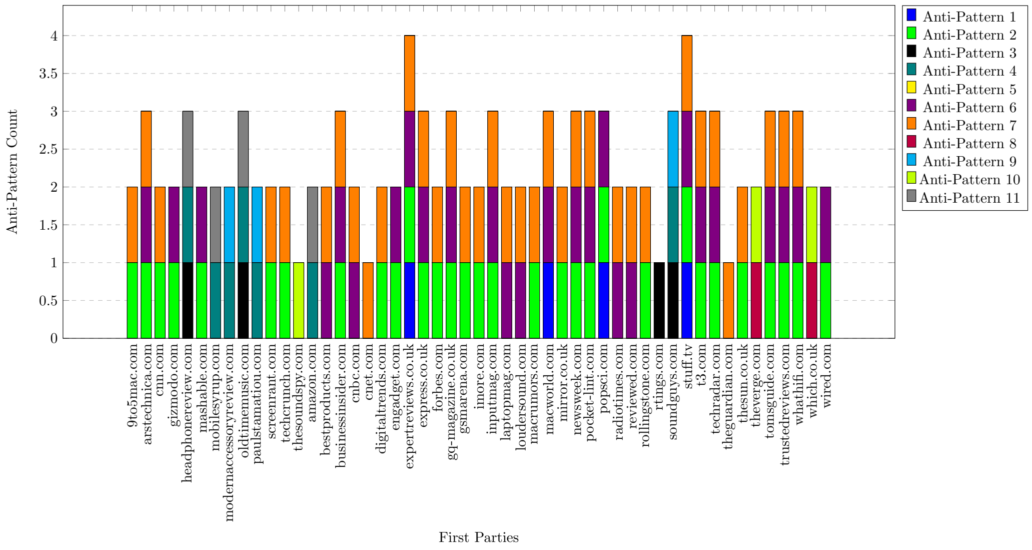


Figure 7.4: Illustration of the Pareto fronts elicited for the first parties

Table 7.2: Illustration of the ratio between anti-pattern inclusion on the solution space vs inclusion in the Pareto front

	Anti Pattern												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Solution Space Count	40	38	8	7	40	40	35	3	3	3	7	0	0
Pareto Front Count	4	31	4	7	0	26	35	2	3	3	4	0	0

appears in 26 Pareto fronts. Conversely, anti-patterns 4, 7, 9 and 10 all have a selection rate of 100% however their instance rates are much lower overall, with the exception of anti-pattern 7 having a maximum inclusion rate and high instance rate of 35. We can conclude that anti-patterns from table 7.2 possessing a high inclusion rate on the solution space, but a low inclusion rate for Pareto fronts generally perform worse in terms of the benefit they offer both data subjects and data processors. Likewise, those anti-patterns with high solution space/Pareto font inclusion ratios generally perform better for either the data subject or the processor.

This data provides the analyst with granular insight into which anti-patterns are Pareto efficient refactoring recommendations for any given first party. It further provides insight on the relative performance of each anti-pattern by quantifying the ratio of its inclusion as a Pareto efficient solution across the entire dataset. In turn this allows analysts to reduce the solution space of decisions at scale and provide rationale for a selection based on its ratio of prevalence in Pareto fronts. In the event an analyst does not wish to consider biases towards the needs of a data subject or the needs of a data processor when looking to make recommendations, they can leverage off the data from figure 7.4 and table 7.2 to justify their decision making. However, the analyst will be required to act as the final decision maker.

These data points serve as a starting point for analysts wishing to provide bias values to the objectives, or investigate the outcome of different bias skews. The challenge with each first party multi objective model being uniquely characterisable, is the manner by which the Pareto front forms in procedure 7 is agnostic of the objective weightings. This means any anti-pattern refactoring recommendation using these datasets will not be sufficiently informed in the event the analyst wishes to investigate objective biases as the method does not natively reason about objective weights. For instance, the Pareto efficient set for inputmag as follows: $F = \{AP_2, AP_6, AP_7\}$. Anti-pattern 7 yields the largest Pareto prevalence score of 100% with comparable solution space inclusion rates to anti-patterns 2 and 6. Selecting anti-pattern 7 as the refactoring strategy yields the highest degree of utility for the data processor as visualised in figure 7.3 a). Similar outcomes are observable on the remaining problems with the exception b). In the event the Pareto set has a single member, this element represents the objectively best outcome for both the data subject and processor irrespective of objective biases. However, this outcome is only observed with 4 out of the 51 analysed first parties. Otherwise, additional analysis is required on the returned Pareto front to better recommend a single outcome when biases are provided by the analyst.

7.4 Utility Analysis

We leverage on the concept of utility functions to perform a posteriori analysis on the returned Pareto front. Utility is often conceptualised as a function operating on weighted parameters to arrive at some output to a given problem. Returning to the objective functions in 6.3 we can see an objective o is a tuple (γ, m) characterising the solution space of the multi objective problem. We can expand on this definition by introducing an additional tuple element w to indicate an optional weight parameter representing the subjective bias an analyst wishes to enact on the objective space. This allows an analyst to scalarise the two objective values as a metric to score a given anti-pattern within the solution space. In the event bias information is not supplied, equivalent default values can be used such that for a given objective space $O = \{o_1, o_n\}$ condition 7.13 holds:

$$\sum_{i=1}^{n} o_i(w) = 1 \tag{7.13}$$

Where $n = |O|$. This simplifies the reporting of anti-pattern suitability by reducing the problem domain to a single property. This will streamline the selection process for analysts as they are no longer required to observe a dashboard plotting the competing objective space, rendering the approach more suitable for workflow integration. The formula we use to calculate the utility of an anti-pattern within a solution space is provided in formula 7.14:

$$u_{AP} = \lambda_1(AP).w_{\lambda_1} + \lambda_2(AP).w_{\lambda_2} \tag{7.14}$$

Where AP is the anti-pattern solution element from within the solution space. The function λ_i returns the value associated with the metric comprising the objective space. The weight parameter w_{λ_i} represents the specific level of bias to be observed on the metric when outputting the utility score for the anti-pattern u_{AP}.

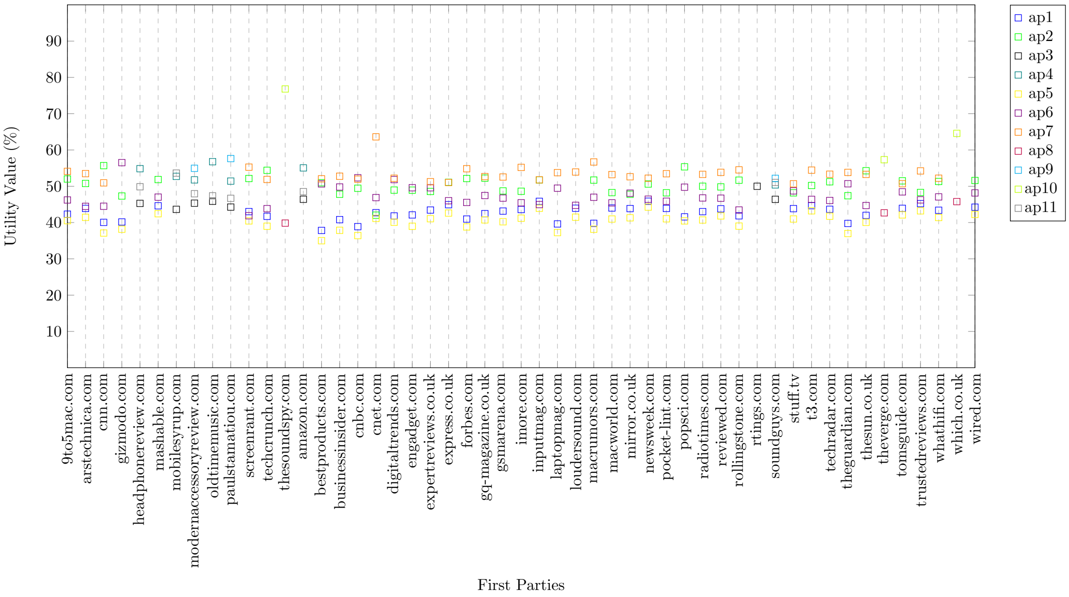


Figure 7.5: Breakdown of refactoring utility values for different anti-patterns with neutral objective biases.

Figure 7.5 illustrates a breakdown of observable utility values for first parties in the event no bias information is presented. For completeness, we illustrate all anti-patterns forming the solution space of a first party. For soundness, we verify each recommended outcome (i.e. those anti-patterns with the highest level of utility) being member elements of the associated Pareto set for the first party.

Identifying an optimal refactoring outcome subject to objective biases involves selecting an anti-pattern yielding the highest utility value. Each column in this figure represents one of the first parties and the associated scatter plot illustrates the utility value of each anti-pattern. We can see generally, anti-pattern 7 is the solution yielding the higher overall utility for 55% of all first parties. Anti-pattern 2 is generally the second highest recommended solution for first parties, whilst anti-pattern 5 is not recommended.

What is of interest are observable trends when stepping over bias values for the two objective functions. Cross referencing the output of the utility approach across all bias skews with the Pareto efficient set of solutions allows us to evaluate whether the recommendations are sound for all bias values. Figure 7.6 illustrates the recommendations associated with varying

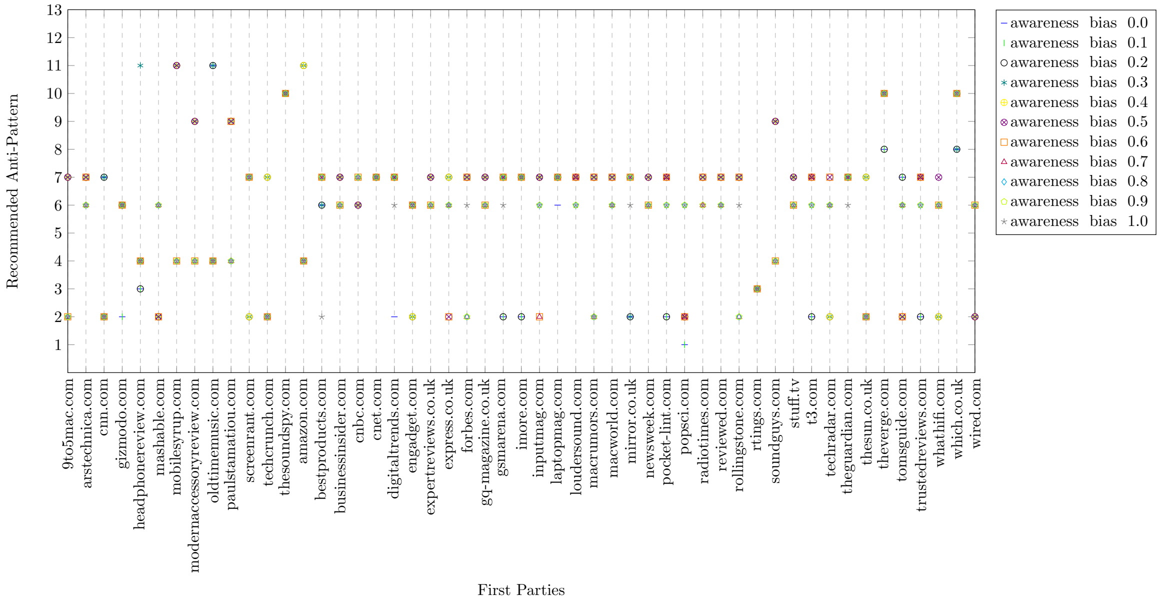


Figure 7.6: Illustrating the recommended anti-patterns for different awareness bias parameters.

bias parameters. The illustrated bias values are associated with the awareness metric. The associated bias value towards the competing metric centrality will be implied as the inverse value of the exhibited bias parameter. For instance, the plots associated with awareness bias value 1.0 indicate the recommended anti-patterns benefiting the data subject the most. The associated bias value to the competing centrality metric is implied to be the inverse of the awareness parameter. In this case the inverse (centrality) value $(1 - x)$ where x is the awareness value will be 0. Conversely, if the illustrated bias parameter were 0.4, the associated implied bias parameter for centrality will be 0.6.

Each column in figure 7.6 represents a first party from the dataset. Each step on the y axis corresponds to a specific anti-pattern, therefore each visualised row serves as an indication of the frequency by which the anti-pattern is recommended. Each delineated series corresponds to a specific bias configuration. We can see anti-pattern 7 is the most commonly recommended anti-pattern with 35 out of the 51 first parties providing it as having the highest level of utility across bias skews. Anti-pattern 6 is the second most common followed by anti-pattern 2.

An interesting observation concerns the recommended outputs when bias values are extreme. In such instances, sub-optimal anti-patterns that do not fall within the Pareto set are recommended. These are illustrated in table 7.3 and are explainable as follows. To re-iterate, the set of Pareto efficient solutions illustrates the collection of optimal outcomes where no solution within the collection F can be dominated by another.

This means for a competing objective space, If a feasible region can be identified with at least two member elements, there is a guarantee one element from the feasible region will

Table 7.3: Illustrating the erroneous output that does not fall within the Pareto efficient set

First Party	bestproducts		digitaltrends		forbes		gsmarena		mirror		rollingstone		theguardian	
Anti-Pattern option	2	7	6	7	6	2	6	7	6	7	6	2	6	7
Awareness Value	12	12	16	16	12	12	12	12	16	16	12	12	8	8
Centrality Value	79.3%	81.5%	76.1%	76.8%	65%	78.2%	70%	81.6%	69.1%	78.1%	64.3%	80.7%	79.2%	85.4%

yield the maximal value for one metric, whilst a second element will yield the maximal value for the opposing metric.

The bias value when configured as 1 will mean the reported utility for a given anti-pattern will be impacted by the yielded metric value for awareness with no consideration for the centrality metric. In such an instance, given a collection S of elements, if each element $s_i \in S$ can yield the maximal observed value for awareness, irrespective of the associated centrality scores, the elements will be returned from the analysis as optimal outcomes. This means the reporting can be erroneous, such that an identified outcome may be superseded by an objectively superior refactoring outcome yielding a better outcome for the data processor (owed to better centrality values). This is occurring with the recommendations in table 7.3. Each column illustrates the two options for a given first party, the shaded elements illustrate the anti-pattern recommended from the utility function and in turn a breakdown of the yielded metric values are provided for comparison. Both anti-patterns for each first party yield the same values for the awareness metric. Therefore with a maximal bias value these solutions are indistinguishable. The centrality values in turn confirm the recommended anti-patterns are not Pareto efficient as the alternative anti-pattern in each instance yields a better value, making it the Pareto efficient solution.

Interestingly this phenomenon only occurs with extreme bias values favouring the data subject. Lower bounded values favouring the data processor did not yield erroneous outcomes. The sub-optimal recommendation problem can be resolved if the utility function were constrained to operate on the resulting Pareto set of solutions, in favour of the solution space. Again we illustrate the entire solution space for completeness and demonstrate the soundness of the recommendations by virtue of encapsulation within the Pareto efficient set, with the contradictions per table 7.3 explained. If multiple recommendations are made when the utility function is constrained to the feasible region, the anti-patterns in the recommendation will be indistinguishable for both objective values.

Figure 7.7 depicts the nature of the recommendations across the bias spectrum. The columns in figure 7.7 represent the different anti-patterns within the Pareto optimal set. Each data point across the y-axis serves to indicate the awareness bias value the associated anti-pattern was recommended. For instance, with inputmag.com we can see there is a data point associated with anti-pattern 7 on awareness bias 0.5, which tells us the bias steps from 0.0 (least possible bias value) through to 0.5. The utility formula in 7.14 recommends anti-pattern 7 as the output for this specific first party. However, the next bias stepping at 0.7 will not

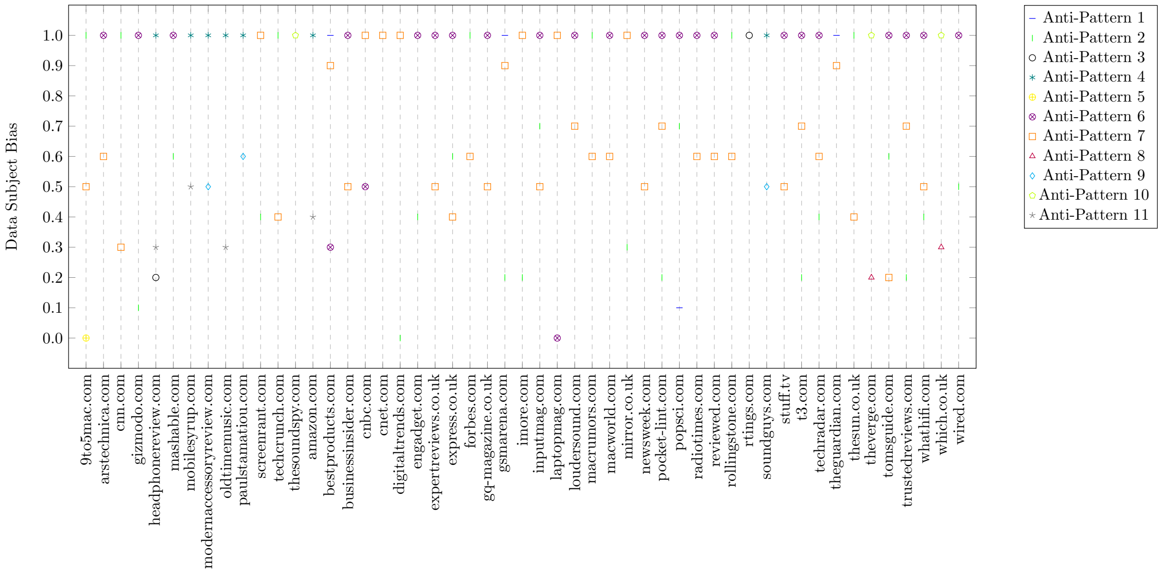


Figure 7.7: Illustration of the bias thresholds for anti-pattern recommendations.

be associated with anti-pattern 7 but instead anti-pattern 2 which has a data point at bias level 0.7. This indicates the bias range of 0.5-0.7 (inclusive) will recommend anti-pattern 2 as the solution for refactoring. Finally, we have a data point with the maximum bias value for awareness (1.0), the data point here indicates anti-pattern 6 will be recommended at this maximum bias value through to bias value 0.8.

Between anti-patterns 6 and 7 being the most commonly recommended for refactoring, the two are often recommended under similar bias contexts. Anti-pattern 6 for instance was recommended as a refactoring option on the Pareto front 9 times in the event bias levels leaned towards preferring the data processor (i.e. the bias values were lower than 0.5), whereas they were recommended 23 times when the bias values preferred the data subject and once if neutral. Conversely, we can see anti-pattern 7 is recommended as optimal output 5 times when the bias values favour the data processor, 22 times when the bias favours the data subject and 8 times when neutral. This is a stark contrast with anti-patterns 11 and 3. We can see in all cases, anti-patterns 3 and 11 are recommended as output when the bias weightings favour the data processor with one exception involving neutral weightings and another involving subject biases. Understanding which anti-patterns lend themselves better to providing utility for the data subject or the data processor will provide analysts with insight they can use to calibrate their decision making. We can see generally, an anti-pattern can be recommended as the optimal output at different levels of bias, which emphasises the utility of a given solution is often specific to the problem being modelled. This in turn demonstrates the importance of a computational approach to help automate and scale the analysis for analysts in order to assist them manage the complexity of the task.

7.5 Evaluation

The output of the utility function will be an anti-pattern to refactor for the purpose of reducing observable privacy risk. The evaluation aims to investigate the efficacy of refactoring recommendations from the utility function by comparing the observable deltas to privacy risk values attained through enacting single metric selection strategies. The closer the observable delta values, the greater levels of privacy risk the anti-pattern will reduce. The objective is to investigate the distribution of bias values for the utility function to determine which recommendations associated with different bias levels stand to attain greater reductions in privacy risk when refactored. We focus on a single refactoring activity. This is firstly because of the combinatorial explosion observable when unpacking all possible refactoring combinations and secondly due to assumed resource limitations of data processors making a single refactoring solution more favourable to their business processes.

The evaluation was conducted in three steps. Firstly, we look to measure the observable deltas from the selection strategy where the prevalence levels of an anti-pattern are used as the decision criterion. Any refactoring activity occurring on a first party will have an impact on the privacy risk associated with the first party. Therefore, a naive approach to conduct refactoring is to refactor the most prevalent anti-patterns, reducing the privacy risk by a maximum amount afforded by a single refactoring option. Privacy risk values associated with the prevalence selection strategy can serve as a benchmark to gauge the effectiveness of the utility function recommendations. We wish to investigate the recommendations from the utility function which provide the closest observable privacy risk scores to the benchmark. Each recommendation from the privacy utility function will in addition to privacy risk, provide some benefit to the data subject and incur a cost to the data processor. The second step of the evaluation is to therefore investigate the inverse relationship by plotting the relative centrality and awareness values associated with an outcome elicited from a bias value. This allows us to investigate whether there is an optimal bias configuration. Given the multi objective problem is characterised by two metrics, a trivial solution to the selection problem involves selecting an anti-pattern yielding maximum benefit to the end user or data processor. Therefore, the third step in this evaluation involves cross referencing the average privacy risk values from the utility strategy against the average values observed from the single selection criteria. The final step involves a time complexity analysis of the approach.

In step 1, for each first party we first plot the privacy risk values observed with the outcome from the prevalence strategy. Then, we step over the bias values and for each first party, plot the observable privacy risk scores when the different anti-pattern recommendations for each bias values are refactored.

Figure 7.8 illustrates a breakdown of the observable privacy risk scores for each first party from the dataset. Each column illustrates the different bias skews for a given first party. The

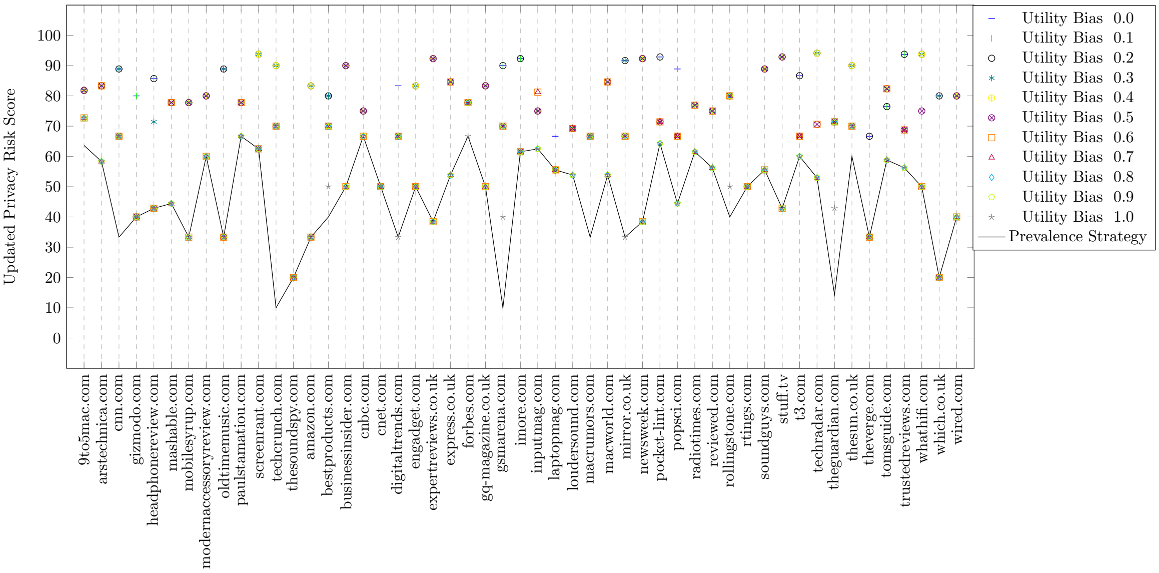


Figure 7.8: Illustrating the updated privacy risk metrics when refactoring first parties based on the recommendations from the bias values.

Table 7.4: Pearson correlation output between bias parameter and privacy risk

r coefficient	-0.524
Sample size (n)	561
t-value	-14.551
p-value	3.278E-41

black line on this diagram represents the observed privacy risk values associated with the first party in the event the prevalence strategy is enacted i.e. an anti-pattern is refactored based solely on the fact it is most prevalent. The objective is to articulate the different bias values to consider which of these provides greater levels of utility in their recommendations.

Overall, we note an average global privacy risk score of 65% across all bias skews with the utility strategy. Conversely, an average privacy risk score of 45% is observable with the prevalence strategy. When we observe low levels of bias for the awareness metric this generally leads to higher levels of privacy risk compared to stronger levels of bias. The average privacy risk score associated with bias levels below 0.5 are 76% whilst the average risk score associated with bias levels above 0.5 are 54%. Table 7.4 illustrates the outcome of Pearson testing which illustrates a negative correlation between the bias values and the average observed privacy risk. This tells us outcomes from bias parameters favouring the data processor do not reduce the privacy risk values as favourably as the alternative outcomes.

The next step in the analysis concerns investigating the relation between the provided centrality and awareness values for those refactoring recommendations across the bias parameters for the utility function. A recommendation from the utility function to reduce privacy risk in addition to providing acceptable trade-offs for stakeholders will be a better outcome for analysts wishing to balance the needs of both stakeholders in their recommendations. Figure 7.9 illustrates the average metric scores for awareness, centrality and privacy risk.

Each column represents the bias parameters for the utility function and each plot represents the associated average value for awareness (blue), centrality (black) and risk (red). Each awareness percentile value is taken against the maximum observed single awareness value associated with the facilitated anti-patterns on a first party. A value of 100% indicates the outcome of the utility function will remove the largest observed levels of transparency in a single refactoring action. The average centrality percentiles are taken against the total centrality values of third parties facilitating anti-patterns. The average privacy risk values are calculated based on remaining proportional third parties exhibiting anti-patterns. Given the analysis only considers a single refactoring action for a first party, the objective is to plot the inverse relation between the two metrics to consider the cost incurred for the data processor as the suitability of a refactoring outcome increases for the data subject.

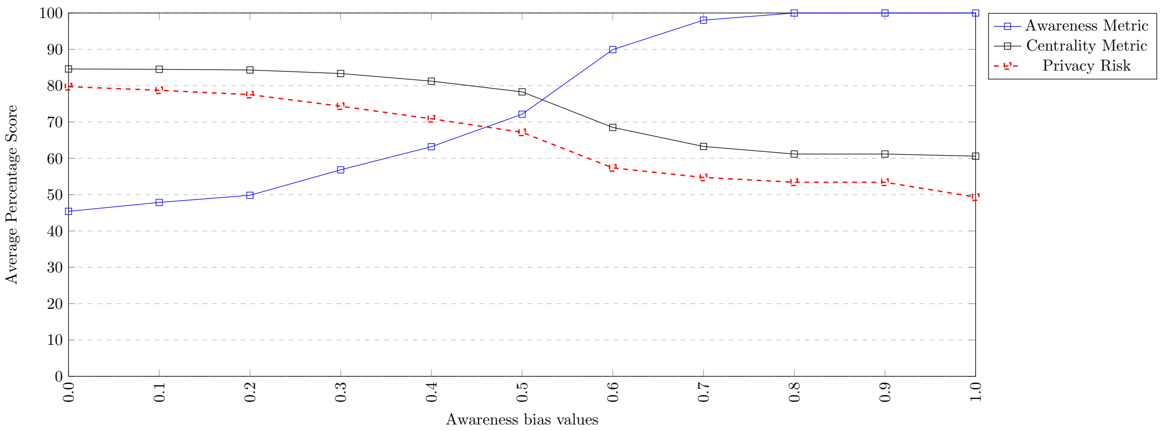


Figure 7.9: Illustrating the average trends between the awareness and centrality values against privacy risk

A near linear relationship is observable between the values of the metrics and the bias configurations. For the awareness metric specifically, we yield the lowest average value of 45% when the bias parameter is 0.0. This value configures the utility function to provide refactoring recommendations based solely on the impact on the data processor. Unsurprisingly, we observe the highest centrality score of 85% for the data processor with this configuration. As we approach the neutral configuration, we see the awareness value for the data subject increases from 45% to 63%. In contrast, the values for the data processor decrease from 85% to 81%. The delta for the data subject combined with the delta for the data processor indicates the default configuration, relative to the extreme bias parameter provides data subjects with a considerable degree of utility in exchange for what appears to be a moderate centrality cost to the first parties. As we continue from the neutral configuration to a configuration preferring the data subject, we see the awareness values unsurprisingly change from 72% to 100% whilst the centrality values change from 78% to 61%.

The average privacy risk scores plotted here serve as an indication on how well an outcome performs when attempting to reduce privacy risk. The trend observable on figure 7.9 for risk is similar to centrality. We observe an average risk score of 79% when the bias parameter strongly favours the data processor. The neutral configuration yields an average risk value of 67% whilst the minimum recorded value of 49% is associated with the subject bias parameter.

The insight provided by the three plots in figure 7.9 allows an analyst to visually consider the average impact of the possible state-space of solutions. We quantify the utility of a given outcome wrt three data points in formula 7.15:

$$v = ((a * 0.5) + (c * 0.5)) - o \tag{7.15}$$

Where a and c are associated values for awareness and centrality respectively. The privacy risk value o acts as a discount metric to the products of a and c. A theoretically optimal outcome is the observation of maximum values for awareness and centrality, with a discount of 0 indicating no observable privacy risk. The higher the score, the better outcome observed. v in turn allows analysts to streamline their analysis in determining where the desirable tradeoffs are on a solution space.

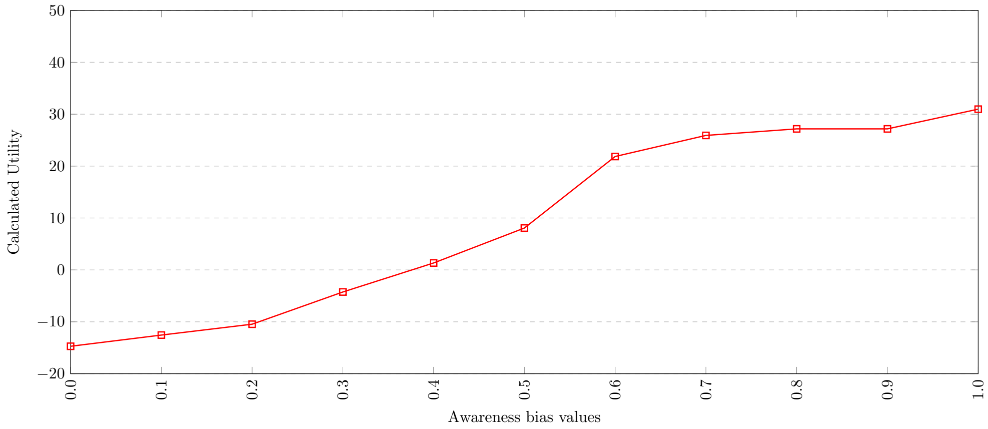


Figure 7.10: Illustrating the applied utility value across multiple objective spaces.

The results of applying formula 7.15 to figure 7.9 are illustrated in figure 7.10. Here, we see a single plot illustrating the calculated utility values for each of the bias parameters. With the bias parameter favouring the data subject we observe low utility values, which is largely owed to the high privacy risk values and the low awareness values. We observe the greatest increase in the calculated utility from 8 to 21 when stepping from the neutral bias configuration to value 0.6.

The outcome of step three per figure 7.11 illustrates a comparison between average privacy risk scores associated with the utility selection strategy and the three single decision crite-

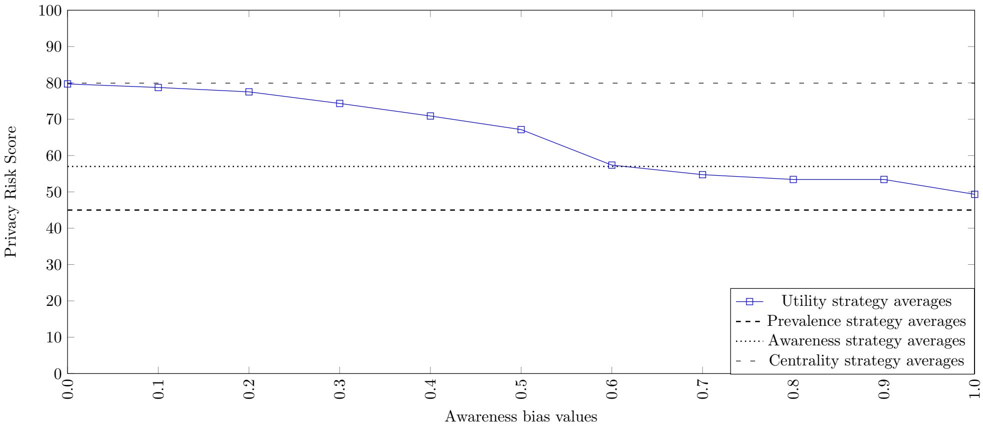


Figure 7.11: Illustrating the average privacy risk across the bias spectrum.

rion strategies. We can see a privacy risk value of 80% being associated with the selection strategy to maximise centrality for a data processor. This represents a worst case scenario comparatively to the other selection strategies. We observe an average risk value of 57% for the selection strategy to maximise awareness for the data subject (which translates into the reduction of information transparency). Finally, we note an average privacy risk reduction to 45% when enacting the outcomes from the prevalence selection strategy. An important concern to note here is the prevalence strategy represents the best possible outcome as the strategy refactors anti-patterns based solely on their frequency which will have the highest degree of impact. Selecting refactoring options based on either maximising the benefit to the data subject or processor in turn serves as a benchmark for the values associated with the utility function.

Starting with the bias value of 0, the utility approach yields a comparatively high privacy risk value. The observable deltas between the centrality selection strategy are negligible, with a 34% delta from the awareness selection strategy and a 46% delta from the best case. When observing the neutral bias configuration the privacy risk of the utility approach now sits below the worst case solution but does not yet sit below the awareness or prevalence strategies. Observed deltas are 12% from the worst case, 10% and 22% from the awareness strategy and best case respectively. Moving to the bias parameter of 1 we drop below the awareness strategy but continue to sit above the best case, with the observable deltas of 50% from the worst case, 8% from the awareness strategy and 4% from the best case.

This is an interesting observation as it appears to contradict the initial conclusions from figures 7.9 which suggested a neutral bias parameter would be suitable to elicit favourable recommendations. It is not until we step to bias parameter 0.6 do we observe the utility outcome match the awareness selection strategy in privacy risk. The data presented in figure 7.11 suggests mild bias values in favour of the data subject will be able to match or better the average privacy risk observable from the awareness selection strategy. Looking at the recommendations associated with bias value 0.6 the observed privacy risk of 56% is only 7% off the observed maximum reduction of 49% but is almost able to half the observable privacy risk of a first party in a single refactoring action. Under the assumption an analyst would wish to remain as neutral as possible in their refactoring recommendations when investigating the outcomes of bias spectrum, the parameter of 0.6 appears to be a favourable choice.

The benefit of the single decision strategies are their ease of implementation and simplification of the problem, however their drawback concerns the inability for analysts to apply nuance in their decision making. Basing refactoring on the awareness metric for example, will be equivalent in the utility strategy to exhibiting extreme bias with the parameter of 1. This in turn means the data subject benefits but at a cost of almost 40% of the observable third party centrality. Being able to investigate and illustrate different bias parameters, will provide valuable insight into the relations between the competing metrics. For instance, basing the refactoring on a mild bias value of 0.6 will afford a transparency reduction of 10% less than what can be maximally achieved, whilst preserving approximately 70% of the first party centrality network.

A complexity analysis is illustrated in figure 7.12. This illustration conveys the average execution time for the three single selection strategies in addition to the utility analysis. The objective of this figure is to compare the recorded execution times for the utility approach to provide insight on its associated performance penalty. Each sample within the dataset is a calculation of the time requirement for a given strategy to execute for a first party within the first party collection. The utility function is likely to exhibit higher time requirements than the other strategies given the analysis requires calculating the Pareto set as a pre-condition. The centrality strategy also requires the calculation of centrality loss through graph operations, whilst the awareness and prevalence strategies are relatively simple in their implementation, requiring only to identify a solution within Π maximising the respective values. For reference, the complexity analysis was performed on a Quad-Core Intel Core i5 Macbook Pro clocked at 2.2Ghz with 16GB of LPDDR4X system memory clocked at 3733 MHz.

We can see four box-plots each corresponding to an associated selection strategy. Generally, the values associated with the awareness and prevalence values are similar. We observe a median execution time of 0.05s for the awareness metric with a 0.052s for the prevalence metric. We further see the centrality execution time being notably longer with a median value of 0.071s. The utility approach again increases the duration with a reported median value of 0.08s. We note the outlier values also appear to correlate with the observable quartile values with the exception of the utility strategy which exhibits the largest outlier at 0.223s. The utility approach in this illustration appears to perform comparatively well to the other three approaches. However, it is important to consider this dataset only provides insight on the time performance on each strategy per first party being analysed. To make a more informed conclusion on the performance of the utility approach, it is necessary for us to consider the

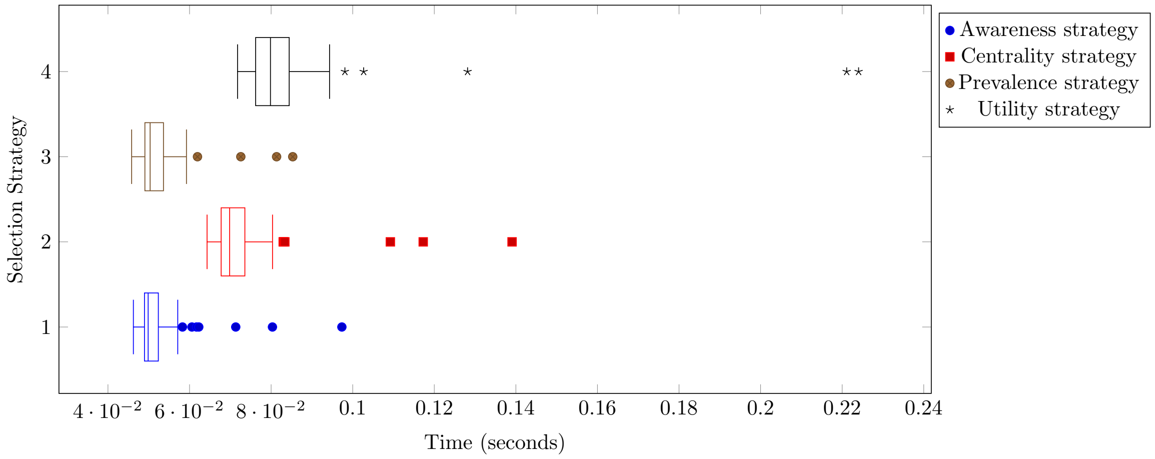


Figure 7.12: Distribution comparison between selection strategies on single first party data processor.

time complexity observable with the entire dataset of first parties.

This analysis is provided in figure 7.13. Here, we measured the time requirement to perform a specific selection strategy on the entire collection of first parties, rather than take the measurements as they are recorded for each first party. The aim of this illustration is to showcase the complexity trends for the four strategies as each technique starts to work with non-trivial workloads. We can immediately see the utility approach starts to deviate considerably from the other three strategies. A 3.24s median is observable with the minimal outlier value recorded at 2.8ms. This is considerably higher than the maximal outlier recorded for the centrality strategy, which lies at 0.59s. The other strategies are somewhat comparable, with the awareness and prevalence strategies yielding median values of 0.25s and 0.27s respectively.

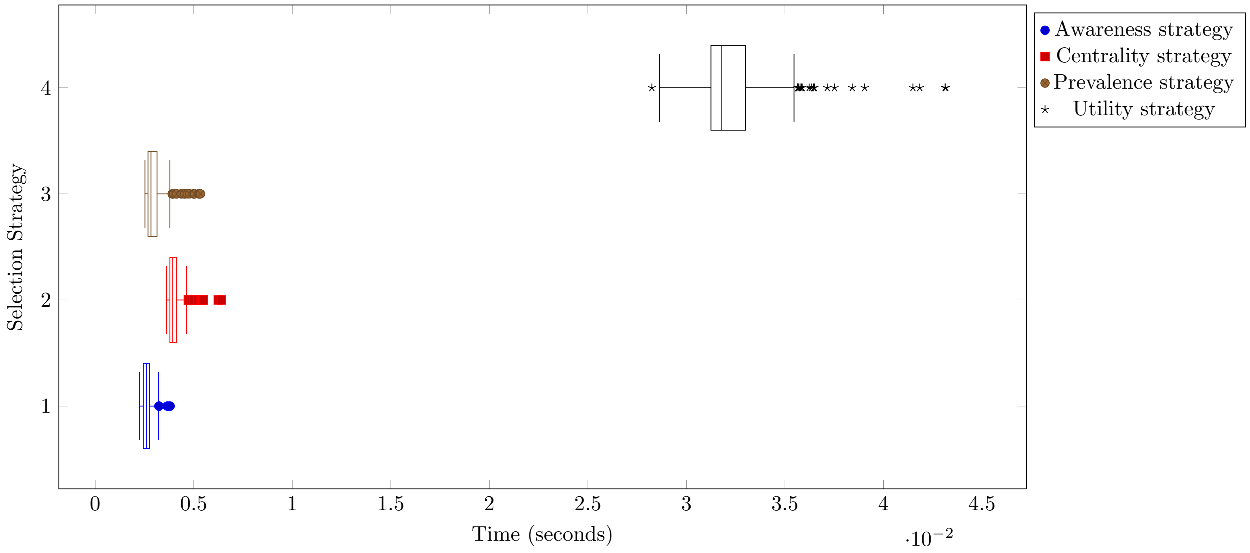


Figure 7.13: Distribution comparison between selection strategies on all visited first party data processors.

An explanation for the performance of the multi objective calculation is the constrained solution space. On average, each first party solution space is composed of 5 anti-patterns, which translates to a somewhat simple multi objective problem. Despite Pareto elicitation being a combinatorial problem, the largest solution space recorded on the dataset is 5 which in turn means the average time complexity impact per figure 7.13 is not substantial. In the event the number of anti-patterns in the solution space were to be increased, we will expect to see a sharp increase in the associated complexity analysis for the utility approach. This is because the utility time complexity is dependant both on the combinatorial calculation of dominance, in addition to the number of first parties in the analysis.

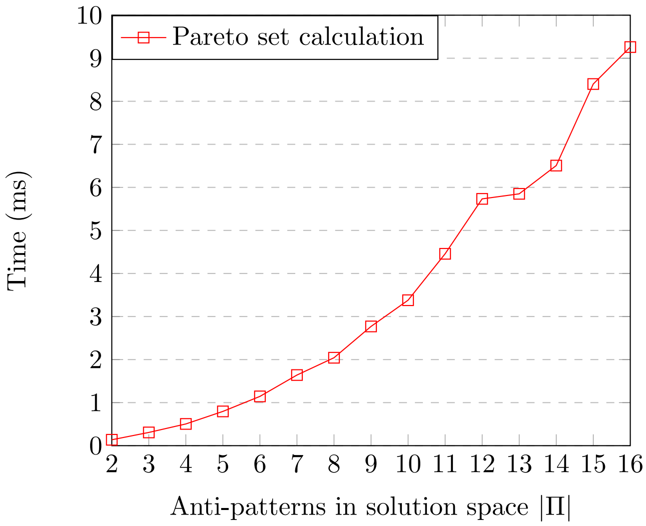


Figure 7.14: Time complexity simulation of an increased solution space of anti-patterns.

Figure 7.14 illustrates a simulated complexity analysis of the Pareto elicitation procedure. The aim of this figure is to provide insight into how the time complexity of the elicitation action increases as the number of anti-patterns within the solution space for a given first party increases. Between the ranges of 2 and 5 anti-patterns (the solution space cardinality observable in the study) the time requirement is somewhat light with a maximum observed value of 0.8ms. However, we can clearly see an exponential trend emerging as we move past these values. The theoretically maximum solution space size in the study is 13, given this corresponds to the 13 anti-patterns articulated in section 6 which yields 5.85ms. If we were to step to 14, 15 and 16 we see the requirement increase to 6.51, 8.4 and 9.26ms respectively. This complexity trend indicates calculating the Pareto set of solutions to base the utility analysis on will not be as effective in the event the number of anti-patterns within the solution space increases to a non-trivial amount.

7.6 Summary

In this chapter we unpacked a series of approaches an analyst can use in order to make a determination on how to begin the refactoring process of evasive selection of third parties in order to mitigate the identified privacy risk on an online web based system. We initially discussed an approach involving a single decision criterion to determine how to select anti-patterns based on metrics that both support the data subject or the first party. We illustrated the results of these approaches, discussing their utility as a quick and decisive way to arrive at output, but leveraged on multi objective modelling to demonstrate their limitations. We discussed multi objective analysis as an approach to illustrate the set of optimal solutions available to an analyst and use this as a benchmark to both validate the output of single decision approaches and the utility function. We position the utility function as an approach to help analysis arrive at an optimal choice for multiple first parties at scale. Alternatively, the multi objective approach can also be used to help analysis narrow down the optimal choices in the event bias information is not available or the analyst wishes to first acquire an understanding of the optimal decisions they can make before taking the analysis further.

Chapter 8

Discussion

8.1 Introduction

In this chapter we present our reflections on the work completed and conclude with an outlook on future prospects for the research along with limitations and threats. In section 8.2 we present a summary of the work completed in this research along with the key motivating factors for the work. In section 8.3. we present the research questions posed in section 1.3 and decompose how we addressed these questions in the relevant sections throughout this paper. In section 8.4 we summarise the main contributions of the work, before discussing the inherit limitations and threats in sections 8.5 and 8.6 respectively. In 8.7 we unpack the potential prospects of the research, discussing how the work can be taken forward before summarising the chapter in section 8.8.

8.2 Discussion

A key motivator for this work concerns the inability for data subjects to make effective information disclosure choices when interacting with data processors online. It is often the responsibility of the data subject to read, understand and determine the suitability of multiple distinct privacy policy for data processors they interact with on a daily basis if they wish to safeguard their privacy, or understand the consequences of interacting with data processors. Such an activity is not feasible for data subjects for three key reasons. Firstly, the language used in the creation of privacy policies can be prohibitively complex and the document itself lengthy and cumbersome for data subjects to read [101] [102]. Finally, there is no guarantee privacy policies accurately reflect the data processing activity occurring with data processors. Various authors discuss instances where privacy policies themselves present conflicting information, may conflict with observed data processing activity on implemented systems or policies from distinct data processors will conflict with one another when aggregated [103] [39]. This means data subjects often have little means by which they can acquire a full understanding of the consequences when interacting with data processors online.

The challenge is further exacerbated within online web based settings, as the capacity for data subjects to enact control over their data processing preferences is often dictated by a series of CMPs that may, or may not provide them the capacity to opt out of data processing activity. The 'take it or leave it' attitude towards privacy often assumes implied consent, where information about data processing activity is made available to the subject, with clauses indicating the usage of a web-service or dismissal of disclaimer notices will be interpreted as an explicit action of consent [104]. Such practices essentially force data subjects into a binary choice of accepting third party data processing, or not interacting with the data processor. This leads to scenarios where data subjects exhibiting conservative attitudes towards privacy may find themselves incurring privacy harms, as data processing activity does not align with their expectations.

In this work we advocate for a technical solution to assist DPOs by addressing the following key challenges. Firstly, one challenge to address concerns modelling subjective privacy conflicts objectively to gauge privacy risk in web-based data processing activities. To address this challenge we first introduce an approach to representing privacy concerns in a computational model. We introduce our awareness formalism leveraging on possible world semantics (section 4) for the purpose of delineating different knowledge states of a software object. We leverage on possible world semantics in the creation of a memory state-space representing possible states of awareness and unawareness a given principle object may exist within. This model of an objects memory allows us to articulate a model of reasoning allowing software objects to consider their own (un)awareness state regarding information they hold about themselves, and of others. Our definition of privacy concerns the capacity to regulate the manner by which information propagates on a network. We leverage on Epistemic Modal Logic to formally express an agents reasoning about knowledge and how such knowledge states become tenable resulting from data flows occurring through web based interactions. Conflict in turn, is defined as the event an agents knowledge state were to evolve in a manner mutually exclusive to the data subject, resulting from interactions a data subject has with privacy consent modules online.

Secondly, the approach allows DPOs to map out the data processing activity between one of multiple data processors. A major challenge DPOs face is a lack of tooling to effectively support the modelling and subsequent analysis of data flow networks they are responsible for [105]. The need for tooling to assist in the modelling of such systems is paramount in the event the first party under review interacts with multiple third party data processors. In section 5 we unpacked a Selenium based workflow to crawl and review the data processing activities of multiple data processors from the perspective of a data subject. The selenium crawl was responsible for eliciting two key pieces of information. Firstly, we were interested modelling the third party connections made from a first party. Secondly, we were interested to understand how both the first party and third party implemented their consent elicitation mechanisms for subject control to understand and categorise the resulting nature of data flow between the first and third party. The main benefit of the Selenium toolkit is its capacity to work with dynamic content through the execution of JavaScript. One inherit limitation we discovered with other web scraping toolkits is their limited capacity to handle dynamic content, as multiple CMP front ends are often rendered when the appropriate script is executed in the browser. Several toolkits do not return the rendered HTML source which informed our choice to leverage on Selenium. We articulate three observable information flow types: mandatory, optional and covert to categorise the data processing activity observable on a client browser.

With this information, an analyst will want to identify which data processing activity contributes to privacy risk. The output of section 5.2 will be an information flow network representative of the different data processing activities between first and third parties. The third key challenge is addressed in this work through the reconciliation of design time instrumentation choices and runtime interaction contexts. Identifying how data processing activity can contribute to privacy risk involves understanding the resulting data flow activity resulting from how data subjects interact with the consent elicitation mechanisms if available from data processors. We articulate a series of privacy anti-patterns per section 6 to formulaically determine whether the data flows between a first and third party result in the misalignment of data subject expectations. We define privacy risk as a measurement of the ratio of 3rd parties facilitate privacy anti-patterns online. The result of eliciting an information flow network and performing a formulaic analysis of privacy conflict on the information flow model will allow analysts to highlight data processing activity to prioritize their analysis.

Finally, a DPO may wish to investigate potential ways to refactor data processing activity with the objective of reducing privacy risk exhibited by a first party. Such an analysis will likely involve balancing the requirements of two important stakeholders, the data subject and the data processor. In section 7 we articulated two quantifiable metrics that can be measured from the elicited information flow network in order to measure the utility of a particular stakeholder. We mentioned data subjects will benefit from refactoring activity primarily through the reduction of information transparency in the network. Given our definition of privacy relates to the capacity to regulate the acquisition of knowledge, the reduction in knowledge transfer can be used in turn as a metric to determine to what extent a subject benefits from the refactoring activity. Conversely, the DPO will require an approach to model the impact such refactoring activity will have on the first party. We discuss in section 7 we can leverage on centrality as one quantifiable metric from the information flow model to determine the level of impact the pruning of a third party will have on a first party. Another major challenge DPOs face is ensuring cooperation amongst other stakeholders in an organisation [106]. Typically, advocating for change requires the presentation of demonstrable evidence that such proposals make sense to the business operations of an organisation. The methods presented in this thesis can benefit a DPO by presenting them with a series of metrics to assess the efficacy of proposed refactoring activities on a first party. Specifically, being able to plot the impact a refactoring activity will have on the data subjects along with the incurred cost to the first party will allow a DPO to present measurable evidence to substantiate any proposals to demonstrate compliance with data protection regulations. Additionally, the different strategies discussed can further be used to articulate the perceived optimal tradeoff for a given first party to stakeholders. A more nuanced analysis through the acquisition of Pareto efficient outcomes in combination with a utility function parameterised with biases can justify any recommendations made.

The approach detailed in this thesis aims to enshrine the principle of accountability by providing a tool and series of methods to facilitate privacy risk analysis pursuant to the identification of data processing activity resulting in incidental privacy harms to the data subject [107] [108]. In light of this, the selective pruning of third party data processors facilitating privacy conflict can in turn be used by DPOs and by extension, organisations, to provide demonstrable evidence to regulatory bodies of their attempts to maintain compliance with data protection regulation through a systematic review of data processing activity and its consequences on data subjects. By modelling how privacy risk manifests in data processing networks, DPOs can advocate for the reduction in data processing activities facilitating privacy harms to data subjects without requiring data subjects to enact their own independent precautionary measures. This shifts the responsibility of managing privacy conflict from the data subjects. Of course, the key benefit to data subjects lies with the reduction of data processing activity leading to misaligned privacy expectations. The benefits to data processors rest with the fostering of user trust in their services and with the ability to demonstrating their commitment to adhering to the principles of privacy by design through modelling and review of their data data processing networks.

8.3 Research Questions

Research Question 1

Research question 1 asked whether there are models that can be leveraged on to articulate the expectations of data subjects and the evolution of data processors knowledge. We address this research question in section 4 through the articulation of the awareness formalism. Epistemic Modal Logic is a logic to reason about the acquisition of knowledge. The application of Epistemic Modal Logic allows us to model how software agents reason about their own understanding of the world, and of others. With respect to privacy, such reasoning concerns propositions that can either relate to data subject attributes (an atomic proposition) or the current knowledge state of another agent (composite proposition).

Given a proposition there are two possible worlds, one where the proposition is true, and the other where the proposition is false. If an agent exists within a current world such that the two worlds of possibility are accessible via accessibility relations, then an agent is said to be unaware of the proposition as they are unable to distinguish the truth or falsehood of the proposition. Conversely, if they exist within a world where only a reflexive accessibility relation pointing to the proposition is true, the agent is aware of the proposition given they are able to distinguish the truth and falsehood of the proposition. We articulate two states, awareness and unawareness that can be used subsequently as a means to model the privacy expectations of data subjects when they interact with CMPs. For instance, in the event a subject were to opt into data processing activity by a third party, it stands to reason they will be aware of the knowledge state of the data processor resulting from data processing activity. Likewise, the resulting awareness state of a data processor can be modelled with awareness statements.

An assertion in turn, maps to a state of awareness or unawareness which prescribes a desirable manner by which knowledge exchange should occur by stating how the (un)awareness states of agents should change. This allows us to model independently the knowledge state of the data subject and the processors. Conflict occurs on the awareness model in the event the awareness state of a data subject does not align with the awareness state of a data processor.

Research Question 2

Research question 2 asks whether there is a methodology to reduce the time requirement for modelling the data flow networks on a series of data processors. We address this research question within section 5. The Selenium implementation allows an analyst to visit a series of first parties in succession to autonomously detect and identify third party data processors. In our implementation, we employed a series of rule bases to allow the Selenium browser to interpret the HTML rendered within the browser to identify implementation signatures relating to specific data processing behaviour of a first party. For instance, data processors operating on the legal basis of informed consent will provide privacy consent modules to the data subject. The Selenium crawler was trained in this work to recognise consent module implementations from the crawled first parties. The same process was followed for those data processors operating on implied consent, through the implementation of data processing notices. Finally, those operating without legal basis for data processing would not have any identifiable HTML elements for the crawler to detect. The Selenium workflow was able to elicit a network of 139 data processors, 51 first parties, 89 third parties.

The workflow occurs in two main steps. Initially, we acquire a series of tuple elements from the observable behaviour of first parties. In addition to this we also acquire the raw requests made to external data processors. The second step involves pruning this data, by cross examining the dataset from the Lightbeam addon with the external corpus to validate those third party trackers. This also occurs for the optional vendors identified through the consent mechanisms on the first party. Finally, we aggregate the data to form a directed information flow network, where edges within the pruned lighbeam data are annotated with the observed behaviour from the first party. The main benefit of the Selenium workflow is in the use of automation. The DPO is not involved in the elicitation of the data flow network. The only requirement of the DPO would be the collection of the Lightbeam dataset and the further enactment of the abstraction procedure. Being able to leverage on a near fully autonomous process will alleviate the time requirement for the DPO in comparison to a manual review of data processing activity as prescribed in legal or technical documentation.

Research Question 3

Research question 3 asks whether a method can be used to validate data processing activity contributing to privacy risk. We address this question within section 6 by introducing the concept of anti-patterns. An anti-pattern contains two elements. Firstly, it describes a usage context between the data subject, first and third party data processors. This high level description prescribes the nature by which CMPs are instrumented on the data processors, and by extension the resulting data sharing principles to disseminate user data. The principle here is the data sharing activities will transform the awareness states of the data processors, and the manner by which CMPs are implemented will in turn transform the data subjects awareness state. An anti-pattern describes a scenario where these activities will result in the evolution of knowledge between the stakeholders in a mutually exclusive manner, such that there are inconsistencies between the expectations of a data subject and/or the knowledge gain of data processors.

The second aspect of an anti-pattern we use in order to validate the existence of conflict. We leverage off the conflicting semantics articulated in section 4 to illustrate a pair of awareness constructs being in a state of conflict. Each privacy anti-pattern presents a function Δ, λ where Δ returns True in the event the constructs can be mapped to explicit or implicit semantics. Likewise, λ returns True in the event the constructs can be mapped to uncertainty conflict semantics.

The next step involves mapping a privacy anti-pattern to an edge within the information flow network model which can be elicited in section 5. Each edge within the information flow network is representative of data processing activity between two data processors represented as vertices. Each vertex within the graph structure can be annotated with data processing behaviour as observed from the Selenium workflow. This in turn allows us to match the observed data processing activity, to one of the rows within table 6.2 which illustrates the data processor states generating privacy conflict. Each row illustrates a distinct pairing of optional, mandatory and covert data processing activity for the first and third party. This in turn allows analysts to formulaically determine which edges on the information flow network contribute to privacy risk.

Research Question 4

Research Question 4 asks what methods can be used to reduce a state-space of possible candidate refactoring recommendations to realise an optimal outcome for data subjects and processors. We address this research question within section 7.1. Being able to elicit an information flow network and identify edges contributing to privacy risk present the analyst with the capacity to model changes to the network in the event edges exhibiting certain anti-patterns are refactored. Each refactoring activity corresponds to the removal of third parties matching a specific anti-pattern. In order to consider whether an outcome can be viewed as optimal for a first party and a data subject, we need to understand two things. Firstly, we model the competing nature of multiple requirements to be catered to in the refactoring process. In section 7.2 we illustrate two metrics to measure the impact for both the data subject and the data processor when refactoring activity takes place. Centrality is leveraged on to provide insight on the business impact a refactoring approach will have, whilst the levels of awareness from data processing behaviour model the benefit to data subjects.

Secondly, we need to model the state-space of possible solutions to acquire insight on the impact of removing third parties facilitating an anti-pattern. This allows us to model the possible refactoring options available to an analyst as a series of solutions within a competing objective space. The challenge of reducing this solution space is well suited to the application of the NGSA-II approach in procedure 7 to elicit a collection of optimal choices for a data subject. The challenge with identifying an optimal outcome in turn involves investigating which outcome in the Pareto efficient set to recommend to an analyst. To achieve this, we leverage on a utility function, which operates on weighted bias parameters programmable by the analyst. The approach detailed in section 7.3 allows an analyst to first elicit the Pareto front, then with configured (or default) biases, be presented with a single anti-pattern recommendation for refactoring. We evaluated whether this was a feasible approach by considering the alternative approaches an analyst can take. Namely the selection of anti-patterns based on whether they maximise awareness, centrality or are the most prevalent on the information flow model. In section 7.5 we illustrated a comparison focusing on the reduction of observable privacy risk, whilst illustrating the tradeoffs faced by the data subject and processor. The conclusions indicate the benefit of leveraging on bias parameters is a more granular analysis to make recommendations that do not fall into one of two extremes in terms of the utility they provide to stakeholders. The decisiveness of the approach allows an analyst in turn, to calibrate what they consider to be optimal by virtue of the bias parameters provided to the method.

The benefit of the approach is the capacity for analysts to formulaically acquire an individual recommendation for each first party whilst respecting any inherit bias parameters the analyst may provide. Alternatively, the analyst can look at the general trends applicable across an entire dataset if they are looking to make a general recommendation. This caters to two envisioned use cases, one where an analyst is looking to make individual or bespoke recommendations for one or multiple data processors they are contracted by. Alternatively, in the event an analyst is responsible for making recommendations on complex systems spanning multiple processors under the same controller, they can make a general recommendation applicable across the data flow network.

8.4 Contributions

This work provides a series of contributions to not only privacy and privacy conflict analysis, but also the wider field of socio-technical system modelling. The contributions of this work are as follows:

A privacy awareness model. This work has illustrated how we can represent privacy concerns in software by leveraging off possible world semantics to articulate an awareness model to capture the current knowledge state of an object. The awareness model is an instantiation of our privacy definition being the regulation of knowledge acquisition online. The privacy awareness model in this work can in turn be used by engineering stakeholders to computationally represent the expressed privacy concerns of objects in the analysis of software models to demonstrate how such models can empirically satisfy these concerns.

Privacy conflict formalisms. This work has illustrated how we can leverage off the awareness model to represent privacy conflict. Specifically, we have illustrated how privacy conflict can manifest through discussion of a series of rules concerning the pairing of incompatible assertions forming explicit, implicit and uncertainty conflicts. We demonstrated in section 5 how the conflict model can be applied to an online web setting where we conducted analysis of privacy conflicts through the articulation of anti-patterns involving the generation of mutually incompatible privacy assertions. It is the intention of this work such formalisms can be used by engineering stakeholders to identify possible areas of concern facilitating privacy conflicts in the review of both developed software systems or their concept models.

Strategies for conflict analysis. We have illustrated in this work an approach utilising the NGSA-II method to first elicit a set of Pareto efficient refactoring options prior to enacting a utility approach on the resulting Pareto set. The benefit of first acquiring the Pareto set is the collection represents the best possible strategies an analyst can recommend when looking to balance two competing objectives in the analysis. We evaluate the outcome of applying bias parameters in the selection of a single recommendation from the Pareto set against a series of trivial solutions. We consider how effective the utility approach is in reducing observable privacy risk compared to these single decision strategies, along with the capacity to consider the impact of granular recommendations not possible with the trivial methods.

A workflow to elicit an information flow model. We have discussed an approach to allow analysts to model the data processing activity on a series of data processors for the purpose of eliciting an information flow network. We have discussed how such an information flow network model can represent the behaviours of data processors and have unpacked two behaviours within legal basis for data processing, in addition to identifying processing activity outwith any legal basis.

A case study on privacy risk refactoring. We have unpacked a case study investigating observable privacy risk associated with the configuration of web based subject control mechanisms online. We articulate an information flow elicitation workflow to autonomously acquire a network model of data flow activity between data processors. We unpack a series of steps to annotate such an information flow network with observable behaviours for the purpose of cross examining such behaviour with privacy anti-patterns to determine whether such connections on the model contribute to privacy risk. We then discuss how we can leverage on the provided utility function to identify a series of optimal trade-offs in a granular analysis of multi objective problems applicable to each first party identified in the analysis. We illustrate how an analyst can make individual decisions based on their preferences, or aggregate the outcomes to make general recommendations.

8.5 Limitations

There are some limitations to the work we wish to address:

Reliance on External Data.

The methodology in section 5 involving the elicitation of a network for conflict analysis required the use of external data-sets for the purpose of network verification. The dependence on an external data-set from [94] to verify 3rd party trackers and a further data set to validate 3rd party data processing behaviour impacts on the scope of the analysis by limiting our capacity to elicit network information. It was however necessary for such measures to be taken to ensure the validity of the elicited network data and in turn reduce noise. An unfortunate consequence of this approach is the study will not have included all of the 3rd parties from the selenium crawl within the analysis. A way by which we can address this shortcoming is to expand on the external data-sets utilised when qualifying elicited 3rd party nodes. If we are able to expand on the public corpus we leverage on when pruning the resulting network of 3rd parties, it will afford us the capacity to more accurately model complex networks.

Formalism Completeness.

We do not argue the observed anti-patterns are exhaustive. Certifying the anti-patterns presented in this thesis are exhaustive will require a brute-force crawl with assumed infinite resources. Beyond 51 first parties we did not gain additional knowledge, and therefore in presentation of the work we weighed further analysis against the benefit of conciseness, illustrative clarity, and significance. However, we note because a privacy anti-pattern remains unconfirmed by the experiments discussed in this thesis, it does not guarantee its non-existence. It is very possible that additional anti-patterns will become known, given enough time to sufficiently crawl a large enough dataset.

Selenium Limitations.

The websites analysed in this work originated from English domains. As such, the codified parser is only able to parse site content from English-based websites. Despite our methodology technically permitting such data sources, we have not codified our parser to interpret privacy consent mechanisms from first and third parties from non-native English speaking geographic locations.

Awareness Scope.

A given awareness statement concerns at least one principal operator p and at most 2 reference objects r_1 and r_2. It is possible to leverage on such semantics to model the awareness state of additional reference objects, however we limited our scope of concern to 2 objects. There is plenty of research within social science supporting the theory that the scope of concern for user understanding is limited to approximately 2 hops from the individual. Therefore, when representing such reasoning in a computational model, we leverage on this literature to gauge how to bound the analysis. Increasing the number of reference objects in the scope of analysis will result in exponentially larger memory state-spaces being generated. In theory this will also mean additional ways by which conflict can be modelled, as naturally this will increase the possibility for mutually exclusive requirements being paired, however the utility of conducting an analysis at such a level given what has already been demonstrated is unlikely to warrant the effort of implementation.

Concurrency.

In this thesis we have discussed analysis of privacy conflict in scenarios that only concern single disclosure actions occurring at any given time. We have not in this work discussed an approach to modelling and addressing concurrent disclosure actions. However, an interesting next step for this work may involve investigation of concurrent disclosure actions for conflict analysis. Instantiating an information flow network model as a full MAS implementation will make an excellent test bed for concurrency experiments.

8.6 Threats to Validity

Selenium Interpreter.

In this work we implemented the Selenium crawler to parse web based consent toggles to elicit a series of optional flows involving third parties. However, the capacity for the crawler to parse such content accurately is largely constrained by the sample size of the crawl we discussed in our methodology. The rule base the crawler has been trained on will not be able to recognise novel privacy consent mechanisms from sites excluded from the original experiments. In such an instance, the crawler will have to be retrained to recognise this new information. A similar issue can arise in the event any substantial changes are made to the consent mechanisms employed by a first party, or whether a new solution altogether is sought. In both instances, the crawler will not be able to autonomously reconcile such changes and it will have to be re-trained.

Assumption of Data Processor Activity.

In this work, we assume third party data processors can be made known to an analyst through external corpus data sets[1]. Throughout this thesis, we make the assumption third party data processors identifiable within such repositories will be active entities online. However this assumption may always hold. It is feasible to consider the active state of third party data processors is not static, but rather a dynamic property of an information flow network, and therefore, there is a requirement to better understand the active state of a data processor to more accurately model the data flows in a network of data processors. This is something we consider to be an immediate next goal of the research going forward.

[1] Such as `https://ssc.io/trackingthetrackers/`

Accuracy of Corpus.

It is assumed when performing the network validation step, the information we leverage on in the external data-sets is accurate. Another problem with this dependency concerns the fact on content delivery networks, acquisitions, re-branding and general updates to the tracking scripts and cookie identifiers can occur. In such instances the accuracy of the data set will start to be affected and it will be necessary to consider an updated data-set if one becomes available. One possible mitigation to this threat is the diversification of additional data-sets online. The more third party data sets we leverage on in the analysis, the greater the estimated accuracy of the results.

Assumed User Behaviour.

A given anti-pattern in this work is assumed to represent a usage scenario involving a user which will leave them with misaligned privacy expectations. However, this assumption may not always hold. For example, privacy anti-pattern 2 illustrates a scenario where a data subject su interacts with the consent mechanisms offered by F and F', they will accept data processing on F but reject data processing on F'. Of course, there is a possibility the user will simply accept data processing on both F, and F' resulting in no misaligned privacy expectations.

Data Subject Knowledge Transfer.

It is assumed in this work the (un)awareness levels modelled in objects are representative of the levels of (un)awareness of their respective users. This assumption however may not always hold. For example, it may be the case information may be held in some technical buffer and therefore the users knowledge is not updated synchronously with their software representation.

Determining Benefit to Data Processors.

In order to determine the measurable impact refactoring an anti-pattern will have on a data processor, we leveraged on the metric of centrality. Centrality is not the only manner by which business impact can be measured, nor is it the only manner by which the inherit value of a third party can be estimated for the first party. For instance, we assume in this thesis third parties exhibiting large degrees of centrality will contribute to the business paradigm of ’big data’ more so than those third parties with low levels of centrality, due to their influence. However, this assumption may not always hold, and the realistic business value may not be something easily acquired from an information flow network without appropriate business insight. For instance, the nature and importance of the service provided by a third party is not an insight easily acquired from a data flow model.

8.7 Future Work

There are several ways by which this work can be expanded on. In this section we look to unpack some potential avenues for further research:

Privacy Patterns.

In this work we discussed the concept of privacy anti-patterns in section 5. This concept affords us the capacity to understand how privacy conflict manifested on the case study involving the configuration of consent elicitation tools. However, it is also feasible for such an analysis to focus on the implementation of consent mechanisms facilitating privacy patterns.

In theory, the investigation of privacy patterns will afford a richer analysis than what has been undertaken in this thesis currently. There is potential for such privacy risk metrics to be updated, to reflect not only the potential for users to be subject to misaligned privacy expectations, but rather illustrate a probabilistic value for such risk. The determination of probability values will likely factor in both such usage contexts facilitating risk and those which do not. Therefore, an interesting outcome of such analysis will be the investigation of whether a given series of consent module implementations facilitate greater probability levels of risk than another.

Learning Consent Patterns.

In section 5 we discussed the method followed with the selenium crawler for the purpose of eliciting network information with Lightbeam and the establishment of third parties in an information flow network offering consent modules. One inherent limitation of this approach concerns the inflexibility of the crawler implementation as it has currently been trained to correctly parse HTML content from the data set acquired in our study to elicit optional flow patterns. The remedy for such a shortcoming involves the application of learning techniques to assist in training the crawler to recognise novel or unseen consent implementations.

The utility of this endeavour would be in the provision of a tool capable of continued accurate reporting on the consent mechanisms for analysts in the event associated partners modify their consent elicitation tools or adopt new mechanisms altogether. Currently, the implementation will require re-training to recognise patterns. Whilst the implementation was satisfactory for our study, having a training set available will make the crawler more flexible in its parsing of HTML supported consent modules, reducing the time investment required for stakeholders to utilise such methods.

Learning Biases.

We discussed a formula to determine a scalarised value for the purpose of determining conflict decision metrics. The advantage of this formula lies with the bias values ω. In the study, the biases of users was not available to us, therefore the default weighting values were used. However, one interesting avenue of research concerns the elicitation of bias values from past user interactions.

Often privacy conflict analysis is a field investigating user relation metrics on social network platforms. The aim of such research is to understand various measurable properties to make assessments on what is considered acceptable and unacceptable in such settings. An interesting avenue of research concerns the capacity to leverage on historic behaviour of users in social network systems, for the purpose of eliciting an understanding on how such users gauge their information sensitivity values. For example, it is intuitive to argue users who frequently utilise location sharing applications to otherwise ’check in’ to locations and events are more relaxed with the sharing such attributes with others, compared to users who seldom disclose such information in public settings. Such information could be utilised to better inform refactoring options aligning with the granule preferences of data subjects.

Privacy Anti-Pattern Repository.

One of the main contributions of this work has been the articulation of privacy anti-patterns. In our work, we illustrated how these privacy anti-patterns can be facilitated in online web interaction scenarios, specifically focusing on the interactions a user has with CMPs. One avenue to explore in the future is the creation of an online repository to articulate not only the privacy anti-patterns identified in this thesis, but additional anti-patterns identified in the future. Having available an online appendix of privacy anti-patterns, their associated formalisms, and applicable contexts can be a useful tool for the wider privacy research community. We see two key potential benefits. The first being the capacity to have additional authors contribute to the repository, the second being the evaluation of the applicability of privacy anti-patterns to different research contexts not covered within the scope of this thesis.

Dynamic Detection of Data Processor State.

In this work we operate under the assumption all third party data processors are active entities looking to track data subjects online. However, we noted this assumption will not always hold. What we have done in this work, is look to the outbound requests made to a data processor in order to make a determination that data processing activity occurs. One way we envision taking this work forward is through the investigation of an approach to identify the active state of a data processor. Being able to acquire this insight will in turn allows us to create a dynamic data flow network more accurately representing the data processing activity between a first party and those active third parties at the point of analysis.

Additional Logical Formalism

The formalisms introduced in this work leverage on Epistemic Modal Logic for the purpose of modelling unambiguously the knowledge a software object can acquire. We introduced the concept of awareness and unawareness to formally express how a data subjects privacy expectations can be formed when interacting with data processors, and how data processor knowledge will change as a result of data processing activity. We can expand on the formalisms by combining our awareness formalism with other formalisms such as Linear Temporal Logic (LTL). One interesting benefit of implementing awareness in addition to LTL is the capacity to leverage on LTL properties to extend the expressive capacity of the formalism to sequence events across a time series.

Exploitable Toolkit

The core focus of this thesis has been in the provision of methods to support DPOs in the modelling of complex data flow networks and the quantification of privacy risk within such models. The outcome of the work is a series of refactoring recommendations based on programmable biases towards competing objective functions mapping to measurable properties on the data flow model. This thesis aimed to demonstrate the benefit towards data subjects in the reduction of privacy risk on evaluated first parties. Another way this work can be expanded on is in the construction of a software suite that can be exploited in industrial sectors. Empirical studies can be conducted with different stakeholders to better inform the development of an open source initiative to instantiate this research.

8.8 Summary

In this section we have provided an account of the work completed throughout this course of research. We have provided a discursive self analysis on the implementation along with a critical take on how specifically this research can assist the key benefactors. We unpacked two key challenges that DPOs face, the first being a lack of tooling to conduct effective analysis of data flow networks on complex systems. The second being the difficulty in persuading other stakeholders within organisations to consider a re-conceptualisation of business processes as necessary in pursuit of demonstrating regulatory compliance. We see the methods and techniques discussed in this thesis being of benefit to DPOs in addressing these fundamental professional challenges. We discussed the inherit limitations in the technical implementations, assumptions made, before concluding with an outlook on potential future research endeavours from this thesis.

Chapter 9

Conclusions

Some argue the current state of privacy in modern sociotechnical systems relies on the implementation of privacy policies. The assumption data subjects will be able to effectively read, understand and agree to the terms at scale and at pace is not a reasonable expectation. It leaves many data subjects leveraging on technical means to safeguard their own privacy, which creates a division rewarding users with a technical understanding and disadvantages those without. CMPs are a contemporary solution to address the concerns of data protection regulation, however the manner by which the legal basis for data processing activity occurs between multiple data processors involved in the dissemination of user data exemplifies the challenge of maintaining data subject privacy expectations in complex inter-dependant settings. If we are able to reduce the privacy risk faced by end users via technical means, in turn we will be able to reduce the cognitive burden placed on users when interacting with data processors online and assist them in better understanding the consequences of their interactions with data processors online.

To this end, we have discussed a series of mechanisms and models to represent and review deployed web based systems for the purpose of acquiring insight on how the implementation choices made between data processors can foster privacy risk for data subjects, with a focus on understanding how misaligned privacy expectations can occur. We position the work in this thesis being of benefit to both businesses and data subjects. Businesses will benefit from understanding which data processors associated with a client contribute to the risk faced by data subjects. Data subjects in turn will benefit from the reduced tracking online quantifiable as a reduction in privacy risk. DPOs in particular can leverage on the methods in this thesis as demonstrable evidence to regulatory bodies in upholding the principle of accountability and fairness in GDPR.

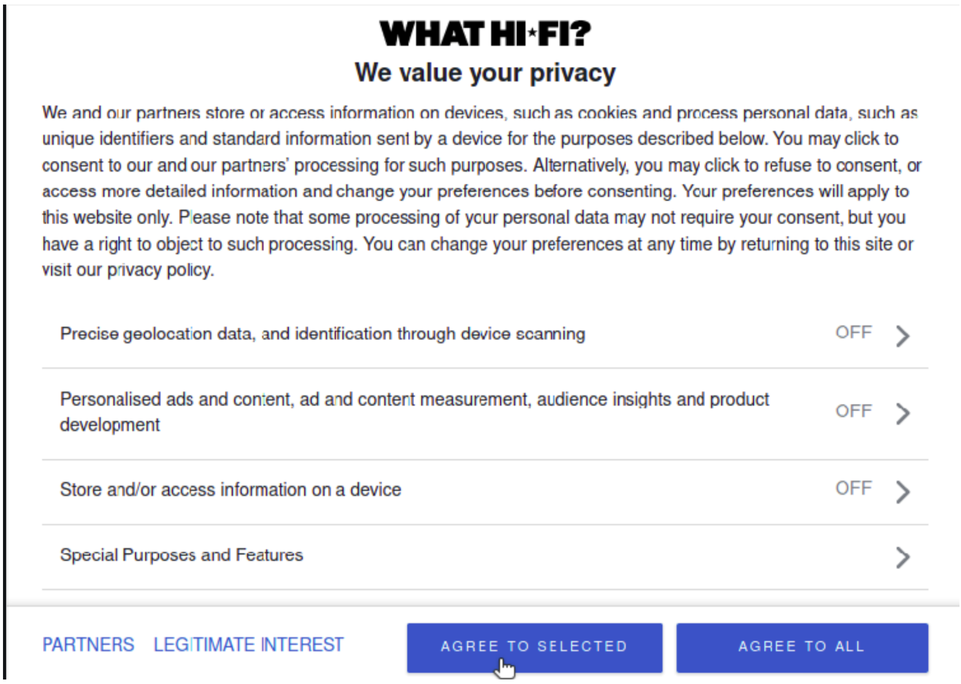
There are various ways by which we can take this work forward. For instance, the capacity to model data processing activities dynamically as to verify the active state of a data processor is something we will look to investigate. We envision a richer, more accurate analysis being possible, with the inclusion of additional online data tracking corpus. One possible way to take this research forward is to investigate the creation of a privacy agent similar in concept to the Platform for Privacy Preferences project. We are interested in the creation of a browser addon, capable of eliciting a data flow network and dynamically identifying privacy conflicts and articulating the nature by which privacy conflicts are occurring. We envision such a tool being an interesting transparency technology to bridge the knowledge gap between user understanding and the ground truth on data processing activity and whether such activity is causing them privacy harm.

The investigation of privacy patterns is another interesting aspect of the work. It is intuitive to consider there will be utility in the articulation of formalisms capturing the implementation of data processor activities which is conducive to privacy expectations of users, as opposed to the opposite case, which has been the main focus of this work. Such work we envision being of benefit to DPOs in a similar usage context as privacy anti-patterns.

We intend to publish an online index of the privacy anti-patterns articulated in this work for the privacy research community. It is hoped the formalisms in this work can be evaluated and applied in different research contexts. In the future, additional anti-patterns and deeper insights into how such anti-patterns manifest or are otherwise applicable in different technical domains can be expanded on, formulating a public corpus that can be exploited to better inform the next generation of privacy aware software systems.

Appendix A

Appendices



We value your privacy

We and our partners store or access information on devices, such as cookies and process personal data, such as unique identifiers and standard information sent by a device for the purposes described below. You may click to consent to our and our partners' processing for such purposes. Alternatively, you may click to refuse to consent, or access more detailed information and change your preferences before consenting. Your preferences will apply to this website only. Please note that some processing of your personal data may not require your consent, but you have a right to object to such processing. You can change your preferences at any time by returning to this site or visit our privacy policy.

Precise geolocation data, and identification through device scanning OFF

Personalised ads and content, ad and content measurement, audience insights and product development OFF

Store and/or access information on a device OFF

Special Purposes and Features

Figure A.1: An example of a privacy consent elicitation module.

Table A.1: Explicit conflict possible instances.

Label	Awareness Class	Assertion a	Assertion a'
1	A1	$A_p f$	$\neg A_p f$
2	A1	$\neg A_p f$	$A_p f$
3	A2	$A_p A_{r1} f$	$\neg A_p A_{r1} f$
4	A2	$\neg A_p A_{r1} f$	$A_p A_{r1} f$
5	A2	$A_p \neg A_{r1} f$	$\neg A_p \neg A_{r1} f$
6	A2	$\neg A_p \neg A_{r1} f$	$A_p \neg A_{r1} f$
7	A3	$A_p A_{r1} A_p f$	$\neg A_p A_{r1} A_p f$
8	A3	$\neg A_p A_{r1} A_p f$	$A_p A_{r1} A_p f$
9	A3	$A_p A_{r1} \neg A_p f$	$\neg A_p A_{r1} \neg A_p f$
10	A3	$\neg A_p A_{r1} \neg A_p f$	$A_p A_{r1} \neg A_p f$
11	A3	$A_p \neg A_{r1} A_p f$	$\neg A_p \neg A_{r1} A_p f$
12	A3	$\neg A_p \neg A_{r1} A_p f$	$A_p \neg A_{r1} A_p f$
13	A3	$A_p \neg A_{r1} \neg A_p f$	$\neg A_p \neg A_{r1} \neg A_p f$
14	A3	$\neg A_p \neg A_{r1} \neg A_p f$	$A_p \neg A_{r1} \neg A_p f$
15	A4	$A_p A_{r1} A_{r2} f$	$\neg A_p A_{r1} A_{r2} f$
16	A4	$\neg A_p A_{r1} A_{r2} f$	$A_p A_{r1} A_{r2} f$
17	A4	$A_p A_{r1} \neg A_{r2} f$	$\neg A_p A_{r1} \neg A_{r2} f$
18	A4	$\neg A_p A_{r1} \neg A_{r2} f$	$A_p A_{r1} \neg A_{r2} f$
19	A4	$A_p \neg A_{r1} A_{r2} f$	$\neg A_p \neg A_{r1} A_{r2} f$
20	A4	$\neg A_p \neg A_{r1} A_{r2} f$	$A_p \neg A_{r1} A_{r2} f$
21	A4	$A_p \neg A_{r1} \neg A_{r2} f$	$\neg A_p \neg A_{r1} \neg A_{r2} f$
22	A4	$\neg A_p \neg A_{r1} \neg A_{r2} f$	$A_p \neg A_{r1} \neg A_{r2} f$

Table A.2: Possible instances of implicit conflict.

Label	Given a	C_a	Indirect Conflicting a'
A2-1	$A_pA_{r1}f$	$A_{r1}f$	$\neg A_{r1}f$
A2-2	$A_p\neg A_{r1}f$	$\neg A_{r1}f$	$A_{r1}f$
A2-3	$\neg A_pA_{r1}f$	$A_{r1}f$	$\neg A_{r1}f$
A2-4	$\neg A_p\neg A_{r1}f$	$\neg A_{r1}f$	$A_{r1}f$
A3-1	$A_pA_{r1}A_pf$	$A_{r1}A_pf, A_pf$	$\neg A_{r1}A_pf, \neg A_pf$
A3-2	$A_pA_{r1}\neg A_pf$	$A_{r1}A_pf, \neg A_pf$	$\neg A_{r1}A_pf, A_pf$
A3-3	$A_p\neg A_{r1}A_pf$	$\neg A_{r1}A_pf, A_pf$	$A_{r1}A_pf, \neg A_pf$
A3-4	$A_p\neg A_{r1}\neg A_pf$	$\neg A_{r1}\neg A_pf, \neg A_pf$	$A_{r1}\neg A_pf, A_pf$
A3-5	$\neg A_pA_{r1}A_pf$	$A_{r1}A_pf, A_pf$	$\neg A_{r1}A_pf, \neg A_pf$
A3-6	$\neg A_pA_{r1}\neg A_pf$	$A_{r1}\neg A_pf, \neg A_pf$	$\neg A_{r1}\neg A_pf, A_pf$
A3-7	$\neg A_p\neg A_{r1}A_pf$	$\neg A_{r1}A_pf, A_pf$	$A_{r1}A_pf, \neg A_pf$
A3-8	$\neg A_p\neg A_{r1}\neg A_pf$	$\neg A_{r1}\neg A_pf, \neg A_pf$	$A_{r1}\neg A_pf, A_pf$
A4-1	$A_pA_{r1}A_{r2}f$	$A_{r1}A_{r2}f, A_{r2}f, A_pA_{r2}f$	$\neg A_{r1}A_{r2}f, \neg A_{r2}f, \neg A_pA_{r2}f$
A4-2	$A_pA_{r1}\neg A_{r2}f$	$A_{r1}A_{r2}f, \neg A_{r2}f, A_p\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, A_{r2}f, \neg A_p\neg A_{r2}f$
A4-3	$A_p\neg A_{r1}A_{r2}f$	$\neg A_{r1}A_{r2}f, A_{r2}f, A_pA_{r2}f$	$A_{r1}A_{r2}f, \neg A_{r2}f, \neg A_pA_{r2}f$
A4-4	$A_p\neg A_{r1}\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, \neg A_{r2}f, A_p\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, A_{r2}f, \neg A_p\neg A_{r2}f$
A4-5	$\neg A_pA_{r1}A_{r2}f$	$A_{r1}A_{r2}f, A_{r2}f, \neg A_pA_{r2}f$	$\neg A_{r1}A_{r2}f, \neg A_{r2}f, A_pA_{r2}f$
A4-6	$\neg A_pA_{r1}\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, \neg A_{r2}f, \neg A_p\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, A_{r2}f, A_p\neg A_{r2}f$
A4-7	$\neg A_p\neg A_{r1}A_{r2}f$	$\neg A_{r1}A_{r2}f, A_{r2}f, \neg A_pA_{r2}f$	$A_{r1}A_{r2}f, \neg A_{r2}f, A_pA_{r2}f$
A4-8	$\neg A_p\neg A_{r1}\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, \neg A_{r2}f, \neg A_p\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, A_{r2}f, A_p\neg A_{r2}f$

Table A.3: Possible instances of implicit conflict by virtue of combined constituents.

Label	**Given** A	**Conflicting** C_α	**Conflicting** C'_α	**Given** A'
A2-1	$A_pA_{r1}f$	$A_{r1}f$	$\neg A_{r1}f$	$A_p\neg A_{r1}f$
A2-2	$A_p\neg A_{r1}f$	$\neg A_{r1}f$	$A_{r1}f$	$A_pA_{r1}f$
A2-3	$\neg A_pA_{r1}f$	$A_{r1}f$	$\neg A_{r1}f$	$\neg A_p\neg A_{r1}f$
A2-4	$\neg A_p\neg A_{r1}f$	$\neg A_{r1}f$	$A_{r1}f$	$\neg A_pA_{r1}f$
A3-1	$A_pA_{r1}A_pf$	$A_{r1}A_pf, A_pf$	$\neg A_{r1}A_pf$, $\neg A_pf$	$(\neg)A_p\neg A_{r1}A_pf$, $(\neg)A_p(\neg)A_{r1}\neg A_pf$
A3-2	$A_pA_{r1}\neg A_pf$	$A_{r1}A_pf, \neg A_pf$	$\neg A_{r1}A_pf$, A_pf	$(\neg)A_p\neg A_{r1}A_pf$, $(\neg)A_p(\neg)A_{r1}A_pf$
A3-3	$A_p\neg A_{r1}A_pf$	$\neg A_{r1}A_pf, A_pf$	$A_{r1}A_pf$, $\neg A_pf$	$(\neg)A_pA_{r1}A_pf$, $(\neg)A_p(\neg)A_{r1}\neg A_pf$
A3-4	$A_p\neg A_{r1}\neg A_pf$	$\neg A_{r1}\neg A_pf, \neg A_pf$	$A_{r1}\neg A_pf$, A_pf	$(\neg)A_pA_{r1}\neg A_pf$, $(\neg)A_p(\neg)A_{r1}A_pf$
A3-5	$\neg A_pA_{r1}A_pf$	$A_{r1}A_pf, A_pf$	$\neg A_{r1}A_pf$, $\neg A_pf$	$(\neg)A_p\neg A_{r1}A_pf$, $(\neg)A_p(\neg)A_{r1}\neg A_pf$
A3-6	$\neg A_pA_{r1}\neg A_pf$	$A_{r1}\neg A_pf, \neg A_pf$	$\neg A_{r1}\neg A_pf$, A_pf	$(\neg)A_p\neg A_{r1}\neg A_pf$, $(\neg)A_p(\neg)A_{r1}A_pf$
A3-7	$\neg A_p\neg A_{r1}A_pf$	$\neg A_{r1}A_pf, A_pf$	$A_{r1}A_pf$, $\neg A_pf$	$(\neg)A_pA_{r1}A_p$, $(\neg)A_p(\neg)A_{r1}\neg A_pf$
A3-8	$\neg A_p\neg A_{r1}\neg A_pf$	$\neg A_{r1}\neg A_pf, \neg A_pf$	$A_{r1}\neg A_pf$, A_pf	$(\neg)A_pA_{r1}\neg A_p$, $(\neg)A_p(\neg)A_{r1}A_pf$
A4-1	$A_pA_{r1}A_{r2}f$	$A_{r1}A_{r2}f, A_{r2}f, A_pA_{r2}f$	$\neg A_{r1}A_{r2}f, \neg A_{r2}f, \neg A_pA_{r2}f$	$(\neg)A_p\neg A_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}\neg A_{r2}f$
A4-2	$A_pA_{r1}\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, \neg A_{r2}f, A_p\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, A_{r2}f, A_p\neg A_{r2}f$	$(\neg)A_pA_{r1}\neg A_{r2}f, A_p(\neg)A_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}\neg A_{r2}f$
A4-3	$A_p\neg A_{r1}A_{r2}f$	$\neg A_{r1}A_{r2}f, A_{r2}f, A_pA_{r2}f$	$A_{r1}A_{r2}f, \neg A_{r2}f, (\neg)A_p\neg A_{r2}f$	$(\neg)A_pA_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}\neg A_{r2}f$
A4-4	$A_p\neg A_{r1}\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, \neg A_{r2}f, A_p\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, A_{r2}f, \neg A_p\neg A_{r2}f$	$(\neg)A_pA_{r1}\neg A_{r2}f, (\neg)A_p(\neg)A_{r1}A_{r2}f, \neg A_p(\neg)A_{r1}\neg A_{r2}f$
A4-5	$\neg A_pA_{r1}A_{r2}f$	$A_{r1}A_{r2}f, A_{r2}f, \neg A_pA_{r2}f$	$\neg A_{r1}A_{r2}f, \neg A_{r2}f, A_pA_{r2}f$	$(\neg)A_p\neg A_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}\neg A_{r2}f, A_p(\neg)A_{r1}A_{r2}f$
A4-6	$\neg A_pA_{r1}\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, \neg A_{r2}f, \neg A_p\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, A_{r2}f, A_p\neg A_{r2}f$	$(\neg)A_p\neg A_{r1}\neg A_{r2}f, (\neg)A_p(\neg)A_{r1}A_{r2}f, A_p(\neg)A_{r1}\neg A_{r2}f$
A4-7	$\neg A_p\neg A_{r1}A_{r2}f$	$\neg A_{r1}A_{r2}f, A_{r2}f, \neg A_pA_{r2}f$	$A_{r1}A_{r2}f, \neg A_{r2}f, A_pA_{r2}f$	$(\neg)A_pA_{r1}A_{r2}f, (\neg)A_p(\neg)A_{r1}\neg A_{r2}f, A_p(\neg)A_{r1}A_{r2}f$
A4-8	$\neg A_p\neg A_{r1}\neg A_{r2}f$	$\neg A_{r1}\neg A_{r2}f, \neg A_{r2}f, \neg A_p\neg A_{r2}f$	$A_{r1}\neg A_{r2}f, A_{r2}f, A_p\neg A_{r2}f$	$(\neg)A_pA_{r1}\neg A_{r2}f, (\neg)A_p(\neg)A_{r1}A_{r2}f, A_p(\neg)A_{r1}\neg A_{r2}f$

Table A.4: Complete memory map of a priniple object M_p

	M_{su}					M_s					M_r			
$A1$	1	$A_{su}f$	2	$\neg A_{su}f$	$A1$	1	A_sf	2	$\neg A_sf$	$A1$	1	A_sf	2	$\neg A_sf$
$A2$	3	$A_{su}A_sf$	4	$\neg A_{su}A_sf$	$A2$	3	$A_sA_{su}f$	4	$\neg A_sA_{su}f$	$A2$	3	$A_rA_{su}f$	4	$\neg A_rA_{su}f$
	5	$A_{su}\neg A_sf$	6	$\neg A_{su}\neg A_sf$		5	$A_s\neg A_{su}f$	6	$\neg A_s\neg A_{su}f$		5	$A_r\neg A_{su}f$	6	$\neg A_r\neg A_{su}f$
	7	$A_{su}A_rf$	8	$\neg A_{su}A_rf$		7	A_sA_rf	8	$\neg A_sA_rf$		7	A_rA_sf	8	$\neg A_rA_sf$
	9	$A_{su}\neg A_rf$	10	$\neg A_{su}\neg A_rf$		9	$A_s\neg A_rf$	10	$\neg A_s\neg A_rf$		9	$A_r\neg A_sf$	10	$\neg A_r\neg A_sf$
$A3$	11	$A_{su}A_sA_{su}f$	12	$A_{su}A_s\neg A_{su}f$	$A3$	11	$A_sA_{su}A_sf$	12	$A_sA_{su}\neg A_sf$	$A3$	11	$A_rA_{su}A_rf$	12	$A_rA_{su}\neg A_rf$
	13	$A_{su}\neg A_sA_{su}f$	14	$A_{su}\neg A_s\neg A_{su}f$		13	$A_s\neg A_{su}A_sf$	14	$A_s\neg A_{su}\neg A_sf$		13	$A_r\neg A_{su}A_rf$	14	$A_r\neg A_{su}\neg A_rf$
	15	$\neg A_{su}A_sA_{su}f$	16	$\neg A_{su}A_s\neg A_{su}f$		15	$\neg A_sA_{su}A_sf$	16	$\neg A_sA_{su}\neg A_sf$		15	$\neg A_rA_{su}\neg A_rf$	16	$\neg A_rA_{su}\neg A_rf$
	17	$\neg A_{su}\neg A_sA_{su}f$	18	$\neg A_{su}\neg A_s\neg A_{su}f$		17	$\neg A_s\neg A_{su}A_sf$	18	$\neg A_s\neg A_{su}\neg A_sf$		17	$\neg A_r\neg A_{su}A_rf$	18	$\neg A_r\neg A_{su}\neg A_rf$
	19	$A_{su}A_rA_{su}f$	20	$A_{su}A_r\neg A_{su}f$		19	$A_sA_rA_sf$	20	$A_sA_r\neg A_sf$		19	$A_rA_sA_rf$	20	$A_rA_s\neg A_rf$
	21	$A_{su}\neg A_rA_{su}f$	22	$A_{su}\neg A_r\neg A_{su}f$		21	$A_s\neg A_rA_sf$	22	$A_s\neg A_r\neg A_sf$		21	$A_r\neg A_sA_rf$	22	$A_r\neg A_s\neg A_rf$
	23	$\neg A_{su}A_rA_{su}f$	24	$\neg A_{su}A_r\neg A_{su}f$		23	$\neg A_sA_rA_sf$	24	$\neg A_sA_r\neg A_sf$		23	$\neg A_rA_sA_rf$	24	$\neg A_rA_s\neg A_rf$
	25	$\neg A_{su}\neg A_rA_{su}f$	26	$\neg A_{su}\neg A_r\neg A_{su}f$		25	$\neg A_s\neg A_rA_sf$	26	$\neg A_s\neg A_r\neg A_sf$		25	$\neg A_r\neg A_sA_rf$	26	$\neg A_r\neg A_s\neg A_rf$
$A4$	27	$A_{su}A_sA_rf$	28	$A_{su}A_s\neg A_rf$	$A4$	27	$A_sA_{su}A_rf$	28	$A_sA_{su}\neg A_rf$	$A4$	27	$A_rA_{su}A_sf$	28	$A_rA_{su}\neg A_sf$
	29	$A_{su}\neg A_sA_rf$	30	$A_{su}\neg A_s\neg A_rf$		29	$A_s\neg A_{su}A_rf$	30	$A_s\neg A_{su}\neg A_rf$		29	$A_r\neg A_{su}A_sf$	30	$A_r\neg A_{su}\neg A_sf$
	31	$\neg A_{su}A_sA_rf$	32	$A_{su}A_s\neg A_rf$		31	$\neg A_sA_{su}A_rf$	32	$A_sA_{su}\neg A_rf$		31	$\neg A_rA_{su}A_sf$	32	$A_rA_{su}\neg A_sf$
	33	$\neg A_{su}\neg A_sA_rf$	34	$A_{su}\neg A_s\neg A_rf$		33	$\neg A_s\neg A_{su}A_rf$	34	$A_s\neg A_{su}\neg A_rf$		33	$\neg A_r\neg A_{su}A_sf$	34	$A_r\neg A_{su}\neg A_sf$
	35	$A_{su}A_rA_sf$	36	$A_{su}A_r\neg A_sf$		35	$A_sA_rA_{su}f$	36	$A_sA_r\neg A_{su}f$		35	$A_rA_sA_{su}f$	36	$A_rA_s\neg A_{su}f$
	37	$A_{su}\neg A_rA_sf$	38	$A_{su}\neg A_r\neg A_sf$		37	$A_s\neg A_rA_{su}f$	38	$A_s\neg A_r\neg A_{su}f$		37	$A_r\neg A_sA_{su}f$	38	$A_r\neg A_s\neg A_{su}f$
	39	$\neg A_{su}A_rA_sf$	40	$\neg A_{su}A_r\neg A_sf$		39	$\neg A_sA_rA_{su}f$	40	$\neg A_sA_r\neg A_{su}f$		39	$\neg A_rA_sA_{su}f$	40	$\neg A_rA_s\neg A_{su}f$
	41	$\neg A_{su}\neg A_rA_sf$	42	$\neg A_{su}\neg A_r\neg A_sf$		41	$\neg A_s\neg A_rA_{su}f$	42	$\neg A_s\neg A_r\neg A_{su}f$		41	$\neg A_r\neg A_sA_{su}f$	42	$\neg A_r\neg A_s\neg A_{su}f$

Table A.5: Exhaustive statespace of principle memory for uncertainty conflict where $r1$ is negated.

Label	Given $\alpha \in M_p$	Conflicting $\alpha' \in M_p$
A2-1	$A_pA_{r1}f$	$A_p\neg A_{r1}f$
A2-2	$A_p\neg A_{r1}f$	$A_pA_{r1}f$
A2-3	$\neg A_pA_{r1}f$	$\neg A_p\neg A_{r1}f$
A2-4	$\neg A_p\neg A_{r1}f$	$\neg A_pA_{r1}f$
A3-1	$A_pA_{r1}A_pf$	$A_p\neg A_{r1}A_pf$
A3-2	$A_pA_{r1}\neg A_pf$	$A_p\neg A_{r1}\neg A_pf$
A3-3	$A_p\neg A_{r1}A_pf$	$A_pA_{r1}A_pf$
A3-4	$A_p\neg A_{r1}\neg A_pf$	$A_pA_{r1}\neg A_pf$
A3-5	$\neg A_pA_{r1}A_pf$	$\neg A_p\neg A_{r1}A_pf$
A3-6	$\neg A_pA_{r1}\neg A_pf$	$\neg A_p\neg A_{r1}\neg A_pf$
A3-7	$\neg A_p\neg A_{r1}A_pf$	$\neg A_pA_{r1}A_pf$
A3-8	$\neg A_p\neg A_{r1}\neg A_pf$	$\neg A_pA_{r1}\neg A_pf$
A4-1	$A_pA_{r1}A_{r2}f$	$A_p\neg A_{r1}A_{r2}f$
A4-2	$A_pA_{r1}\neg A_{r2}f$	$A_p\neg A_{r1}\neg A_{r2}f$
A4-3	$A_p\neg A_{r1}A_{r2}f$	$A_pA_{r1}A_{r2}f$
A4-4	$A_p\neg A_{r1}\neg A_{r2}f$	$A_pA_{r1}\neg A_{r2}f$
A4-5	$\neg A_pA_{r1}A_{r2}f$	$\neg A_p\neg A_{r1}A_{r2}f$
A4-6	$\neg A_pA_{r1}\neg A_{r2}f$	$\neg A_p\neg A_{r1}\neg A_{r2}f$
A4-7	$\neg A_p\neg A_{r1}A_{r2}f$	$\neg A_pA_{r1}A_{r2}f$
A4-8	$\neg A_p\neg A_{r1}\neg A_{r2}f$	$\neg A_pA_{r1}\neg A_{r2}f$

Table A.6: Exhaustive illustration of uncertainty conflict with $r2$ negated.

Label	Given $\alpha \in M_p$	Given $\alpha' \in M_p$
A4-1	$A_pA_{r1}A_{r2}f$	$A_pA_{r1}\neg A_{r2}f$
A4-2	$A_pA_{r1}\neg A_{r2}f$	$A_pA_{r1}A_{r2}f$
A4-3	$A_p\neg A_{r1}A_{r2}f$	$A_p\neg A_{r1}\neg A_{r2}f$
A4-4	$A_p\neg A_{r1}\neg A_{r2}f$	$A_p\neg A_{r1}A_{r2}f$
A4-5	$\neg A_pA_{r1}A_{r2}f$	$\neg A_pA_{r1}\neg A_{r2}f$
A4-6	$\neg A_pA_{r1}\neg A_{r2}f$	$\neg A_pA_{r1}A_{r2}f$
A4-7	$\neg A_p\neg A_{r1}A_{r2}f$	$\neg A_p\neg A_{r1}\neg A_{r2}f$
A4-8	$\neg A_p\neg A_{r1}\neg A_{r2}f$	$\neg A_p\neg A_{r1}A_{r2}f$

Bibliography

[1] J. Bhatia and T. D. Breaux, “Empirical measurement of perceived privacy risk,” *ACM Trans. Comput.-Hum. Interact.*, vol. 25, no. 6, dec 2018. [Online]. Available: https://doi.org/10.1145/3267808

[2] X. Zhao, X. Zheng, X. Yang, X. Liu, and J. Tang, “Jointly learning to recommend and advertise,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, ser. KDD ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3319–3327. [Online]. Available: https://doi.org/10.1145/3394486.3403384

[3] S. Adshead, G. Forsyth, S. Wood, and L. Wilkinson, “Online advertising in the uk,” *PLUM Consulting report for the UK Department for Digital, Culture, Media and Sport*, 2019.

[4] M. Koop, E. Tews, and S. Katzenbeisser, “In-depth evaluation of redirect tracking and link usage,” *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 4, pp. 394–413, 2020. [Online]. Available: https://doi.org/10.2478/popets-2020-0079

[5] K. Varnali, “Online behavioral advertising: An integrative review,” *Journal of Marketing Communications*, vol. 27, no. 1, pp. 93–114, 2021.

[6] R. N. Zaeem and K. S. Barber, “The effect of the gdpr on privacy policies: Recent progress and future promise,” *ACM Trans. Manage. Inf. Syst.*, vol. 12, no. 1, dec 2020. [Online]. Available: https://doi.org/10.1145/3389685

[7] X. Hu and N. Sastry, “Characterising third party cookie usage in the eu after gdpr,” in *Proceedings of the 10th ACM Conference on Web Science*, ser. WebSci ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 137–141. [Online]. Available: https://doi.org/10.1145/3292522.3326039

[8] M. Hils, D. W. Woods, and R. Böhme, “Measuring the emergence of consent management on the web,” in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC ’20. New York, NY, USA: Association

for Computing Machinery, 2020, p. 317–332. [Online]. Available: https://doi.org/10.1145/3419394.3423647

[9] M. Nouwens, I. Liccardi, M. Veale, D. Karger, and L. Kagal, *Dark Patterns after the GDPR: Scraping Consent Pop-Ups and Demonstrating Their Influence*. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–13. [Online]. Available: https://doi.org/10.1145/3313831.3376321

[10] C. Utz, M. Degeling, S. Fahl, F. Schaub, and T. Holz, "(un)informed consent: Studying gdpr consent notices in the field." New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3319535.3354212

[11] T. H. Soe, O. E. Nordberg, F. Guribye, and M. Slavkovik, *Circumvention by Design - Dark Patterns in Cookie Consent for Online News Outlets*. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3419249.3420132

[12] E. Papadogiannakis, P. Papadopoulos, N. Kourtellis, and E. P. Markatos, "User tracking in the post-cookie era: How websites bypass gdpr consent to track users." New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3442381.3450056

[13] C. M. Gray, C. Santos, N. Bielova, M. Toth, and D. Clifford, "Dark patterns and the legal requirements of consent banners: An interaction criticism perspective," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3411764.3445779

[14] V. Bannihatti Kumar, R. Iyengar, N. Nisal, Y. Feng, H. Habib, P. Story, S. Cherivirala, M. Hagan, L. Cranor, S. Wilson, F. Schaub, and N. Sadeh, *Finding a Choice in a Haystack: Automatic Extraction of Opt-Out Statements from Privacy Policy Text*. New York, NY, USA: Association for Computing Machinery, 2020, p. 1943–1954. [Online]. Available: https://doi.org/10.1145/3366423.3380262

[15] M. Horák, V. Stupka, and M. Husák, "Gdpr compliance in cybersecurity software: A case study of dpia in information sharing platform," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ser. ARES '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3339252.3340516

[16] A. S. Ahmadian, D. Strüber, V. Riediger, and J. Jürjens, “Supporting privacy impact assessment by model-based privacy analysis,” in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ser. SAC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1467–1474. [Online]. Available: https://doi.org/10.1145/3167132.3167288

[17] M. E. Kaminski and G. Malgieri, “Multi-layered explanations from algorithmic impact assessments in the gdpr,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, ser. FAT* ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 68–79. [Online]. Available: https://doi.org/10.1145/3351095.3372875

[18] M. Humbert, B. Trubert, and K. Huguenin, “A survey on interdependent privacy,” *ACM Comput. Surv.*, vol. 52, no. 6, oct 2019. [Online]. Available: https://doi.org/10.1145/3360498

[19] D. Le Métayer, “Privacy by design: Formal framework for the analysis of architectural choices,” *CODASPY 2013 - Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*, pp. 95–104, 2013.

[20] G. Duncan, “Engineering: Privacy by design,” *Science*, vol. 317, no. 5842, pp. 1178–1179, 2007.

[21] A. Cavoukian, “Understanding How to Implement Privacy by Design, One Step at a Time,” *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 78–82, 2020.

[22] S. Sirur, J. R. C. Nurse, and H. Webb, “Are we there yet? Understanding the challenges faced in complying with the General Data Protection Regulation (GDPR),” no. iii, pp. 88–95, 2018. [Online]. Available: http://arxiv.org/abs/1808.07338

[23] N. Kersh and H. Toivainen, “Broad Research on Adult Education in the EU,” vol. 50, no. 693388, 2017.

[24] R. Layton, “Statement before the Senate Judiciary Committee On the General Data Protection Regulation and California Consumer Privacy Act: Opt-ins, Consumer Control, and the Impact on Competition and Innovation,” vol. 501, no. c, 2019. [Online]. Available: https://www.judiciary.senate.gov/imo/media/doc/LaytonTestimony1.pdf

[25] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” *Proceedings - International Conference on Data Engineering*, vol. 2006, p. 24, 2006.

[26] C. C. Aggarwal, “On k-anonymity and the curse of dimensionality,” in *Proceedings of the 31st International Conference on Very Large Data Bases*, ser. VLDB ’05. VLDB Endowment, 2005, p. 901–909.

[27] A. Friedman and A. Schuster, “Data mining with differential privacy,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 493–502. [Online]. Available: https://doi.org/10.1145/1835804.1835868

[28] C. Dwork, “Differential privacy: A survey of results,” in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19.

[29] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, “k-Anonymity,” *Advances in Information Security*, vol. 33, pp. 323–353, 2007.

[30] N. Holohan, D. J. Leith, and O. Mason, “Optimal differentially private mechanisms for randomised response,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2726–2735, 2017.

[31] S. L. Warner, “Randomized response: A survey technique for eliminating evasive answer bias,” *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965, pMID: 12261830. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1965.10480775

[32] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, “Our data, ourselves: Privacy via distributed noise generation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4004 LNCS, pp. 486–503, 2006.

[33] C. Dwork and G. N. Rothblum, “Concentrated differential privacy,” 2016.

[34] I. Mironov, “Rényi differential privacy,” in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, vol. 1, no. 1, 2017, pp. 263–275.

[35] L. Sweeney, “A model for protecting privacy 1,” *Ieee Security And Privacy*, vol. 10, no. 5, pp. 1–14, 2002.

[36] A. P. Dempster, “A generalization of bayesian inference,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 30, no. 2, pp. 205–232, 1968. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1968.tb00722.x

[37] C. A. Brodie, C.-M. Karat, and J. Karat, “An empirical study of natural language parsing of privacy policy rules using the sparcle policy workbench,” in *Proceedings of the second symposium on Usable privacy and security*, 2006, pp. 8–19.

[38] T. D. Breaux, H. Hibshi, and A. Rao, “Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements,” *Requirements Engineering*, vol. 19, no. 3, pp. 281–307, 2014.

[39] T. D. Breaux, D. Smullen, and H. Hibshi, “Detecting repurposing and over-collection in multi-party privacy requirements specifications,” in *2015 IEEE 23rd international requirements engineering conference (RE).* IEEE, 2015, pp. 166–175.

[40] X. Zhang, X. Wang, R. Slavin, T. Breaux, and J. Niu, “How does misconfiguration of analytic services compromise mobile privacy?” in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE).* IEEE, 2020, pp. 1572–1583.

[41] R. Slavin, X. Wang, M. B. Hosseini, J. Hester, R. Krishnan, J. Bhatia, T. D. Breaux, and J. Niu, “Toward a framework for detecting privacy policy violations in android application code,” in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 25–36.

[42] J. Caramujo, A. R. da Silva, S. Monfared, A. Ribeiro, P. Calado, and T. Breaux, “Rsl-il4privacy: a domain-specific language for the rigorous specification of privacy policies,” *Requirements Engineering*, vol. 24, no. 1, pp. 1–26, 2019.

[43] M. Olurin, C. Adams, and L. Logrippo, “Platform for privacy preferences (P3P): Current status and future directions,” *2012 10th Annual International Conference on Privacy, Security and Trust, PST 2012*, pp. 217–220, 2012.

[44] T. Yu, N. Li, and A. I. Antón, “A formal semantics for P3P,” *Proceedings of the 2004 Workshop on Secure Web Service, SWS 2004*, pp. 1–8, 2015.

[45] C. A. Ardagna, S. De Capitani Di Vimercati, S. Paraboschi, E. Pedrini, and P. Samarati, “An XACML-based privacy-centered access control system,” *Proceedings of the ACM Conference on Computer and Communications Security*, no. May 2014, pp. 49–57, 2009.

[46] A. S. Ahmadian, D. Strüber, and J. Jürjens, “Privacy-enhanced system design modeling based on privacy features,” *Proceedings of the ACM Symposium on Applied Computing*, vol. Part F1477, pp. 1492–1499, 2019.

[47] A. Barth, J. C. Mitchell, and J. Rosenstein, “Conflict and combination in privacy policy languages,” *WPES’04: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pp. 45–46, 2004.

[48] A. H. Anderson, “A comparison of two privacy policy languages: EPAL and XACML,” *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 53–60, 2006.

[49] N. M. Hoang and H. X. Son, “A dynamic solution for fine-grained policy conflict resolution,” *ACM International Conference Proceeding Series*, pp. 116–120, 2019.

[50] X. Xia, “A conflict detection approach for XACML policies on hierarchical resources,” *Proceedings - 2012 IEEE Int. Conf. on Green Computing and Communications, GreenCom 2012, Conf. on Internet of Things, iThings 2012 and Conf. on Cyber, Physical and Social Computing, CPSCom 2012*, pp. 755–760, 2012.

[51] Y. Wang, H. Xia, and Y. Huang, “Examining American and Chinese internet users’ contextual privacy preferences of behavioral advertising,” *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, vol. 27, pp. 539–552, 2016.

[52] F. Giunehiglia, R. Zhang, and B. Crispo, “RelBAC: Relation based access control,” *Proceedings of the 4th International Conference on Semantics, Knowledge, and Grid, SKG 2008*, no. May 2014, pp. 3–11, 2008.

[53] P. W. Fong, “Relationship-based access control: Protection model and policy language,” *CODASPY’11 - Proceedings of the 1st ACM Conference on Data and Application Security and Privacy*, pp. 191–201, 2011.

[54] A. S. Kayes, W. Rahayu, T. Dillon, E. Chang, and J. Han, “Context-aware access control with imprecise context characterization for cloud-based data resources,” *Future Generation Computer Systems*, vol. 93, pp. 237–255, 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.10.036

[55] H. Hu, G. J. Ahn, and J. Jorgensen, “Detecting and resolving privacy conflicts for collaborative data sharing in online social networks,” *ACM International Conference Proceeding Series*, pp. 103–112, 2011.

[56] H. Zhong, A. Squicciarini, and D. Miller, “Toward automated multiparty privacy conflict detection,” *International Conference on Information and Knowledge Management, Proceedings*, pp. 1811–1814, 2018.

[57] J. M. Such and N. Criado, “Multiparty privacy in social media,” *Communications of the ACM*, vol. 61, no. 8, pp. 74–81, 2018.

[58] J. M. Such and M. Rovatsos, “Privacy policy negotiation in social media,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 11, no. 1, 2016.

[59] J. M. Such, A. Espinosa, and A. García-Fornes, "A survey of privacy in multi-agent systems," *Knowledge Engineering Review*, vol. 29, no. 3, pp. 314–344, 2014.

[60] C. T. Do, N. H. Tran, C. Hong, C. A. Kamhoua, K. A. Kwiat, E. Blasch, S. Ren, N. Pissinou, and S. S. Iyengar, "Game theory for cyber security and privacy," *ACM Computing Surveys*, vol. 50, no. 2, pp. 30–37, 2017.

[61] K. Binmore and N. Vulkan, "Applying game theory to automated negotiation," *Netnomics*, vol. 1, no. 1, pp. 1–9, 1999.

[62] O. When, "Chapter 7 Logic , Games and Interaction."

[63] S. H. Schwartz, "An overview of the schwartz theory of basic values," *Online readings in Psychology and Culture*, vol. 2, no. 1, pp. 2307–0919, 2012.

[64] F. Mosca, J. M. Such, and P. McBurney, "Value-driven collaborative privacy decision making," *CEUR Workshop Proceedings*, vol. 2335, pp. 13–20, 2019.

[65] N. Kökciyan, N. Yaglikci, and P. Yolum, "An argumentation approach for resolving privacy disputes in online social networks," *ACM Transactions on Internet Technology*, vol. 17, no. 3, 2017.

[66] D. Keküllüoğlu, N. Kökciyan, and P. Yolum, "Strategies for privacy negotiation in online social networks," *Frontiers in Artificial Intelligence and Applications*, vol. 285, pp. 1608–1609, 2016.

[67] D. Kekulluoglu, N. Kokciyan, and P. Yolum, "Pre serving privacy as social responsibility in online social networks," *ACM Transactions on Internet Technology*, vol. 18, no. 4, 2018.

[68] intersoft consulting, "Gdpr principles relating to the processing of personal data." Available at https://gdpr-info.eu/art-5-gdpr/ (2022/04/18).

[69] ——, "Gdpr data protection by design and by default." Available at https://gdpr-info.eu/art-25-gdpr/ (2022/04/18).

[70] A. Cavoukian, "Privacy by design," 2009.

[71] intersoft consulting, "Gdpr data protection impact assessment." Available at https://gdpr-info.eu/art-35-gdpr/ (2022/04/18).

[72] ——, "Tasks of the data protection officer." Available at https://gdpr-info.eu/art-39-gdpr/ (2022/04/18).

[73] I. C. Office, “Data sharing code of practice.” Available at https://ico.org.uk/media/for-organisations/guide-to-data-protection/ico-codes-of-practice/data-sharing-a-code-of-practice-1-0.pdf (2022/04/18).

[74] S. Sannon, B. Stoll, D. DiFranzo, M. F. Jung, and N. N. Bazarova, ““i just shared your responses”: Extending communication privacy management theory to interactions with conversational agents,” *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. GROUP, Jan. 2020. [Online]. Available: https://doi.org/10.1145/3375188

[75] S. Petronio, “Communication privacy management theory: What do we know about family privacy regulation?” *Journal of Family Theory & Review*, vol. 2, no. 3, pp. 175–196, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1756-2589.2010.00052.x

[76] A. Chagrov, “Modal logic,” 1997.

[77] V. Hendricks, “Epistemic logic,” 2008.

[78] J.-J. C. Meyer and W. Van Der Hoek, *Epistemic logic for AI and computer science*. Cambridge University Press, 2004, vol. 41.

[79] T. Bolander and M. B. Andersen, “Epistemic planning for single- and multi-agent systems,” *Journal of Applied Non-Classical Logics*, vol. 21, no. 1, pp. 9–34, 2011. [Online]. Available: https://doi.org/10.3166/jancl.21.9-34

[80] H. Van Ditmarsch, W. van Der Hoek, and B. Kooi, *Dynamic epistemic logic*. Springer Science & Business Media, 2007, vol. 337.

[81] W. H. Holliday, “Epistemic logic and epistemology,” in *Introduction to Formal Philosophy*. Springer, 2018, pp. 351–369.

[82] J. Hintikka, “Individuals, possible worlds, and epistemic logic,” *Nous*, pp. 33–62, 1967.

[83] T. E. Zimmermann, “Free choice disjunction and epistemic possibility,” *Natural Language Semantics*, vol. 8, no. 4, 2000.

[84] M. Huemer, “Epistemic possibility,” *Synthese*, vol. 156, no. 1, 2007.

[85] E. Orłowska, “Kripke semantics for knowledge representation logics,” *Studia Logica*, vol. 49, no. 2, pp. 255–272, 1990.

[86] T. Williamson, “Gettier cases in epistemic logic,” *Inquiry*, vol. 56, no. 1, pp. 1–14, 2013. [Online]. Available: https://doi.org/10.1080/0020174X.2013.775010

[87] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic: Graph. Darst*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2002. [Online]. Available: https://books.google.co.uk/books?id=pbb_Asgoq0oC

[88] M. Cresswell, *Semantical Essays: Possible Worlds and their Rivals*, ser. Studies in Linguistics and Philosophy. Springer Netherlands, 2012. [Online]. Available: https://books.google.co.uk/books?id=orC1BwAAQBAJ

[89] B. Skyrms, "Possible worlds, physics and metaphysics," *Philosophical Studies*, vol. 30, no. 5, pp. 323–332, 1976.

[90] I. Omoronyia, U. Etuk, and P. Inglis, "A privacy awareness system for software design," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 10, pp. 1557–1604, 2019.

[91] H. Lee and C.-H. Cho, "Digital advertising: present and future prospects," *International Journal of Advertising*, vol. 39, no. 3, pp. 332–341, 2020. [Online]. Available: https://doi.org/10.1080/02650487.2019.1642015

[92] A. Karaj, S. Macbeth, R. Berson, and J. M. Pujol, "Whotracks. me: Monitoring the online tracking landscape at scale," *CoRR, abs/1804.08959*, 2018.

[93] T. Mandl, W. Thode, and J. Griesbaum, ""i would have never allowed it": User perception of third-party tracking and implications for display advertising," 05 2015.

[94] S. Schelter and J. Kunegis, "Tracking the trackers: A large-scale analysis of embedded web trackers," in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[95] D. Radcliffe, "The publisher's guide to ecommerce," 2019.

[96] ——, "The publisher's guide to ecommerce: Case studies," 2020.

[97] G. Chiandussi, M. Codegone, S. Ferrero, and F. E. Varesio, *Comparison of multi-objective optimization methodologies for engineering applications*. Elsevier Ltd, 2012, vol. 63, no. 5. [Online]. Available: http://dx.doi.org/10.1016/j.camwa.2011.11.057

[98] S. Deshpande, L. T. Watson, and R. A. Canfield, "Pareto front approximation using a hybrid approach," *Procedia Computer Science*, vol. 18, pp. 521–530, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.05.216

[99] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, "A survey of multi-objective sequential decision-making," *Journal of Artificial Intelligence Research*, vol. 48, no. February, pp. 67–113, 2013.

[100] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[101] Y. Meier, J. Schäwel, and N. C. Krämer, "The shorter the better? effects of privacy policy length on online privacy decision-making," *Media and Communication*, vol. 8, no. 2, pp. 291–301, 2020.

[102] J. Gluck, F. Schaub, A. Friedman, H. Habib, N. Sadeh, L. F. Cranor, and Y. Agarwal, "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)*, 2016, pp. 321–340.

[103] B. Andow, S. Y. Mahmud, W. Wang, J. Whitaker, W. Enck, B. Reaves, K. Singh, and T. Xie, "Policylint: investigating internal privacy policy contradictions on google play," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 585–602.

[104] F. J. Zuiderveen Borgesius, S. Kruikemeier, S. C. Boerman, and N. Helberger, "Tracking walls, take-it-or-leave-it choices, the gdpr, and the eprivacy regulation," *Eur. Data Prot. L. Rev.*, vol. 3, p. 353, 2017.

[105] B. Eggl, "Learning to walk a tightrope: Challenges dpos face in the day-to-day exercise of their responsibilities," *Journal of Data Protection & Privacy*, vol. 3, no. 1, pp. 69–81, 2019.

[106] P. Lambert, *The Data Protection Officer: Profession, Rules, and Role*. CRC Press, 2016.

[107] P. Ryan, M. Crane, and R. Brennan, "Design challenges for gdpr regtech," *arXiv preprint arXiv:2005.12138*, 2020.

[108] C. Tikkinen-Piri, A. Rohunen, and J. Markkula, "Eu general data protection regulation: Changes and implications for personal data collecting companies," *Computer Law & Security Review*, vol. 34, no. 1, pp. 134–153, 2018.