POLICYCOMP: Counterpart Comparison of Privacy Policies Uncovers Overbroad Personal Data Collection Practices

Lu Zhou[1, 2], Chengyongxiao Wei[2], Tong Zhu[2], Guoxing Chen[2]
Xiaokuan Zhang[3], Suguo Du[2], Hui Cao[2], and Haojin Zhu[2]
[1]*Xidian University, zhoulu@xidian.edu.cn*
[2]*Shanghai Jiao Tong University, {smash-wind, tongzhu, guoxingchen, sgdu, huicao, zhu-hj}@sjtu.edu.cn*
[3] *George Mason University, xiaokuan@gmu.edu*

Abstract

Since mobile apps' privacy policies are usually complex, various tools have been developed to examine whether privacy policies have contradictions and verify whether privacy policies are consistent with the apps' behaviors. However, to the best of our knowledge, no prior work answers whether the personal data collection practices (PDCPs) in an app's privacy policy are necessary for given purposes (*i.e.*, whether to comply with the principle of *data minimization*). Though defined by most existing privacy regulations/laws such as GDPR, the principle of *data minimization* has been translated into different privacy practices depending on the different contexts (*e.g.*, various developers and targeted users). In the end, the developers can collect personal data claimed in the privacy policies as long as they receive authorizations from the users.

Currently, it mainly relies on legal experts to manually audit the necessity of personal data collection according to the specific contexts, which is not very scalable for millions of apps. In this study, we aim to take the first step to automatically investigate whether PDCPs in an app's privacy policy are overbroad from the perspective of *counterpart comparison*. Our basic insight is that, if an app claims to collect much more personal data in its privacy policy than most of its counterparts, it is more likely to be conducting overbroad collection. To achieve this, POLICYCOMP, an automatic framework for detecting overbroad PDCPs is proposed. We use POLICYCOMP to perform a large-scale analysis on 10,042 privacy policies and flag 48.29% of PDCPs to be overbroad. We shared our findings with 2,000 app developers and received 52 responses from them, 39 of which acknowledged our findings and took actions (*e.g.*, removing overbroad PDCPs).

1 Introduction

To provide services, mobile apps will collect various types of personal data. Due to frequent privacy leakage reports and increasing consciousness to privacy of users, people are paying more attention to the personal data collection of apps [34, 36, 38, 42]. To regulate personal data collection, some privacy protection laws have been enacted, such as the General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), and Personal Information Protection Law of the People's Republic of China (PIPL). These laws require *mobile app* to clearly disclose any *personal data collection practice* (PDCP) and *clear purposes* for processing it, which should be written in the app's *privacy policy*.

Amos *et al.* curated and analyzed a dataset of millions of privacy policies, which revealed that privacy policies have become substantially longer and difficult to read [14]. Since most of the privacy policies are usually complex, it creates a situation in which the user inattentively clicks "yes" without a complete understanding of the privacy policy [23, 28, 30]. Therefore, various tools have been developed to help users understand PDCPs in the privacy policy easier, *e.g.*, by extracting PDCPs [13, 20] and analyzing the usage of PDCPs [24, 47]. Another line of research focuses on 1) examining whether a privacy policy is *logically sound* by detecting contradictions within it [15, 19, 43], and 2) verifying whether a privacy policy is *consistent* with the app's behaviors [16, 21, 37, 40, 49, 50].

However, to the best of our knowledge, no prior work answers **whether PDCPs in a privacy policy are necessary** according to the purposes for which they are processed, even if the privacy policy is *logically sound and consistent* with the app's behaviors. By "necessary", the principle of *data minimization* under GDPR states that "Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed" [4].

Answering the above question should face the difficulty of translating an ambiguous law into a clear boundary between "necessary" and "unnecessary". Privacy is a context-dependent concept [12]. In the context of an app's privacy policy, the context means different kinds of services and different personal data necessary for the services. Further, the user may have a perspective of what PDCPs are necessary different from the app developer, when he or she does not fully agree with the claimed purposes. Considering a gaming app that collects location data, the developer thinks the location

is "necessary" for marketing, while the users may not think so since the location has no impact on gameplay and marketing is out of their interests. This hinders the lawmakers from defining a clear and widely applicable privacy boundary on a wide range of apps. Typically, to access services provided by developers, users are likely to give consent to the PDCPs claimed in the privacy policies without actually reading the privacy policies [33]. This may open a door for **overbroad collection**, which means that the app developers claim more PDCPs in privacy policies than actually needed for desired services of users. When we communicated with the developer of a dictionary app (Package Name: com.plexx.xxx, installs: $1M+$), the developer even stated that "it is always better to err on the side of saying you collect more information than you actually do".

Due to the strictness and complexity of judging the necessity of PDCPs, it mainly relies on legal experts to manually audit each PDCP and draw legal conclusions by jointly considering app functionalities, business needs, compliance issues, liability management, *etc*, which is not scalable for millions of apps. Therefore, an automated tool is needed to preliminarily screen out overbroad PDCPs for the legal experts to review, before drawing legal conclusions on violations of *data minimization* requirements. The challenges are twofold:

- **The unclear purposes of PDCPs in privacy policies**: It is difficult to determine exact purposes for each PDCP since many privacy policies only specify purposes at the app level (*e.g.*, "we may use collected personal data for any purpose as below") or explain purposes using unclear language (*e.g.*, provide services). Hence, it is challenging to judge if a PDCP is acceptable.
- **The lack of detailed standards about what types of personal data are necessary to fulfill a purpose**: most existing privacy protection laws do not specify what types of personal data are necessary given a specific purpose. Hence, it is quite difficult, if not impossible, to directly determine if a PDCP for a given purpose in a privacy policy follows the principle of *data minimization*.

To solve these challenges, we propose PolicyComp, an automatic framework for the detection of overbroad PDCPs in privacy policies. PolicyComp tackles this through the concept of *counterpart comparison*: given the privacy policy of a target app, PolicyComp first obtains a set of apps with similar functionality, coined as *counterpart apps*. After that, it extracts and regularizes the PDCPs of the policies of the target app and its counterpart apps, and further computes a likelihood of being overbroad for each PDCP in the target app's privacy policy, based on whether the counterpart apps also claim to collect the same type of personal data.

The basic intuition behind PolicyComp is that the apps having similar functionalities/purposes share similar privacy contexts, which may lead to similar PDCPs. Therefore, it is possible to leverage the PDCPs in counterpart apps' privacy policies as the potential standards to judge overbroad PDCPs based on the following insight: a PDCP in the target app's privacy policy is more likely to be necessary if it is also in counterpart apps' privacy policies.

While PolicyComp aims to make the first attempt to flag overbroad PDCPs using counterpart comparison, it cannot indicate whether the flagged overbroad PDCPs and the claimed purposes are legitimate from a legal standpoint, which is very hard to achieve even with state-of-the-art NLP techniques. Whether the claimed purposes and the PDCPs are legitimate have to be determined by legal experts after jointly considering more context-dependent factors (*e.g.*, business needs and liability management).

Contributions. The contributions of our study include:

- **Formal definition and model.** We formally define *data minimization* and *overbroad PDCP*, and propose a solution to estimate overbroad likelihood of PDCPs by comparing them with PDCPs in the target app's counterpart apps. Moreover, overbroad PDCP analysis models, including risk classification and overbroad PDCP reasoning models, are introduced to provide guidance for follow-up exploration of the overbroad PDCPs. (Sec. 3)
- **An automatic framework, PolicyComp, for the detection of overbroad PDCPs in privacy policies.** We design and implement PolicyComp, a system that could automatically find top-k counterpart apps of a target app, extract PDCPs from their privacy policies, as well as analyze overbroad PDCPs based on the above models. (Sec. 4)
- **A large-scale measurement.** PolicyComp achieves a 76% F1-score based on a ground-truth dataset labeled by 3 Ph.D. students from the law school. From the analysis results on $10,042$ privacy policies of Android apps, we select $2,000$ apps and share our findings with their developers. We receive 52 responses from these developers, 39 of which acknowledge our findings (*e.g.*, removing these overbroad PDCPs). (Sec. 5)

Ethical considerations. 1) Similar to other studies [15, 29], we mainly use apps' descriptions and privacy policies in our experiments. All datasets (e.g., from recommendation websites and Google Play) used in our experiments are publicly available; 2) Since PolicyComp is not to directly draw legal conclusions, we partially anonymize the measurement results in this paper to avoid legal dispute. We have already shared our findings with corresponding developers to enhance the privacy protection of personal data.

The remainder of this paper is organized as follows. Sec. 2 introduces background and related work. Sec. 3 gives the formal definition and analysis model. Sec. 4 describes the design of PolicyComp, and Sec. 5 shows the large-scale experimental results. We then present some cases in Sec. 6. Sec. 7 and Sec. 8 present the limitation and discussion. Finally, we conclude this work in Sec. 9.

2 Background and Related Work

2.1 Background

Privacy protection laws. More and more laws have been drafted aiming to provide legal frameworks on how to collect/use personal data, including the three most influential laws: GDPR [4], CCPA [2], and PIPL (China) [7]. GDPR defines *personal data* as "any information that relates to an individual who can be directly or indirectly identified", and requires that "Personal data shall be collected for specified, explicit and legitimate purposes."

These privacy protection regulations/laws follow similar important principles for personal data collection: explicitly describing the *type(s)* of the collected personal data and the specific *purpose(s)* for each personal data, as well as following the principle of *data minimization*.

Data minimization. To guide personal data collection, GDPR introduces the principle of *data minimization*: "personal data shall be adequate, relevant and limited to what is *necessary* in relation to the purposes for which they are processed [4]." *Data minimization* enforces that a data controller (*e.g.*, developers) should limit the collection of personal data to what is directly relevant to fulfill a specific purpose.

Privacy policies of mobile apps. A privacy policy of a mobile app is a public/legal document that explains how the app processes personal data, including the collection/usage of any type of personal data (*i.e.*, PDCP) and how it follows data protection principles [11, 18]. If developers want to collect any personal data, they should consider the necessity of it and explicitly describe the specific purpose(s) of it in the app's privacy policy. After they received explicit authorizations from the users [32, 39], they could collect the types of personal data which is described in the app's privacy policy.

Natural language processing (NLP). NLP techniques are widely used to extract data collection, usage, sharing practices from privacy policies, enabling large-scale analysis of privacy policies. Some NLP techniques used in this paper include:

Part-of-speech (POS) tagging [34] is the process of tagging the part of speech of words. Particularly, this technique could tag sentences (Fig. 1 [middle]) and help determine sentences describing personal data collection by judging whether a sentence contains specific verbs (*e.g.*, "collect").

Dependency parsing [45] aims to analyze the structure of a sentence and construct relationships between words (Fig. 1 [top]), which could be used to find *data objects* (*i.e.*, potential personal data such as "name") that have syntactic dependencies on a collection verb.

Named-entity recognition (NER) is the task of tagging entities in a sentence with predefined categories, such as *data objects* in Fig. 1 [bottom]. Particularly, NER models could be trained for the privacy policy domain to accurately identify personal data in a sentence [15].

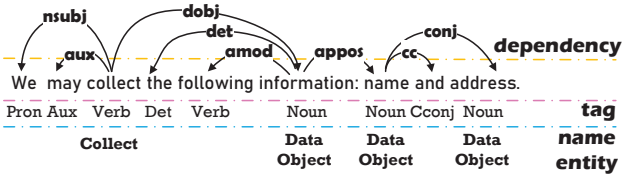


Figure 1: Sentence parsing

Semantic role labeling (SRL) aims to label the semantic roles of words (or phrases), such as the *purposes* in a sentence [21].

2.2 Related Work

2.2.1 Privacy Policy Understanding

Various tools have been developed to automatically extract privacy practices (*e.g.*, data collection/usage) from privacy policies, and enable question answering [24, 47]. While these studies focused on identifying the sentences/segments relevant to a privacy practice, some other studies aimed to extract fine-grained information. Ahmad *et al.* proposed PolicyIE, an English corpus spanning 31 privacy policies, which could be used to train models for extracting fine-grained personal data [13]. Bui *et al.* created a large annotated dataset from 30 privacy policies and presented a neural model-based automated system to extract fine-grained PDCPs [20]. Andow *et al.* also designed a tool to automatically generate personal data ontologies and extract fine-grained PDCPs from privacy policies [15], which provided a solid foundation for our work.

2.2.2 Personal Data Collection Analysis

Based on automatic understanding tools, studies have been carried out to analyze personal data collection. We organize related work according to the research object (*i.e.*, privacy policy, apps, or privacy $\times$ apps) and the target properties (*i.e.*, consistency or necessity) as shown in Table 1. To the best of our knowledge, our work is the first to study the *necessity* of personal data collection in *privacy policies*.

Privacy Policy + Consistency. As some privacy policies are written by developers who might be careless or with malicious intentions, contradictions of collection practices may exist in a privacy policy. For example, a privacy policy declares that it does collect "email address" in one place and later declares that it does not. Yu *et al.* identified contradictions between a privacy policy and policies of third-party libs [43]. Breaux *et al.* identified contradictions within a privacy policy and among multiple policies in a data supply chain [19]. And Andow *et al.* were the first to characterize and automatically analyze potential contradictions of sharing and collection practices within a privacy policy based on their automatic personal data extraction tool [15].

Table 1: Related work on personal data collection analysis.

	Consistency	Necessity
Privacy Policy	Internal Contradiction [15] First- vs. Third-party Contradiction [19, 43]	**Our work**
Privacy Policy × App	System Call-to-policy Consistency [37, 50] User Input-to-policy Consistency [40] Entity-sensitive Flow-to-policy Consistency [16, 49] Purpose-to-policy Consistency [21]	/
App	/	Reducing Permission Requests [25, 35] Detecting Privacy Disclosures [29]

Privacy Policy × App + Consistency. Besides checking consistency within privacy policies, studies are focusing on whether PDCPs of apps follow corresponding privacy policies strictly. Slavin *et al.* [37] and Zimmeck *et al.* [50] identified the used personal data from API calls and compared them with declared personal data in privacy policies. Especially, Zimmeck *et al.* distinguished between first and third-party practices. Wang *et al.* [40] extended the flow-to-policy inconsistency analysis to cover user input data. Andow *et al.* enhanced flow-to-policy inconsistency analysis by considering the data-receiving entity and proposing a formal analysis model [16]. Zimmeck *et al.* [49] evaluated inconsistency issues on millions of apps. Bui *et al.* further detected the inconsistencies between data-usage purposes stated in a privacy policy and the actual execution behavior of Android apps [21].

App + Necessity. Another line of research focuses on the necessity of permission requests and privacy disclosures in apps, *e.g.*, by comparing with their similar apps. Peddinti *et al.* designed an algorithmic mechanism to reduce permission requests in mobile apps [35]. Jana *et al.* identified the least privilege violation of Chrome extensions and applications from Google Play Store [25]. Lu *et al.* presented a system to detect suspicious privacy disclosures, which improved existing works by filtering out legitimate disclosures [29].

3 Problem Formulation

We firstly give a formal definition of *data minimization* and overbroad PDCP in Sec. 3.1. Then we propose a solution to estimate the likelihood of PDCPs being overbroad in Sec. 3.2. Lastly, we introduce a model for estimating the severity of overbroad PDCPs and reasoning about why overbroad PDCPs occur in Sec. 3.3. Table 2 lists some important symbols.

3.1 Data Minimization and Overbroad PDCP

Data Minimization. Generally, *data minimization* indicates the minimal set of personal data required for fulfilling a given purpose. Let $\mathcal{R} = \{r_1, r_2, \dots\}$ denote the set of all purposes an app might have (*e.g.*, authentication). Let $\mathcal{D} = \{d_1, d_2, \dots\}$ denote all types of personal data an app might collect (*e.g.*, phone number). Further, let $2^{\mathcal{R}}$ and $2^{\mathcal{D}}$ denote the power sets of $\mathcal{R}$ and $\mathcal{D}$, respectively. The power set of a given set is a set that consists of the given set's all subsets. Considering that *data minimization* specifications might vary under different laws, we define *data minimization* as follows:

Table 2: Symbols and descriptions

Symbol	Description
$\mathcal{R}$	the set of all purposes an app might have
$\mathcal{D}$	the set of all types of personal data an app might collect
Ω	a privacy protection law
$\mathsf{M}_\Omega(r_j)$	*data minimization*: the set of necessary types of personal data for serving purpose $r_j \in \mathcal{R}$
C_P	PDCPs in a privacy policy P
$\mathsf{D}(P)$	all types of collected personal data in P
$\mathsf{R}(P)$	all purposes of PDCPs in P
$\mathsf{L}(d_i)$	the overbroad likelihood in collecting $d_i \in \mathcal{D}$
α	a threshold for determining an overbroad collection
$\mathcal{S}_\Omega$	types of highly protected personal data expressly stated under Ω
$\mathsf{E}(d_i, P)$	potential justifications for collecting d_i

Definition 1. ***Data Minimization:*** *Data minimization under a privacy protection law Ω is a function M_Ω:*

$$\mathsf{M}_\Omega : \mathcal{R} \to 2^{\mathcal{D}} \tag{1}$$

which takes in a purpose $r_j \in \mathcal{R}$ as input and outputs a set $\mathsf{M}_\Omega(r_j) \in 2^{\mathcal{D}}$ containing the necessary types of personal data for serving the purpose r_j.

Overbroad PDCP. Before we model overbroad PDCP, we need to model the personal data collection specifications in a privacy policy. Ideally, a privacy policy describes each type of personal data to be collected and the corresponding purpose(s) for the collection. Therefore, PDCPs in a privacy policy P, denoted as C_P, can be modeled as a subset of $\mathcal{D} \times 2^{\mathcal{R}}$. Each element $(d_i, R) \in C_P$ indicates the privacy policy's claim of collection of personal data d_i for a set $R \in 2^{\mathcal{R}}$ of purposes.

On the other hand, if there exists a purpose $r_j \in R$ such that d_i is not within the set $\mathsf{M}_\Omega(r_j)$, that is, d_i is not the necessary personal data for the purpose r_j, an overbroad PDCP is found. Therefore, the overbroad PDCP can be defined as follows:

Definition 2. ***Overbroad PDCP:*** *Given the data minimization function M_Ω under a privacy protection law Ω and the personal data collection practices C_P for privacy policy P, the collection of the types of personal data in the following set is considered overbroad PDCPs:*

$$\{d_i | (d_i,\ R) \in C_P,\ and\ \exists\ r_j \in R : d_i \notin \mathsf{M}_\Omega(r_j)\} \qquad (2)$$

3.2 A Solution to Estimate overbroad PDCP

Determining overbroad PDCPs is especially challenging due to the existence of unclear purposes of PDCPs described in privacy policies. Particularly, we observed that the purposes presented in privacy policies could be written in the following two ways: 1) using a separate section to explain the purposes of all collected data, such as "We may use collected personal data for any purpose as below ...", making it difficult to link the exact purpose(s) to each PDCP; 2) using unclear language to describe purposes, such as "We may use your personal data to develop new services" (it is unclear what the 'services' are or how the collected data could help develop them) [5].

Another challenge lies in lacking detailed standards of *data minimization* as most regulations/laws do not clearly define how to meet *data minimization* (*i.e.*, what types of personal data are necessary to fulfill a purpose), but authorize the controllers (*e.g.*, developers) to collect what they think is "necessary" due to the diversity of purposes/services.

To solve these challenges, we propose to estimate overbroad PDCPs through *counterpart comparison*. The intuition is that the personal data collected by the majority of apps fulfilling similar purposes, dubbed as *counterpart apps*, is more likely to be necessary for those purposes, providing potential standards for determining overbroad PDCPs.

Our work leverages the PDCPs in counterpart apps' privacy policies as the standards to signal potential violations of the above formal definition. Due to the limitations of automatic tools, drawing legal conclusions on violations of *data minimization* should still be determined by legal experts.

3.2.1 Overbroad Likelihood

With the idea of counterpart comparison, we propose to estimate the likelihood of a PDCP being overbroad, dubbed as *overbroad likelihood*, by comparing PDCPs in privacy policies of counterpart apps. We assume all types of collected personal data claimed in a privacy policy are used to fulfill all purposes described in that privacy policy. Let $\mathsf{D}(P) = \{d_i | (d_i, R) \in C_P\}$ and $\mathsf{R}(P) = \bigcup_{(d_i,R)\in C_P} R$ denote all types of collected personal data and all purposes of the personal data collection practices C_P in a privacy policy. For a privacy policy P, we collect the policies $P_1, \ldots, P_k$ of apps with similar purposes (we will describe how to find such apps in Sec. 4) to calculate the overbroad likelihood, which is defined as follows:

Table 3: The classification of PDCPs

Criteria	Risk level	Category
$\mathsf{L}(d_i) > \alpha$ & $d_i \in \mathcal{S}_\Omega$	High	Overbroad collection of *Class-I personal data*
$\mathsf{L}(d_i) > \alpha$ & $d_i \notin \mathcal{S}_\Omega$	Medium	Overbroad collection of *Class-II personal data*
$\mathsf{L}(d_i) \leq \alpha$	Low	Mostly agreed personal data collection

Definition 3. ***Overbroad Likelihood:*** *Given a privacy policy P of an app and the policies $P_1, \ldots, P_k$ of k counterpart apps, for each type of collected personal data $d_i \in \mathsf{D}(P)$, the overbroad likelihood in collecting d_i is defined as the ratio of policies in $P_1, \ldots,\ P_k$ that do not collect d_i:*

$$\mathsf{L}(d_i) = \frac{1}{k} \sum_{m=1,\ldots,\ k} \begin{cases} 1, & if\ d_i \notin \mathsf{D}(P_m) \\ 0, & otherwise \end{cases} \qquad (3)$$

3.3 Overbroad PDCP Analysis Model

After estimating the overbroad likelihood of PDCPs, we further classify them into three risk levels based on the severity and propose an overbroad PDCP reasoning model to promote follow-up exploration of why overbroad PDCP occurs. We just propose a feasible classification method for PDCPs, which can be changed according to different needs.

3.3.1 Risk Classification

To show the severity of overbroad PDCPs, we classify overbroad PDCPs into three risk levels. Particularly, we introduce (1) a threshold α such that collecting personal data d_i is considered overbroad if $\mathsf{L}(d_i) > \alpha$ (*e.g.*, $\alpha = 0.5$ corresponds to the majority principle); (2) a subset $\mathcal{S}_\Omega \subset \mathcal{D}$ representing highly protected personal data expressly stated under a privacy protection law Ω, such as health and biometric data under GDPR. For convenience, in the rest of paper, we call such highly protected personal data as *Class-I personal data*, and the other types of personal data in $\mathcal{D} \setminus \mathcal{S}_\Omega$ as *Class-II personal data*. Table 4 shows an example of *Class-I personal data* under GDPR. As shown in Table 3, we classify PDCPs into the following three levels:

High Risk Level: Overbroad collection of *Class-I personal data*. When the overbroad likelihood of d_i is beyond the threshold, *i.e.*, $\mathsf{L}(d_i) > \alpha$, and the type of personal data is *Class-I personal data*, *i.e.*, $d_i \in \mathcal{S}_\Omega$, we define the collection of d_i as *high-risk* collection since it is an overbroad collection of highly protected personal data expressly stated under law.

For example, considering the following sentence from a stock trading app:

– "In the exceptional circumstance that we collect special category information (information about your health, sexual orientation, racial or ethnic profile, political opinions, philosophical beliefs or biometric data), we will treat it with extra care."

This app uses vague statements to describe the collection of sensitive personal data without stating the specific purposes. As a stock trading app, the necessity of such collected personal data may be questionable given the provided services. This app has deleted these claims after we shared our findings with the developer. Compared with other personal data, the processing of sensitive personal data is more likely to be overbroad, especially when no specific purposes are stated [22].

Medium Risk Level: Overbroad collection of *Class-II personal data*. When the likelihood of overbroad collection of d_i is beyond the threshold, *i.e.*, $\mathsf{L}(d_i) > \alpha$, but the type of personal data d_i is not highly protected personal data expressly stated under law, *i.e.*, $d_i \notin \mathcal{S}_\Omega$, we define the collection of d_i as *medium-risk* collection. This does not mean that *Class-II personal data* is not important to users. Compared to *Class-I personal data*, *Class-II personal data*, *i.e.*, phone number, age, and gender, are widely used by apps to provide services, causing users to be accustomed to providing them for obtaining better services while ignoring potential privacy risks. However, when most counterpart apps do not require such personal data, users need to be cautious before providing it.

For example, considering the following sentence from a calculator app:

–"Information you provide when you register for the Services, such as *name, home or work addresses, e-mail address, telephone and fax numbers, and birth date*."

Different from its counterpart apps, this app collects physical address, phone number, and birth date, and claims such personal data is used for registration. Since these personal data are widely used across different apps, users may ignore the legitimacy of collecting these types of personal data when providing it to this calculator app.

Low Risk Level: Mostly agreed personal data collection. When the likelihood of overbroad collection of d_i is below the threshold, *i.e.*, $\mathsf{L}(d_i) \leq \alpha$, the collection of personal data is agreed by the developers of most counterpart apps. Thus, it is considered as *low-risk*.

3.3.2 Overbroad PDCP Reasoning Model

For high-risk and medium-risk PDCPs, we further analyze why overbroad PDCPs occur by presenting an overbroad PDCP reasoning model. The idea is to check whether the target app's privacy policy provides additional purposes (coined as *justifications*) for collecting d_i, compared with the purposes specified in the privacy policies of the counterpart apps that do not collect d_i.

Particularly, for overbroad collection of d_i in a privacy policy P, we collect sentences in P that describe d_i, denoted by $P(d_i)$, from which we try to extract purposes for collecting d_i, denoted by $\mathsf{R}(P(d_i))$. We then filter $\mathsf{R}(P(d_i))$ by removing purposes that exist in the counterpart apps' privacy policies

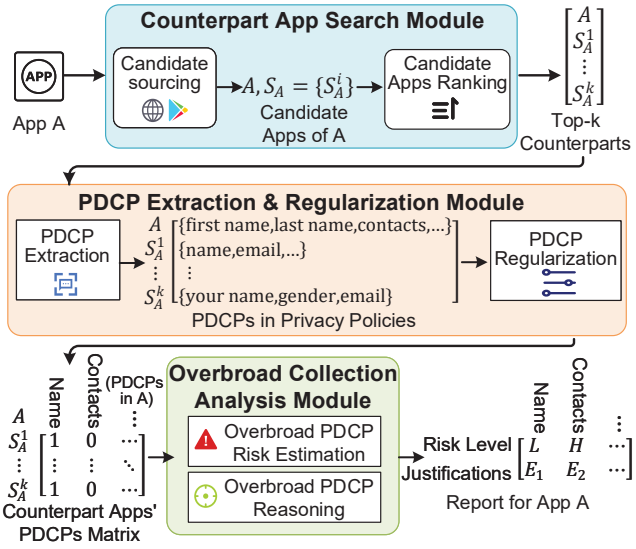


Figure 2: Workflow of POLICYCOMP

which do not collect d_i. The remaining purposes are considered *justifications* in privacy policy P for collecting d_i. We define the overbroad collection *justifications* as follows:

Definition 4. ***Overbroad Collection justifications:*** *Given a privacy policy P of an app and the policies $P_1, \ldots, P_k$ of k counterpart apps, for each type of collected personal data $d_i \in \mathsf{D}(P)$ with a high or medium risk level, the purposes in the following set are considered justifications for the collection:*

$$\begin{aligned} \mathsf{E}(d_i, P) &= \mathsf{R}(P(d_i)) \setminus \bigcup_{d_i \notin \mathsf{D}(P_m), m=1,\ldots,k} \mathsf{R}(P_m) \\ &= \{r_j \in \mathsf{R}(P(d_i)) \mid r_j \notin \bigcup_{d_i \notin \mathsf{D}(P_m), m=1,\ldots,k} \mathsf{R}(P_m)\} \end{aligned} \tag{4}$$

Note that even with justifications (*i.e.*, a non-empty $\mathsf{E}(d_i, P)$) , an overbroad PDCP is not considered excusable. The justifications provided by the reasoning model are mainly used to facilitate legal experts to determine whether the PDCP is acceptable (necessary).

4 POLICYCOMP: Privacy Policy Comparison System

In this section, we propose POLICYCOMP, an automatic framework for analyzing overbroad PDCPs in the privacy policies following the analysis model introduced in Sec. 3.3.

4.1 Design

As shown in Fig. 2, POLICYCOMP works as follows: A *counterpart App Search Module* takes as input the identifier of a target app A (*e.g.*, a package name: com.xxx.xxx) whose privacy policy will be analyzed, and outputs the identifiers of

k counterpart apps of A. A *PDCP Extraction and Regularization Module* then extracts and regularizes the PDCPs in the privacy policies of A and its counterpart apps. Lastly, an *Overbroad Collection Analysis Module* estimates the risk level of each PDCP in app A's privacy policy and further analyzes potential justification for those high/medium-risk PDCPs.

4.1.1 Counterpart App Search Module

This module aims to identify k counterpart apps of target app A from existing apps. The difficulties include: (1) it is impractical to examine all existing apps for counterpart app search; (2) how to define the similarity between an existing app and A. Two steps are introduced to tackle these two difficulties:

Candidate sourcing. Note that various sources exist to recommend related apps to users. For example, app stores usually provide a list of "similar" apps for a given app. Therefore, the first step of *Counterpart Apps Search Module*, named *Candidate sourcing*, is to collect A's related apps as candidate apps S_A from such sources (only identifiers are recorded).

Semantic similarity-based ranking. Since apps usually come with descriptions that explain the features/purposes of the apps to users [41], the second step of *counterpart Apps Search Module* uses the semantic similarities between the descriptions of A and the candidate apps S_A to rank the candidate apps. The larger the semantic similarity, the more likely that the candidate app achieves similar purposes to the target app. After ranking, this step outputs the top-k candidate apps as counterpart apps.

4.1.2 PDCP Extraction and Regularization Module

With A and its k counterpart apps, this module extracts the types of collected personal data (*i.e.*, PDCPs) in their privacy policies and regularizes them for subsequent comparison.

PDCP extraction. PDCPs in privacy policies usually follow specific patterns, such as "We may collect xxx". Therefore, in the first step, *PDCP extraction*, each sentence in a given privacy policy is parsed into POS tags, dependencies, and named entities (as shown in Fig. 1). If one collection verb (*e.g.*, "collect") is found, data objects with syntactic dependency on the collection verb(s) in this sentence are considered potential PDCPs.

PDCP regularization: The extracted PDCPs might be diverse (*e.g.*, using alternative terms "address book", "contact list", or "your contacts") and noisy (*e.g.*, "different types of information" may be extracted from the sentence "We collect different types of information from users", which does not represent any specific PDCP). Therefore, regularization needs to be performed. In the second step, *PDCP regularization*, fuzzing match (a technique that helps identify two approximately similar strings) is used to map each potential PDCP to a manually defined personal data ontology which is extracted from privacy protection laws.

4.1.3 Overbroad Collection Analysis Module

With the regularized PDCPs from the privacy policies of A and its k counterpart apps, this module is introduced to calculate the overbroad likelihood of each PDCP of A, and conduct further analysis following the models defined in Sec. 3.3.

Overbroad PDCP risk estimation: This step is to calculate the overbroad likelihood $\mathsf{L}(d_i)$ for each PDCP d_i in A's privacy policy according to Eq. (3). Based on the overbroad likelihood and whether the PDCP is *Class-I personal data* (*i.e.*, highly protected personal data expressly stated under law), each PDCP is classified into one of the three risk levels defined in Sec. 3.3.1: high-risk, medium-risk, or low-risk.

Overbroad PDCP reasoning: The last step, *overbroad PDCP reasoning*, aims to identify potential justifications (additional purposes) for each overbroad PDCP d_i. Particularly, for each overbroad PDCP d_i, sentences that describe d_i in the privacy policies of A are labeled using NLP techniques and purpose phrases could be extracted. All purposes of each counterpart app could also be extracted from its privacy policy using similar procedures. The overbroad collection justifications could then be identified following Eq. (4).

4.2 Implementation

In this subsection, we detail our prototype implementation of PolicyComp, in which we leverage state-of-the-art NLP techniques for multiple steps and customize them to achieve our goals. We will keep upgrading PolicyComp in the future when more advanced NLP techniques are available.

Candidate sourcing. In addition to sources used in existing work [26, 29, 35], such as Google Play's "similar apps", we also crawled popular crowdsourced alternative app recommendation websites (AlternativeTo[1], Top Best Alternatives[2], and Games Like[3]) and merged the results for better coverage. Given the identifier of a target app, this step outputs a list of identifiers of candidate apps.

Semantic similarity-based ranking. We leveraged the tool developed by Jiang *et al.* [26] to calculate the semantic similarities between the descriptions of A and the candidate apps. During ranking, whenever multiple candidate apps are from the same developer, only the one with the largest similarity is kept, ensuring the diversity of counterpart apps. After ranking, the top k candidate apps are kept as counterpart apps.

PDCP extraction. We customized the tool proposed by Andow *et al.* [15] in our implementation. Besides the existing "collection verbs" pattern, we added a new search pattern named "collection verbs + include" pattern that worked well for cases that the specific PDCPs are described in the sentence right after the sentence containing the collection verb(s), *e.g.*,

[1] https://alternativeto.net/
[2] https://www.topbestalternatives.com/
[3] https://www.moregameslike.com/

Table 4: Classification of personal data types

Class	Types
I	Race, Political opinions, Religious view, Trade union membership, Genetic data, Biometric data, Health, Sex life, Sexual orientation
II	SSN, Passport number, Driver's license number, State identification card number
	Payment information
	Contact log, Calendar, Contact list
	Precise location, Coarse location
	Name, Age, Date of birth, Gender
	Phone number, Email address, Physical address
	Education information, Professional information
	Device identifier, IP address, Browsing and search history, Purchasing history
	Audio, Photo

"We will collect personal data from you. This may include your postal address, e-mail address...". Our statistics from 10,042 privacy policies suggest that our new search pattern discovers 8.95% more collection sentences.

PDCP regularization. Since the existing automatic method for generating personal data ontology are not accurate enough due to the developers' irregular writing [15], we first manually defined the personal data ontology by extracting types of common personal data from privacy protection laws, as shown in Table 4. For clear expression, we define *Class-I personal data* based on highly protected personal data expressly stated under GDPR in this paper, which can be easily extended to other laws according to different needs. We then integrated the defined personal data ontology with the synonym defined by Andow *et al.* [15] for personal data matching and regularization (*e.g.*, your address book → contact list). Since most privacy policies just claim collect "location", we distinguish "precise location" from "coarse location" by checking whether an app requests "precise location" permission (listed in Google Play) [44]. If "precise location" permission is requested, both "precise location" and "coarse location" are considered to be collected.

Overbroad PDCP risk estimation. For each extracted and regularized PDCP of the target app, we calculated the overbroad likelihood following Eq. (3) and determined the risk level according to Table 3.

Overbroad PDCP reasoning. We implemented the tool proposed by Bui *et al.* [21] to extract purpose phrases (*e.g.*, "to verify your identity and prevent fraud") from sentences that describe the collection of d_i, which is based on the SRL technique. Then we decomposed them into uncompounded purposes (represented by a predicate-object pair, *e.g.*, [verify, identity] and [prevent, fraud]) to obtain single purpose and remove redundant information, which could facilitate the comparison of the purposes of the target app and counterpart apps. After that, we identified the overbroad PDCP justifications based on Eq. (4). When conducting comparison, some general purposes, including "provide service", "personalize service", "improve service", "support service" and "develop service" will not be removed even if its counterpart apps that do not collect d_i also claim these general purposes. This is because the scope of "service" in these apps are close but not identical.

5 Evaluation

In this section, we first describe the dataset to be evaluated and the performance of PolicyComp. Then, we provide a large-scale analysis of Android apps and present our findings. It is possible to extend PolicyComp to focus on a specific type of app by re-tweaking parameters.

5.1 Dataset Collection

Similar to [29], we started to collect target apps whose privacy policies will be analyzed by crawling three alternative app recommendation websites (AlternativeTo, Top Best Alternatives, and Games Like) since they come with lists of alternative apps recommended by users. We kept those apps that were also available on Google Play (*i.e.*, with package names) where we could obtain detailed information (*e.g.*, descriptions, developer name, and categories) and privacy policies. Apps that did not have privacy policies available on Google Play (*e.g.*, no/wrong download links, non-English policies) were excluded, resulting in 10,042 target apps, 72.85% of which have over 100,000 downloads on Google Play.

For each target app, its candidate apps consist of both *alternative apps* from the three alternative app recommendation websites and *similar apps* from Google Play. We also kept only those apps that were present and had privacy policies available on Google Play. Additionally, we removed candidate apps that did not fall in the same app category as the target app since these apps are likely recommended due to reasons other than achieving similar purposes, resulting in 30,281 distinct candidate apps in total. The number of candidate apps for a given target app ranges from 3 to 58. The distribution of the number of candidate apps is shown in Fig. 3.

5.2 End-to-end Identification of Overbroad PDCPs

Evaluation Metrics. We evaluated PolicyComp's end-to-end identification (*i.e.*, inputting the identifier of a target app A and automatically outputting overbroad PDCPs of A) of overbroad PDCPs in privacy policies. Given a high/medium-risk PDCP, subsequent manual audit by legal experts is required for this PDCP to draw a legal conclusion.

Ground-truth Dataset Creation. We randomly selected privacy policies of 300 target apps to be independently annotated

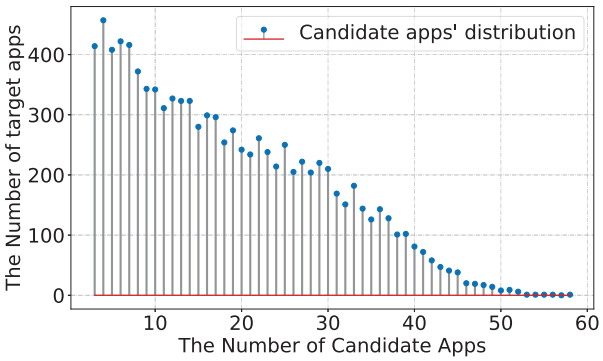


Figure 3: Distribution of the number of candidate apps

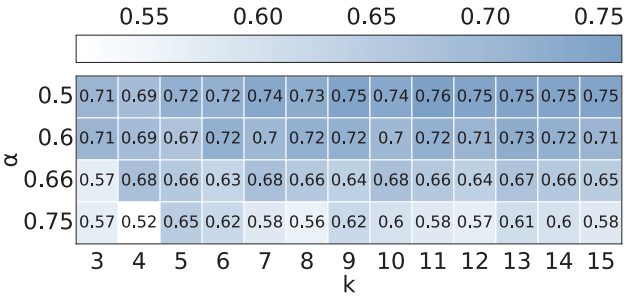


Figure 4: Performance (F1-scores) of overbroad PDCPs identification under different parameters

by 3 annotators (3 Ph.D. students from the law school working on the aspect of privacy protection laws) as follows:

Since there are no clear legal standards, we mainly rely on the experience of annotators to determine whether a PDCP is overbroad. When conducting annotation, 3 annotators were shown the detail information (e.g., app's description, privacy policy, PDCPs, and corresponding sentences) of the 300 target apps (data format is shown in Appendix A). These annotators also referred to enforcement cases[1] where the principle of *data minimization* was violated. Firstly, they read an app's description to understand its functionalities. The annotators could also use any publicly information or download the app to fully understand its functionalities. Then they determined whether a PDCP was overbroad based on the app's functionalities and the purposes claimed in the privacy policy (*e.g.*, from sentences describing the PDCP).

For each PDCP extracted from a selected privacy policy, the annotators assigned a *positive* label if they considered that the PDCP was potentially unnecessary (overbroad), or a *negative* label if they considered that the PDCP was necessary. The labeled results of the 3 annotators were merged based on the majority principle, i.e., at least two annotators reached an agreement. The resulting ground-truth dataset consists of $1,224$ positive labels and $1,186$ negative labels.

Experimental Configurations. We used the ground-truth dataset to evaluate the performance of PolicyComp on overbroad PDCPs identification under different parameters (220 apps was used after data cleaning). Since nearly half (47.18%) of target apps have less or equal to 15 candidate apps (as shown in Fig. 3), we conducted the evaluation with different values of k ranging from 3 to 15. As for the threshold α which is used to distinguish overbroad (high/medium-risk) PDCPs from low-risk PDCPs, we adopted the majority rule ($\alpha = \frac{1}{2}$) and common supermajorities, *i.e.*, three-fifths ($\alpha = \frac{3}{5}$), two-thirds ($\alpha = \frac{2}{3}$), and three-quarters ($\alpha = \frac{3}{4}$) [9].

[1]For example, a bank was punished since it collected users' biometric signatures for concluding electronic contracts [10]; a Belgian merchant was fined for collecting electronic identity cards to introduce a loyalty system [3].

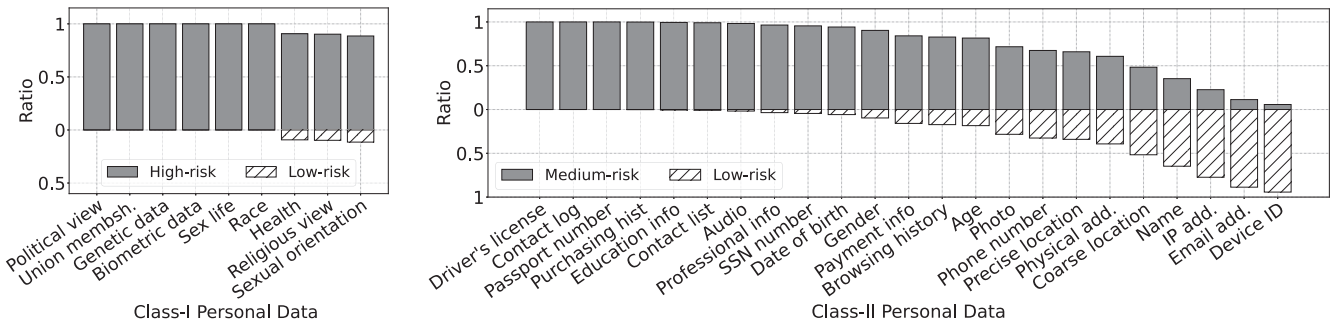
Evaluation Results. Since this work is the first to study the necessity of PDCPs, we prefer to jointly consider both the precision and recall by using F1-score when choosing parameters for the large-scale analysis. The F1-scores under different k and α ranged from 0.52 to 0.76, as shown in Fig. 4. The highest F1-score is achieved when $k = 11$ (70.04% of target apps have more than or equal to 11 candidate apps) and $\alpha = \frac{1}{2}$. The corresponding precision and recall are 0.70 and 0.82 respectively. We further investigate false positives (necessary PDCPs that PolicyComp incorrectly flags as overbroad) and false negatives (overbroad PDCPs that PolicyComp incorrectly flags as necessary) and summarize potential reasons as follows:

For false positives, the potential reasons may include: a) PDCPs for special functionalities in an app are more likely to be identified as overbroad when its counterparts fail to provide these functionalities. b) When a PDCP in counterpart app's privacy policy is not extracted due to the limitations of existing NLP tools [20], the PDCP's overbroad likelihood in the target app would be higher than expectation, leading to a case of false positive.

For false negatives, the potential reasons may include: a) When counterpart apps have the same overbroad PDCPs as the target app, those PDCPs' overbroad likelihoods would stay low, causing PolicyComp to consider them as necessary; b) When a PDCP in counterpart apps' privacy policies is incorrectly extracted (*i.e.*, the PDCP is not claimed to be collected), the PDCP's overbroad likelihood in the target app would be decreased.

It is important to point out that the false positives/negatives are expected to be continuously reduced by adopting more advanced counterpart app search algorithms as well as NLP analysis algorithms.

We also evaluate the performance of the elements of the pipeline (*i.e.*, counterpart app search and PDCP extraction and regularization) under different parameter settings: For k ranging from 3 to 15, the average similarity between target apps and their top-k counterpart apps ranges from 2.72 to 2.51 when evaluated using a scale from 0 to 3 (0-not similar,



(a) *Class-I personal data*

(b) *Class-II personal data*

Figure 5: The ratios of overbroad and low-risk PDCPs (Table version is added in the Appendix B)

Table 5: The overall results of overbroad PDCP analysis on 10,042 target apps

	PDCP Risk Estimation	Overbroad PDCP Reasoning	
		PDCPs with justifications	*PDCPs without justifications*
High-risk	871 (1.5%)	184 (21.13%)	687 (78.87%)
Medium-risk	27,132 (46.79%)	9,647 (35.56%)	17,485 (64.44%)
Low-risk	29,990 (51.71%)	/	

1-a little similar, 2-similar, 3-very similar). Especially, the average similarity reaches 2.62 when $k = 11$. POLICYCOMP achieves an overall 89.6% precision for extracting and regularizing PDCPs from privacy policies. The details are shown in Appendix A.

5.3 Large-scale Overbroad PDCP Analysis

With the selected parameters ($k = 11$ and $\alpha = \frac{1}{2}$), we conducted an overbroad PDCP analysis on the entire dataset. If the number of candidate apps of a target app is less than k, the actual number would be used for the calculation. From the total 10,042 target apps, POLICYCOMP extracted and analyzed 57,993 PDCPs.

PDCP level results are shown in Table 5. Particularly, 48.29% of PDCPs have the overbroad likelihood that are beyond the threshold α (high-risk+ medium-risk). Based on whether they are *Class-I personal data* (as listed in Table 4), they are then either classified as high-risk (1.50%) or medium-risk (46.79%). Such a high percentage of overbroad PDCPs cast a shadow over the privacy of billions of mobile users. Therefore, it is urgent to formulate and enforce clear standards for developers to regulate personal data collection.

We further evaluate privacy policy-level results and find that only 27.79% of privacy policies contain no high-risk nor medium-risk PDCPs. On average each privacy policy contains 0.09 high-risk PDCPs and 2.7 medium-risk PDCPs, suggesting the severity of overbroad collection. In the following, we will describe our findings in detail. Other evaluations, *i.e.*, longitudinal study, are listed in Appendix B.

5.3.1 Overbroad PDCP Risk Estimation

FINDING 1: *Class-II personal data may introduce an underestimated privacy risk.*

Since *Class-II personal data* (*i.e.*, types of personal data that are not highly protected personal data) are widely used by apps to provide services, users may be accustomed to providing such personal data in exchange for better services while underestimating the privacy risks of providing them. For 23 types of *Class-II personal data* (two types of personal data were ignored since they are too few), we calculated the ratios of the corresponding PDCPs that were classified as medium-risk and low-risk, as shown in Fig. 5(b). The ratios of medium-risk PDCPs range from 5.73% to 100%. Up to 78.26% (18/23) of *Class-II personal data* have overbroad (*i.e.*, medium-risk) ratios above 50%, including some widely used personal data, such as gender, age, and phone number. Hence, developers need to be more cautious before collecting these types of personal data when most of the counterpart apps do not require them.

FINDING 2: *Collecting Class-I personal data are a strong signal of overbroad collection.*

Although high-risk PDCPs (*i.e.*, overbroad collection of *Class-I personal data*) only account for 1.5%, the ratio between overbroad collection and the total collection of *Class-I personal data* is abnormally high. For each type of *Class-I personal data*, we calculated the ratio of the corresponding PDCPs that were classified as high-risk and low-risk. As shown in Fig. 5(a), the high-risk ratios are high, ranging from 88.57% to 100%. Compared with the results in Fig. 5(b), such high percentages suggest that collecting *Class-I personal data* is more likely overbroad and should be avoided without specific and reasonable purposes.

5.3.2 Overbroad PDCP Reasoning

After identifying the overbroad PDCPs, we further tried to find their additional purposes (*i.e.*, *justifications*) using the model presented in Sec. 3.3.2. As shown in Table 5 (right),

Table 6: Effect of One-to-many Privacy Policies

	#	H-risk	M-risk	L-risk
Apps that **do** use one-to-many policies	5,705	417 (1.21%)	16,703 (48.65%)	17,217 (50.14%)
Apps that **don't** use one-to-many policies	4,337	454 (1.92%)	10,429 (44.09%)	12,773 (53.99%)

only 35.1% $(= (184+9647)/(871+27132))$ of overbroad PDCPs (high-risk+medium-risk) have additional purposes.

FINDING 3: *Only* 31.07% *overbroad PDCPs have clear additional purposes.*

Among the 35.1% overbroad PDCPs that have additional purposes (*i.e.*, justifications), 11.47% claimed only unclear/general purposes, *e.g.*, "provide service", "support service", and "improve service" (listed in Sec. 4.2), resulting in 31.07% (= 35.1% × (1-11.47%)) overbroad PDCPs that have clear additional purposes. The usage of such unclear language in purpose descriptions is not recommended. According to the GDPR guidelines, "The information should be concrete and definitive; it should not be phrased in abstract or ambivalent terms or leave room for different interpretations. In particular, the purposes of, and legal basis for, processing the personal data should be clear" [5].

5.3.3 One-to-many Privacy Policies

By checking the developer names, we found that the 10,042 selected target apps belonged to 7,200 different developers, 3,703 (51.43%) of which owned multiple apps. By further checking the privacy policy links of apps belong to the same developer, we found that 2,863 developers owning multiple apps used one single privacy policy for their apps (*e.g.*, Gallery and Internet Browser use the same privacy policy), and indiscriminately described the collected personal data of all apps in this privacy policy. We coin this type of privacy policy as *one-to-many privacy policies*.

FINDING 4: *One-to-many privacy policies tend to include more overbroad PDCPs.*

As shown in Table 6, we can see that *one-to-many privacy policies* have a higher percentage (3.85% = 1.21% + 48.65% − 1.92% − 44.09%) of high&medium-risk PDCPs, which may be caused by PDCPs collected by other apps that share the same *one-to-many privacy policies*. Though declaring PDCPs in a privacy policy without actually collecting them is less harmful than actually collecting PDCPs without mentioning them, it may reduce the users' trust in the privacy policy since they need to give consent to the processing of more personal data than an app needs.

5.4 Notification to Developers

We selected 2,000 apps with highest overbroad likelihoods and shared our findings with their developers via the email addresses obtained from Google Play. Particularly, we shared our detection method and overbroad PDCPs (along with the overbroad likelihoods) detected by POLICYCOMP with these developers, and asked for their opinions on these findings. 1,661 emails are successfully delivered, as others are invalid or no longer being monitored.

Table 7: The responses from developers

		No. of Policies	No. of PDCPs
Acknowledge our findings	all findings	34	112
	partial findings	5	8 + 10 (necessary)
Disagree with our findings	Don't admit to collect	4	16
	PDCPs are necessary	9	23

As shown in Table 7, at the time of this writing, we have received responses from the developers of 52 apps, 39 of which acknowledge our findings, which is relatively substantial considering possible liability concerns [15]. For the remaining apps which did not reply to us, we examined their privacy policies one month after sending the emails, and found that the privacy policies of 74 apps have been updated by removing 180 overbroad PDCPs we sent.

In the following, we will discuss the responses in detail.

5.4.1 39 developers (30 apps with more than 100K+ downloads per app) acknowledge our findings

Among them, 34 developers acknowledge all our findings and provide the following explanations.

- 22 *developers commit to remove all overbroad PDCPs* (14 privacy policies have been updated). These developers mainly state that 1) their privacy policies are outdated and will remove these PDCPs from the latest version as they do not use them or need them; 2) their privacy policies are automatically generated by privacy policy generators which include these PDCPs by default; 3) overbroad PDCPs are wrongly added to the privacy policies.
- 4 *developers acknowledge that the overbroad PDCPs detected by* POLICYCOMP *are optional*. These developers acknowledge that overbroad PDCPs are optional for providing services and clarify that users could use their apps without these PDCPs. For example, one developer states that "These contents are voluntarily contributed to the app for only those users that want to contribute to the app".
- 8 *developers state that overbroad PDCPs detected by* POLICYCOMP *are collected by other apps of these developers.* Some developers claim that "This policy is designed to be one policy covering many mobile apps. As such, this app does not collect all the information you referenced in your email". As defined in Sec. 5.3.3, We coin this type of privacy policy as *one-to-many privacy policy*, which may reduce the users' trust in the privacy policy. We contact these developers again and tell them the harms of using *one-to-many privacy policies*. One developer promises that "We will revise the policy soon and update it accordingly".

Besides the above developers who totally agree with our findings, the remaining 5 developers acknowledge our partial findings. Further, some developers claim that partial overbroad PDCPs are necessary without providing the reasons for collecting them. We sent 10 such overbroad PDCPs to annotators from the law school for further analysis. Through analyzing apps' functionalities, annotators label 5 PDCPs as unnecessary.

5.4.2 13 developers disagree with our findings

For 4 developers that do not admit to collect these overbroad PDCPs, we manually inspected their privacy policies and found 3 developers actually declare to collect these PDCPs in privacy policies. We contact them again to report results of the manual inspection but receive no response.

The remaining 9 developers state these overbroad PDCPs are necessary for providing services. However, 3 of them do not explain the clear purposes for collecting these PDCPs, *e.g.*, one developer just states that "My app is an automation app and so requires all personal data you mentioned." Therefore, to further check the overbroad PDCPs that are considered as necessary, we also sent these PDCPs to annotators from the law school. The annotators label them using the same method as described in Sec. 5.2. For 23 overbroad PDCPs that are claimed by the developers as necessary, annotator label 9 PDCPs as unnecessary. We have sent the annotation results to these developers.

6 Case Study

In this section, we present two types of representative cases to explain why overbroad PDCPs occur.

How to Support Interaction: Requesting Contact List vs. Out-of-process Sharing. As data privacy becomes more crucial, various techniques that could help improve privacy have been proposed. One technique of particular interest is app referral. One common way of referring an app to a friend is through the use of the "address book" ("contact list"). That is why many apps claim to collect "address book". Alternatively, the out-of-process picker or a share sheet provided by the mobile operating system, which was introduced in [1, 8] enables the developers to achieve app referral without access to "contact list".

We observed this new technique when PolicyComp indicated that one food delivery app (com.gxxx.xxx, installs: 10M+) collects users' "contact list" while most of its counterpart apps do not. By checking its counterpart apps that achieve the same purpose, we found that they achieved their referral programs by using the share sheet, rather than requesting users' "contact list". We provide a redrew user interface in Fig. 6 to illustrate how such features could be integrated.

Moreover, we additionally performed code analysis on the counterpart apps and found that one counterpart app

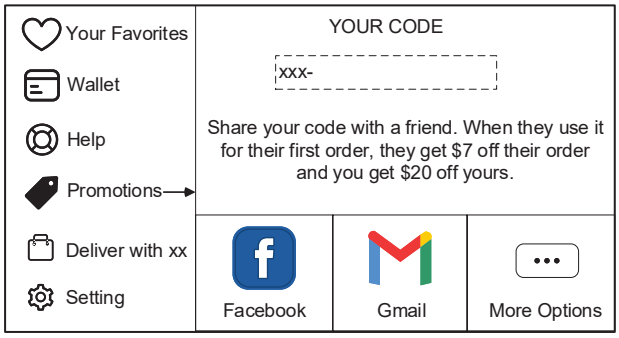


Figure 6: Invite friends

```
private android.net.Uri a(java.lang.String) throws java.lang.Exception{
    ......
1.$r4=<android.provider.ContactsContract:android.net.Uri CONTENT_URI>;
  Create the URI for the contacts information to retrieve.
    ......
2.$r7 = ContentResolver.query($r4, null, $r6, null, null)>;
  Retrieve the Cursor about the contacts information for further query.
```

Figure 7: Suspicious dead code about contacts API

(com.uxxx.xxx, installs: 100M+) might collect "contact list" information in the past, but stopped doing so now. Particularly, by using static analysis, we found a code snippet for requesting the "contacts", as shown in Fig. 7. We then set the "ContentResolver.query" method as source and tried to find it in the taint path generated by Flowdroid [17, 46]. However, it was not found. Therefore, we suspect that the code in Fig. 7 is a dead code whose functionality is replaced by new techniques.

Interestingly, when we shared our findings with developers, we found some developers have adopted this new technique (using a share sheet or shared link), but they did not realize to remove the statements about collecting "contact list" from their privacy policies. One habit tracking app (com.wayxxx.xxx, installs: 500K+) states that "The app actually do not have access to contacts, as this is a share sheet handled by the mobile operating system. I will change my policy! Keep up your good work." We believe that this new technique has been adopted by many apps, but the issue of synchronizing with the privacy policies has not been resolved.

Privacy Policy Generators Should be Carefully Used. Privacy policy generators are widely used by developers [48]. Although privacy policy generators are meaningful tools for developers, these generators are limited by the manually designed templates which cannot generate a dedicated privacy policy to cover all requirements of an app, as well as the developer's ability to carefully maintain the generated coarse privacy policy.

Actually, some developers tend to ignore the flaws of generators. After sharing our findings (overbroad PDCPs) with developers, 6 developers clarify that overbroad collections are caused by the misuse of privacy policy generators, *e.g.*, "the privacy policy are automatically generated by privacy

policy generator so its content does not represent the data that the app collects". By manually checking these apps' privacy policies and generators used by them, we found that the generated coarse privacy policies contain some examples of personal data, and these developers directly treat the coarse privacy policies as final privacy policies without changing these examples according to actual situations.

Moreover, when we deeply inspected overbroad collections in our dataset, we found some apps' privacy policies have the same issue. For example, 12 privacy policies of different app developers that are detected to have overbroad collections use the same sentence to describe their overbroad collections, which is caused by using the same privacy policy generator:

> For a better experience while using our Service, we may require you to provide us with certain personally identifiable information, including but not limited to your name, phone number, and postal address.

Through manually checking their privacy policies, we found that these privacy policies are generated by the same generator, "Privacy Policy Template Generator". Surprisingly, we found that 10 out of 12 privacy policies are exactly the same as the template on the website of the generator, including data collections and purposes. Especially, the above sentence describing data collections is generated by the generator and directly used by these privacy policies without any modification. We further checked these apps to find their actual collections and observed that actual data collections are different from what the privacy policies state, which means these developers directly treat the coarse privacy policy template as the final privacy policy without modifying it.

7 Limitation

PolicyComp is an automatic tool to identify overbroad PDCPs in apps' privacy policies for subsequent manual audit by legal experts. The current implementation is limited in handling the following scenarios:

(1) The same overbroad PDCPs shared by some target apps and their counterpart apps. As PolicyComp leverages PDCPs in counterpart apps' privacy policies as the potential standards, whether most counterpart apps have decent PDCPs will affect PolicyComp's performance. When most counterpart apps share the same overbroad PDCPs, these overbroad PDCPs in the target app will be identified as necessary.

(2) Lack of highly similar counterpart apps for some target apps. For some target apps, especially unpopular apps, it is difficult to collect enough highly similar counterpart apps. Therefore, the overbroad likelihoods of the targets app's PDCPs calculated using these dissimilar counterpart apps may be inaccurate. Furthermore, for a target app providing a special functionality, PDCPs collected for this functionality may be incorrectly identified when counterpart apps do not have similar functionalities.

(3) Inaccurate PDCPs extraction and regularization due to the limitations of existing NLP tools. The current implementation largely relies on state-of-the-art tools (*e.g.*, PolicyLint [15]) in this area that are facing the same challenges in extracting PDCPs: PDCPs extraction and regularization is highly dependent on the completeness of the collection verb list and synonyms of each type of personal data. If PDCPs in counterpart apps are incorrectly extracted and regularized, the overbroad likelihoods of PDCPs in the target apps would be inaccurate.

8 Discussion

Causes of overbroad PDCPs. We notice two main causes during our study. Firstly, many developers do not take personal data collection seriously (*e.g.*, directly use the coarse privacy policies generated by generators), or do not clearly know what personal data should be collected. Moreover, writing privacy policies requires continuous effort since the world is evolving. If developers do not pay attention to the development of privacy-friendly techniques (*e.g.*, the share sheet) and make timely updates, there is still a risk of overbroad collection. Solving the problem of overbroad collection is complex, which needs the help from multiple communities (*e.g.*, lawmakers, app stores, users, and developers).

Real-world application. PolicyComp could be useful for app developers to identify overbroad PDCPs. Since most existing privacy protection laws do not specify what types of personal data are necessary given a specific purpose, it is difficult for app developers to clearly know what personal data should be collected. PolicyComp could help app developers analyze the risks of PDCPs by computing a likelihood of being overbroad for each PDCP. App developers need to be cautious with PDCPs with higher likelihoods and learn how to achieve the same functionalities without collecting overbroad PDCPs from those counterpart apps.

Future directions for better data minimization. To draw legal conclusions for an app, it is important to understand the legitimate purposes of the app (*e.g.*, PDCPs could be collected for a broad variety of business purposes or only for the core functionalities), which can be distinguished by legal experts according to specific contexts (*e.g.*, applicable laws). Currently, it is difficult for automatic tools to extract clear purposes of PDCPs and understand legitimate purposes of an app. Therefore, drawing legal conclusions by automatic tools still remains a great challenge. Standardizing the privacy policy is a promising direction [27, 31], which requires developers to provide a standardized-table format representing all PDCPs and the corresponding purposes. Another direction relies on legal experts to formulate more detailed standards for reference to help developers write compliance privacy policies, *e.g.*, more cases of how PDCPs are collected in specific contexts.

Scalability. In the future, we will work on improving the performance of PolicyComp by integrating state-of-the-art NLP tools. Besides the advent of new NLP techniques for better understanding privacy policies, app stores are working on regularizing personal data collections. For example, Google Play will add " new safety section" in 2022, which requires developers to clearly disclose the collections of personal data [6]. We plan to integrate these improvements into PolicyComp for providing a more accurate analysis.

9 Conclusion

In this paper, we propose PolicyComp, an automatic framework for the detection of overbroad PDCPs in privacy policies. Instead of directly drawing legal conclusions that *data minimization* requirements have been breached, our work is to flag overbroad PDCPs for subsequent manual audit by legal experts. PolicyComp computes a likelihood of being overbroad for each PDCP in the target app's privacy policy, based on whether its counterpart apps also claim to collect the same type of personal data. We use PolicyComp to perform a large-scale analysis on 10,042 privacy policies of Android apps and flag 48.29% of extracted PDCPs to be overbroad. From the large-scale analysis results, we select 2,000 apps with highest overbroad likelihoods and share our findings with their developers. We receive 52 responses from these developers, 39 of which acknowledge our findings (*e.g.*, removing these overbroad PDCPs).

Acknowledgments

We thank the shepherd and other reviewers for their insightful comments. The work was supported in part by the National Natural Science Foundation of China under Grant 62132013, 72171145, and 61972453. Lu Zhou and Chengyongxiao Wei are the co-first authors. Haojin Zhu and Guoxing Chen are the corresponding authors.

References

[1] App store review guidelines. https://developer.apple.com/app-store/review/guidelines/#legal.

[2] California consumer privacy act of 2018. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.

[3] GDPR fine for merchant. https://easygdpr.eu/en/gdpr-incident/gdpr-fine-for-merchant/.

[4] General data protection regulation (GDPR). https://gdpr.eu/tag/gdpr/.

[5] Guidelines on transparency under regulation. https://ec.europa.eu/newsroom/article29/items/622227.

[6] New safety section in google play will give transparency into how apps use data. https://android-developers.googleblog.com/2021/05/new-safety-section-in-google-play-will.html.

[7] Personal information protection law of the People's Republic of China (PIPL). http://www.npc.gov.cn/npc/c30834/202108/a8c4e3672c74491a80b53a172bb753fe.shtml.

[8] Sending simple data to other apps. https://developer.android.com/training/sharing/send.

[9] Supermajority. https://en.wikipedia.org/wiki/Supermajority.

[10] UOOU-10138/18-8. https://www.uoou.cz/assets/File.ashx?id_org=200144&id_dokumenty=34470.

[11] Writing a GDPR-compliant privacy notice. https://gdpr.eu/privacy-notice/.

[12] Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. Privacy and human behavior in the age of information. *Science*, 347(6221):509–514, 2015.

[13] Wasi Uddin Ahmad, Jianfeng Chi, Tu Le, Thomas B. Norton, Yuan Tian, and Kai-Wei Chang. Intent classification and slot filling for privacy policies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, ACL '21, 2021.

[14] Ryan Amos, Gunes Acar, Elena Lucherini, Mihir Kshirsagar, Arvind Narayanan, and Jonathan Mayer. Privacy policies over time: Curation and analysis of a million-document dataset. In *Proceedings of the Web Conference 2021*, WWW '21, page 2165–2176, New York, NY, USA, 2021. Association for Computing Machinery.

[15] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. Policylint: investigating internal privacy policy contradictions on google play. In *28th USENIX Security Symposium*, USENIX Security '19, pages 585–602. USENIX Association, 2019.

[16] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. Actions speak louder than words: Entity-sensitive privacy policy and data flow analysis with policheck. In *29th USENIX Security Symposium*, USENIX Security '20, pages 985–1002. USENIX Association, 2020.

[17] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '14, page 259–269, New York, NY, USA, 2014. Association for Computing Machinery.

[18] Vinayshekhar Bannihatti Kumar, Roger Iyengar, Namita Nisal, Yuanyuan Feng, Hana Habib, Peter Story, Sushain Cherivirala, Margaret Hagan, Lorrie Cranor, Shomir Wilson, Florian Schaub, and Norman Sadeh. Finding a choice in a haystack: Automatic extraction of opt-out statements from privacy policy text. In *Proceedings of The Web Conference*, WWW '20, page 1943–1954, New York, NY, USA, 2020. Association for Computing Machinery.

[19] Travis D. Breaux and Ashwini Rao. Formal analysis of privacy requirements specifications for multi-tier applications. In *21st*

IEEE International Requirements Engineering Conference, RE '13, pages 14–23, 2013.

[20] Duc Bui, Kang G Shin, Jong-Min Choi, and Junbum Shin. Automated extraction and presentation of data practices in privacy policies. In *Proceedings on Privacy Enhancing Technologies*, PETs '21, pages 88–110, 2021.

[21] Duc Bui, Yuan Yao, Kang G Shin, Jongmin Choi, and Junbum Shin. Consistency analysis of data-usage purposes in mobile apps. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21. Association for Computing Machinery.

[22] José González Cabañas, Ángel Cuevas, and Rubén Cuevas. Unveiling and quantifying facebook exploitation of sensitive personal data for advertising purposes. In *27th USENIX Security Symposium*, USENIX Security '18, pages 479–495, Baltimore, MD, August 2018. USENIX Association.

[23] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy ... now take some cookies: Measuring the GDPR's impact on web privacy. In *26th Annual Network and Distributed System Security Symposium*, NDSS '19, San Diego, California, USA, 2019.

[24] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium*, USENIX Security '18, pages 531–548, Baltimore, MD, August 2018. USENIX Association.

[25] Suman Jana, Úlfar Erlingsson, and Iulia Ion. Apples and oranges: Detecting least-privilege violators with peer group analysis. *CoRR*, abs/1510.07308, 2015.

[26] He Jiang, Jingxuan Zhang, Xiaochen Li, Zhilei Ren, David Lo, Xindong Wu, and Zhongxuan Luo. Recommending new features from mobile app descriptions. *ACM Trans. Softw. Eng. Methodol.*, 28(4), October 2019.

[27] Patrick Gage Kelley, Lucian Cesca, Joanna Bresee, and Lorrie Faith Cranor. Standardizing privacy notices: An online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1573–1582, New York, NY, USA, 2010. Association for Computing Machinery.

[28] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. The privacy policy landscape after the gdpr. In *Proceedings on Privacy Enhancing Technologies*, PETs '20, pages 47–64, 2020.

[29] Kangjie Lu, Zhichun Li, Vasileios P Kemerlis, Zhenyu Wu, Long Lu, Cong Zheng, Zhiyun Qian, Wenke Lee, and Guofei Jiang. Checking more and alerting less: detecting privacy leakages via enhanced data-flow analysis and peer voting. In *22nd Annual Network and Distributed System Security Symposium*, NDSS' 15, 2015.

[30] Miti Mazmudar and Ian Goldberg. Mitigator: Privacy policy compliance using trusted hardware. In *Proceedings on Privacy Enhancing Technologies*, PETs '20, pages 204–221, 2020.

[31] Aleecia M. McDonald, Robert W. Reeder, Patrick Gage Kelley, and Lorrie Faith Cranor. A comparative study of online privacy policies and formats. In Ian Goldberg and Mikhail J. Atallah, editors, *Privacy Enhancing Technologies*, pages 37–55, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[32] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. Share first, ask later (or never?) studying violations of GDPR's explicit consent in android apps. In *30th USENIX Security Symposium*, USENIX Security '21, pages 3667–3684. USENIX Association, August 2021.

[33] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. Dark patterns after the gdpr: Scraping consent pop-ups and demonstrating their influence. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[34] Rahul Pandita, Xusheng Xiao, Wei Yang, William Enck, and Tao Xie. WHYPER: Towards automating risk assessment of mobile applications. In *22nd USENIX Security Symposium*, USENIX Security '13, pages 527–542, Washington, D.C., USA, August 2013. USENIX Association.

[35] Sai Teja Peddinti, Igor Bilogrevic, Nina Taft, Martin Pelikan, Úlfar Erlingsson, Pauline Anthonysamy, and Giles Hogben. Reducing permission requests in mobile apps. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 259–266, New York, NY, USA, 2019. Association for Computing Machinery.

[36] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *25th Network & Distributed System Security Symposium*, NDSS '18. The Internet Society, 2018.

[37] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D Breaux, and Jianwei Niu. Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, pages 25–36, 2016.

[38] Suibin Sun, Le Yu, Xiaokuan Zhang, Minhui Xue, Ren Zhou, Haojin Zhu, Shuang Hao, and Xiaodong Lin. Understanding and detecting mobile ad fraud through the lens of invalid traffic. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 287–303, New York, NY, USA, 2021. Association for Computing Machinery.

[39] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. (un)informed consent: Studying GDPR consent notices in the field. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, New York, NY, USA, 2019. Association for Computing Machinery.

[40] Xiaoyin Wang, Xue Qin, Mitra Bokaei Hosseini, Rocky Slavin, Travis D. Breaux, and Jianwei Niu. Guileak: Tracing privacy policy claims on user input data for android applications. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, page 37–47, New York, NY, USA, 2018. Association for Computing Machinery.

[41] Takuya Watanabe, Mitsuaki Akiyama, Tetsuya Sakai, and Tatsuya Mori. Understanding the inconsistencies between text

descriptions and the use of privacy-sensitive resources of mobile apps. In *11th Symposium On Usable Privacy and Security*, SOUPS '15, pages 241–255, Ottawa, July 2015. USENIX Association.

[42] Charles Weir, Ben Hermann, and Sascha Fahl. From needs to actions to secure apps? The effect of requirements and developer practices on app security. In *29th USENIX Security Symposium*, USENIX Security '20, pages 289–305. USENIX Association, August 2020.

[43] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. Can we trust the privacy policies of android apps? In *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '16, pages 538–549, 2016.

[44] Mu Zhang, Yue Duan, Qian Feng, and Heng Yin. Towards automatic generation of security-centric descriptions for android apps. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 518–529, New York, NY, USA, 2015. Association for Computing Machinery.

[45] Xiaoqing Zheng, Jiehang Zeng, Yi Zhou, Cho-Jui Hsieh, Minhao Cheng, and Xuanjing Huang. Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 6600–6610, Online, July 2020. Association for Computational Linguistics.

[46] Tong Zhu, Yan Meng, Haotian Hu, Xiaokuan Zhang, Minhui Xue, and Haojin Zhu. Dissecting click fraud autonomy in the wild. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 271–286, New York, NY, USA, 2021. Association for Computing Machinery.

[47] Sebastian Zimmeck and Steven M. Bellovin. Privee: An architecture for automatically analyzing web privacy policies. In *23rd USENIX Security Symposium*, USENIX Security '14, pages 1–16, San Diego, CA, August 2014. USENIX Association.

[48] Sebastian Zimmeck, Rafael Goldstein, and David Baraka. Privacyflash pro: automating privacy policy generation for mobile apps. In *28th Network and Distributed System Security Symposium*, NDSS' 21, 2021.

[49] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel R Reidenberg, N Cameron Russell, and Norman Sadeh. Maps: Scaling privacy compliance analysis to a million apps. In *Proceedings on Privacy Enhancing Technologies*, PETs '19, page 66, 2019.

[50] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman M Sadeh, Steven M Bellovin, and Joel R Reidenberg. Automated analysis of privacy requirements for mobile apps. In *Network & Distributed System Security Symposium*, NDSS '17, 2017.

A Other Performance Evaluations

Table 8 shows the data provided to annotators for determining whether a PDCP is overbroad and annotation results.

For evaluating the performance of elements of the pipeline (e.g., counterpart app search and PDCP extraction and regularization), we also annotated selected privacy policies of 300 target apps and calculated evaluation results as follows:

counterpart app search. 3 human annotators who are senior Ph.D. students in privacy research were shown the detail information (*e.g.*, app name and description) of the 300 target apps and their candidate similar apps. Similar to [35], annotators evaluated the functional similarities (0-not similar, 1-a little similar, 2-similar, 3-very similar) between a target app and its candidate similar apps based on the detail information. In this step, each annotator rated over $5,891$ app pairs. The labeled results of the 3 annotators were averaged for constructing the final labels.

Then we calculated the average similarity between 300 target apps and their top-k counterpart apps determined by PolicyComp. The average similarity ranges from 2.72 to 2.51 when k ranging from 3 to 15. Especially, when $k = 11$ (the selected parameters for large-scale analysis), the average similarity reaches 2.62.

PDCP extraction and regularization. For $2,410$ PDCPs extracted from 300 target apps, we showed the 3 Ph.D. students each PDCP and corresponding sentences describing the PDCP. For each PDCP, each annotator assigned a 'Y' label if the PDCP is correctly extracted and regularized, otherwise the annotator assigned a 'N' label. The labeled results of the 3 annotators were merged based on the majority principle. The resulting ground-truth dataset consists of $2,159$ 'Y' labels and 251 'N' labels. That is to say, PolicyComp achieves a 89.6% $(2,159/2,410)$ precision for extracting and regularizing PDCPs from privacy policies.

B Other Evaluations

In this section, we present other evaluations. Table 9 shows the table version of Fig. 5.

Longitudinal Study. We also evaluated whether developers update PDCPs in their privacy policies between different versions.

Finding 5: *With multiple privacy protection laws coming into effect, developers have begun to regulate their personal data collection.*

To conduct a longitudinal study, we compared the results from the dataset described in Sec. 5.1 (collected in August 2021) with those from another smaller privacy policy dataset we built four months earlier in April 2021. Particularly, $5,889$ target apps in total exist in both datasets, $1,233$ of which have updated privacy policies. Among the $1,521$ PDCPs removed from the newer versions of the $1,233$ privacy policies, $1,304$ were considered overbroad (high-/medium-risk) by PolicyComp. And 534 out of 771 PDCPs added to the newer version were flagged as overbroad by PolicyComp.

Table 8: Data format provided to annotators for determining whether a PDCP is overbroad and annotation results (APP ID: com.cixxxs.loxxx; APP Name: Loxx of the Faxx; Category: Game-Action; Description: https://play.google.com/store/apps/details?id=com.cixxxs.loxxx; privacy policy: http://mobile.lxxxx.com/; P: positive, N: negative)

PDCP	Sentences	Label
Date of birth	(1) When you register to play our games, we may ask you to provide certain pieces of information, which could include: your e-mail, username, phone number, gender, birthdate, home address, and address book. (2) Your gender and birthdate may be used to analyze user trends and target certain promotions.	P/P/P
Contact list	(1) When you register to play our games, we may ask you to provide certain pieces of information, which could include: your e-mail, username, phone number, gender, birthdate, home address, and address book. (2) If you sign into our Service with Facebook Connect we will collect information that is visible via your Facebook account such as: your first and last name, and list of Facebook friends.	P/P/P
Payment info	If we begin offering our service on a platform without an in-app purchase billing system, we may need to collect credit card and billing information.	N/N/P
Gender	(1) When you register to play our games, we may ask you to provide certain pieces of information, which could include: your e-mail, username, phone number, gender, birthdate, home address, and address book. (2) Your gender and birthdate may be used to analyze user trends and target certain promotions.	P/P/P
Phone number	(1) When you register to play our games, we may ask you to provide certain pieces of information, which could include: your e-mail, username, phone number, gender, birthdate, home address, and address book. (2) Your phone number may be used to help connect you with other users via our social networking system and for SMS notifications.	N/P/P
Physical address	When you register to play our games, we may ask you to provide certain pieces of information, which could include: your e-mail, username, phone number, gender, birthdate, home address, and address book.	P/P/P
Name	(1) If you sign into our Service with Facebook Connect we will collect information that is visible via your Facebook account such as: your first and last name, and list of Facebook friends. (2) Your name will be used for user registration and in social features, which may include friend-to-friend interaction, chat or messaging functionality, public leader boards, and other similar features.	P/N/P
IP add.	During your use of our website there is data concerning your visit that is collected, e.g. your IP address.	N/N/N
Email address	(1) If you provide your email address in connection with any game, the e-mail address will be retained and we may use it to contact you about your gaming experience and notify you about company news. (2) When you register to play our games, we may ask you to provide certain pieces of information, which could include: your e-mail, username, phone number, gender, birthdate, home address, and address book.	N/N/N

Table 9: Table Version of Fig. 5

(a) Table Version of Fig. 5(a)

	Political view	Union membsh.	Genetic data	Biometric data	Sex life	Race	Health	Religious view	Sexual orientation
H-risk	1.0	1.0	1.0	1.0	1.0	1.0	0.90	0.90	0.88
L-risk	0	0	0	0	0	0	0.10	0.10	0.12

(b) Table Version of Fig. 5(b)

	Driver's license	Contact log	Passport number	purch. hist	Education info	Contact list	Audio	PRO info	SSN number	Date of birth	Gender	Payment info
M-risk	1.0	1.0	1.0	0.99	0.99	0.99	0.98	0.96	0.95	0.94	0.90	0.84
L-risk	0	0	0	0.01	0.01	0.01	0.02	0.04	0.05	0.06	0.10	0.16
	Browsing history	Age	Photo	Phone number	Precise location	Physical add.	Coarse location	Name	IP add.	Email add.	Device ID	
M-risk	0.82	0.81	0.71	0.67	0.65	0.60	0.48	0.35	0.22	0.11	0.06	
L-risk	0.18	0.19	0.29	0.33	0.35	0.40	0.52	0.65	0.78	0.89	0.94	

This indicates that developers are trying to regulate their personal data collection in privacy policies but the problem of overbroad collection has not been completely solved.

C The details of 52 developers' responses

Table 10 lists the target apps that reply to us and the overbroad PDCPs in these apps' privacy policies.

Table 10: The details of 52 developers' responses

Package Name	Downloads	Overbroad PDCPs	Type
uk.xxx.chexxx	50M+	Age, Gender	remove all overbroad PDCPs
com.redxxx.twoxxx	10M+	Gender, Date of birth	remove all overbroad PDCPs
com.insxxx.cast.webxxx	10M+	Photo, IP address	remove all overbroad PDCPs
com.woxxx	10M+	Biometric data	remove all overbroad PDCPs
com.mycatxxx.xxxt	10M+	Gender, Date of birth	remove all overbroad PDCPs
uk.co.aifaxxxx	10M+	Age, Gender	remove all overbroad PDCPs
com.andxxx	10M+	Contact list, Payment information, Precise location	remove all overbroad PDCPs
com.xxx.vixxx	5M+	Gender, Date of birth	remove all overbroad PDCPs
com.myxxx.louxxx	5M+	Gender, Date of birth	remove all overbroad PDCPs
com.phxxx.idlexxx	1M+	Contact list	remove all overbroad PDCPs
com.xxxlaus	1M+	Gender, Date of birth	remove all overbroad PDCPs
com.apoxxx.ligxxx	500K+	Physical address, Phone number, Payment information	remove all overbroad PDCPs
com.waxxx.app	500K+	Professional information, Contact list, Browsing and search history	remove all overbroad PDCPs
com.arxxx.overxxx	500K+	Calendar, Contact list, Physical address, Photo	remove all overbroad PDCPs
com.monkxxx.bananxxx	100K+	Gender, Date of birth	remove all overbroad PDCPs
com.juxxx.opxxx	100K+	Precise location	remove all overbroad PDCPs
it.feio.xxx	100K+	Phone number, Precise location, Name, Physical add., Device id, Email address	remove all overbroad PDCPs
com.difxxx.xxx	100K+	Gender, Date of birth	remove all overbroad PDCPs
ponydxxx.ponyxxx	10K+	Professional information, Date of birth, Gender, Name	remove all overbroad PDCPs
com.Privaxxxx	5,000+	Date of birth, Browsing and search history, Payment information	remove all overbroad PDCPs
com.woxxx.ombxxx	5,000+	Payment information, Name, Email address, Physical address	remove all overbroad PDCPs
com.vanxxx.privaxxx	5,000+	Precise location, Name	remove all overbroad PDCPs
com.avxxx.cleaner	50M+	Date of birth, Payment information, Precise location, Physical address, Name	optional
com.slxxx.app	1M+	Audio, Physical address, Professional information, Photo	optional
com.app.xxx	100K+	Sexual orientation, Religious view, Date of birth, Name	optional
conxxx.app	10K+	Gender, Calendar, Photo, Contact list, Precise location, Payment information	optional
com.uxxx.android	10M+	Race, Gender, Date of birth, Phone number, Professional information	One-to-many Policy
com.sxxxx.reaxxxx	10M+	Health, Driver's license, Passport number, Date of birth, SSN, Contact list	One-to-many Policy
com.rexxx.android	5M+	Health, Race, Physical address, Phone number, Name, IP address	One-to-many Policy
com.pxxx.chinxxxx	1M+	Purchasing hist, Professional info, Education info, Gender, Phone number	One-to-many Policy
com.a3xx.comxx	500K+	Education information, Photo, Gender, Physical address, Professional info	One-to-many Policy
com.xxx.app	10K+	Passport Number, Driver's Licence Number	One-to-many Policy
net.fxxx.xxxd	10K+	Phone number, Name, Physical address, Email address, IP address	One-to-many Policy
com.gaxxx.xxxx	10K+	Audio, Contact list, Photo	One-to-many Policy
com.rexxxx.android	500K+	Gender, Date of birth, Professional information, IP address, Device identifier	remove partial PDCPs
com.langxxxx.drops	5M+	Browsing and search history, Photo	partially optional
com.wikxxx.wxxx	1M+	Audio, Photo, Phone number	partially optional
com.pcxxx.pcxxx	1M+	Gender, Purchasing history, Browsing and search history	partially optional
com.taxxx.ipn	10K+	Professional information, Phone number, Photo, Name, Email address	partially optional
org.freedxxxx.fdm	1M+	Audio, Browsing and search history, Photo, Name	does not admit to collect
air.com.xxxx	1M+	SSN, Photo, Physical add., Browsing history, Phone number, IP add.	does not admit to collect
com.bexxxx.thixxx	500K+	Precise location, Calendar, Contact list, Photo	does not admit to collect
com.xxxx.rxx	50K+	Professional information, Photo	incorrect extraction
at.ner.lexxxx	50M+	Purchasing history, Browsing and search history	claim to be necessary
net.dinxxx.tasxxx	1M+	Location, Photo, Name, Email address	claim to be necessary
com.mc.amaxxxx	1M+	Audio, IP address	claim to be necessary
at.ner.zombxxxx	10M+	Purchasing history, Browsing and search history, Physical address	claim to be necessary
com.mixxxx.app	1M+	Audio, Photo	claim to be necessary
com.faxxxx.app	100K+	Professional info, Calendar, Payment info, IP add., Phone number	claim to be necessary
xxx.com.Zimxx	50K+	Precise location	claim to be necessary
com.tixxxx.dexxx	10K+	Physical address, Email address	claim to be necessary
me.thxxx.app	5,000+	Browsing and search history, Email address	claim to be necessary