Are We Getting Well-informed? An In-depth Study of Runtime Privacy Notice Practice in Mobile Apps

Shuai Li*
lis19@fudan.edu.cn
Fudan University
Shanghai, China

Zhemin Yang*
yangzhemin@fudan.edu.cn
Fudan University
Shanghai, China

Yuhong Nan
nanyh@mail.sysu.edu.cn
Sun Yat-sen University
Guangzhou, China

Shutian Yu
yushutian@fudan.edu.cn
Fudan University
Shanghai, China

Qirui Zhu
qrzhu23@m.fudan.edu.cn
Fudan University
Shanghai, China

Min Yang
m_yang@fudan.edu.cn
Fudan University
Shanghai, China

ABSTRACT

Under the General Data Protection Regulation (GDPR), mobile app developers are required to inform users of necessary information at the time when user data is collected (called users' "Right-to-be-Informed"). This is typically done by app developers via providing runtime privacy notices (RPNs for short). However, given the heterogeneous privacy data types and data access patterns in modern apps, it is not clear to what extent apps (app developers) effectively fulfill this compliance requirement in practice.

In this paper, we perform the first systematic study of current RPN practices in mobile apps. Our research endeavors to comprehend (1) the ecosystem of RPN, (2) potential gaps between legal requirements and RPN practices, and (3) the underlying reasons for such gaps. To achieve this, we design an automated pipeline - RENO that can effectively identify, extract, and analyze RPN at a large scale. With the help of RENO, we investigated 4,656 mobile apps selected from 19 European Union countries. Our analysis reveals a number of interesting findings. For example, 77.10% of user data collection behaviors lack RPNs. Among those provided RPNs, 86.35% of them have no more than three required notice elements when GDPR requires seven. In addition, to further understand the reasons behind such gaps, we perform a notification campaign and ask for feedback from the app developers. Indeed, the collected responses highlighted several critical reasons. For instance, a substantial proportion of app developers regard RPN as an optional complement to their privacy policies as RPNs are not strictly enforced by app stores. Our study shows the pressing need for better transparency in user data collection delivered by RPN.

CCS CONCEPTS

• **Security and privacy** → *Usability in security and privacy*;

KEYWORDS

Runtime Privacy Notice, Right to be Informed, Mobile Application, GDPR Compliance

ACM Reference Format:
Shuai Li, Zhemin Yang, Yuhong Nan, Shutian Yu, Qirui Zhu, and Min Yang. 2024. Are We Getting Well-informed? An In-depth Study of Runtime Privacy Notice Practice in Mobile Apps. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3658644.3670377

1 INTRODUCTION

While benefiting from the rapid development of mobile applications (apps for short), mobile users complain about the extensive collection of private data and expect better transparency [47, 78]. To protect user privacy, multiple privacy-preserving laws such as GDPR [20], CCPA [6] and PIPL [33], are enacted in different regions across the world. Among such legislation, a commonly shared, fundamental agreement on private data is the "Right-to-be-Informed"[1]. It means that people have the right to be notified about what, how and why their data is collected, used, shared, and sold [7, 11, 21]. To protect users' "Right-to-be-Informed", mobile apps typically provide detailed privacy policies for disclosing data practices related to user privacy. However, prior research has shown that privacy policies share inherent limitations for effectively delivering data usage practices. For example, due to the long and tedious content, privacy policy cost users too much effort to comprehensively understand [51, 68, 71, 81].

In addition to privacy policies, regulations also require app developers to provide Runtime Privacy Notices (RPNs for short) for more accurate, contextual in-app disclosures of data practices. In this way, users can get informed in a more timely and concise fashion regarding their data usage and collection. For example, GDPR states that "Where data is obtained directly, the person must be *immediately informed*" [21]. Similarly, the Federal Trade Commission (FTC) emphasizes that app developers should provide just-in-time disclosures [45] to users about their data access and collection.

Given the RPNs expected by regulators, it is more important for app developers to implement informative RPNs to fulfill such

*Co-first authors.

CCS '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-0636-3/24/10
https://doi.org/10.1145/3658644.3670377

[1]This right may have various names, e.g., "Right-to-be-Informed" in GDPR [21] and "Right-to-Know" in CCPA [7]. We use the term "Right-to-be-Informed" as its unified name for illustration.

requirements. Unfortunately, this problem has been overlooked for a long time. More specifically, prior research on mobile privacy (non-)compliance is mostly focused on analyzing app's privacy policies [36, 37, 56, 60, 63, 82, 89–91], such as checking statement contradictions [36] within the privacy policy itself, or checking inconsistencies between policy statements and app behaviors [37, 60, 63, 82, 89–91]. To the best of our knowledge, no prior research has discussed the real practices of RPN in mobile apps, from the perspective of law-compliance. Besides, although RPNs have been sufficiently discussed in the web-ecosystem [67, 74, 85], these works are mainly focused on the cookie consent notice, which is quite different from RPNs in mobile apps in terms of data types, usage scenarios, etc.

Our Work. In this paper, we perform the first in-depth study of the ecosystem and law (non-)compliance of RPNs in mobile apps. Our research aims at seeking answers to three key research questions, including (1) the ecosystem of RPN, (2) potential gaps between legal requirements and RPN practices, and (3) the underlying reasons for such gaps.

A systematic understanding of mobile RPN practices requires analyzing a large number of real-world RPN instances. Achieving this, however, is by no means trivial. Particularly, different from privacy policies which are quite easy to access (e.g., from the app description page in app stores), RPNs are often hidden inside the interactions between users and mobile apps. Therefore, it is rather difficult to identify and extract RPNs in scale. For example, we need to differentiate RPNs from those normal UIs in the app. Besides, compared to the content of privacy policies that are well-structured, RPNs are more fragmented and semantics-vague. For example, as shown in Figure 1(d), the data subject (contacts, call log, etc.) and usage descriptions are separated from one complete sentence. Such cases make it more difficult to accurately understand the data practices stated in RPNs in an automated manner.

To this end, this paper first designs and implements RENO - an automatic, end-to-end pipeline for analyzing RPNs at a large scale through the following steps. (1) Firstly, RENO automatically explores the app and monitors the data collection behaviours at runtime. In the meantime, RENO continuously inspects the app UI to check whether there is an RPN presented. To identify RPN, RENO trains a customized classifier that considers both the context of RPN appearance and semantics. This methodology works because RPNs are typically presented in a specialized context to gain user attention and inform users, which can be well-differentiated from other ones (i.e., a normal UI that does not serve the purpose of RPN). (2) Secondly, RENO performs a fine-grained analysis to check whether a particular element required by law is present in the identified RPN, as well as its content. This is done by a two-stage analysis: the first stage implements a set of element-specific classifiers to learn and identify the existence of notice elements (e.g., the collected data type and purposes of collection), while the second stage utilizes named entity recognition (NER for short) to extract the notice elements from the sentence (short term) identified by the element-level classifier. (3) Lastly, based on the identified RPNs and extracted notice elements, RENO performs a (non-)compliance analysis and reports those missed RPNs or substandard RPNs.

Measurement and Findings. With the help of RENO, we perform a large-scale measurement regarding the RPN compliance in the mobile ecosystem. Our research takes GDPR with apps in European Countries as an instance for alignment. Our research showed that the practices of RPN are far from sufficient to comply with statements required by law enforcement, raising a concerning situation that threatens user privacy. Particularly, we first built a research dataset that comprises 4,656 mobile apps downloaded from Google Play Stores across 19 EU countries. Subsequently, by employing RENO upon this dataset, our study unveiled two critical gaps (i.e., RPN existence and RPN quality) between GDPR and RPN practices.

Firstly, RPNs *are significantly missed in mobile apps.* For example, among our analyzed apps, only 51.22% of them provide the required RPNs. Besides, 77.10% user data collection behaviors are conducted without any RPNs, meaning that users may not be aware of such data transmission (usage). In the meantime, *even for those* RPNs *provided by apps, their quality is far from satisfied from the perspective of law requirements.* For example, 72.43% of our collected RPNs provide either two or three notice elements, while GDPR requires the existence of seven notice elements. More importantly, our research shows several dark patterns adopted by apps when presenting RPNs, such as "collect before providing notice" and "detailed notice after refusal" (see §6.2 for more details).

To further understand the root causes behind such RPN non-compliance, we launched a notification campaign and analyzed feedback collected from app developers. This campaign enables us to see why certain gaps exist between law requirements and the real RPN practices in mobile apps. Indeed, the collected responses highlighted several critical reasons. For example, developers mistakenly believe that compliance with GDPR could be achieved merely by providing privacy policies. Through multiple rounds of communication with app developers, we found that they face a dilemma of balancing user experience and law compliance requirements.

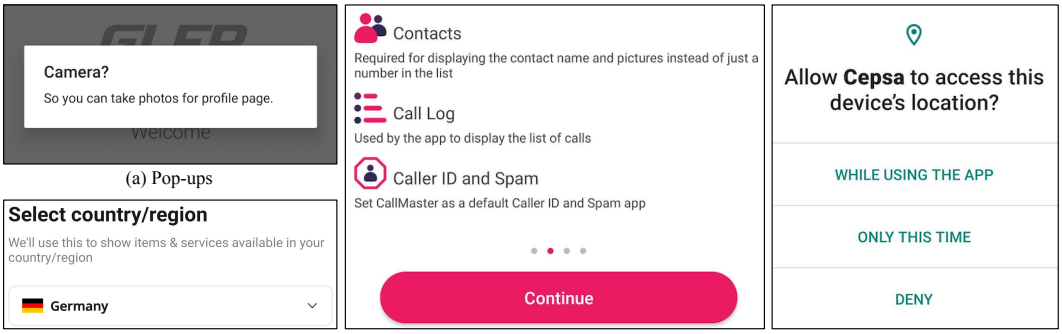
Our research highlights a long-term overlooked issue in the domain of privacy compliance. We believe that our study can serve as a valuable resource for the community. Particularly, developers can better align their apps with legal regulations and concurrently enhance the transparency of private data practices for app users.

In summary, this paper's contributions are outlined as follows:

- We perform the first in-depth analysis of the ecosystem of RPNs in mobile apps.
- We design and implement RENO, an automatic pipeline that supports analyzing RPN practices in mobile apps at a large scale.
- We systematically analyze RPN practices across 4,656 mobile apps in the wild. Our analysis highlighted the gaps between law requirements and RPN practices adopted by app developers.
- We thoroughly investigate why the identified gaps exist through a notification campaign, which can help understand and improve the situation of RPNs.

2 BACKGROUND

In this section, we introduce the necessary background for establishing a comprehensive understanding of RPN and relevant law requirements on it.



(a) Pop-ups (b) UI text label (c) Hint page (d) Runtime permission prompt

Figure 1: Examples of runtime privacy notices (RPNs) in mobile apps.

Table 1: Summarized key points related to the "Right-to-be-Informed" in privacy-preserving laws in different countries (regions).

Law Name	Country/Area	Requirement of Notice Timing	Content of Privacy Notice*	Format of Privacy Notice
GDPR [19, 21]	European Union	✓	6 (IC UR PP LB SP ER __)	clear, intelligible, and easily accessible
CCPA [7]	California-US	✓	5 (__ UR PP __ SP ER TD)	clear, conspicuous
CPRA [11]	California-US	✓	5 (__ UR PP __ SP ER TD)	clear, conspicuous
COPPA [8]	United States	✓	4 (IC UR PP __ __ __ TD)	direct, prominent
VCDPA [10]	Virginia-US	-	4 (__ UR PP __ __ ER TD)	accessible, clear, meaningful
APPI [2]	Japan	-	1 (__ __ PP __ __ __ __)	-
PDPA [25]	Singapore	✓	3 (IC UR PP __ __ __ __)	-
Privacy Act 1988 [28]	Australia	✓	6 (IC UR PP LB __ ER TD)	-
PIPL [33]	China	✓	5 (IC UR PP __ SP __ TD)	conspicuous, clear, understandable
CSL [12]	China	-	2 (__ __ PP __ __ __ TD)	-
DPA2018 [13, 32]	United Kingdom	✓	6 (IC UR PP LB SP ER __)	concise, transparent, intelligible and easily accessible form, using clear and plain language
FADP [15, 17]	Switzerland	✓	4 (IC __ PP __ __ ER TD)	concise, transparent, clear and readily accessible
BDSG [16]	Germany	-	6 (IC UR PP LB SP ER __)	in a general form and accessible

* IC: identity of data controller; UR: user rights; PP: processing purpose of user data; LB: legal basis of processing user data; SP: storage period of user data; ER: entity of user data receiver; TD: types of collected user data; Note that several laws (e.g., DPA2018 and GDPR) also require notices upon automated user data processing behavior and cross-border data transmission. However, as clarified in "Scope of Our Research" below, these two notice elements are out of our research scope and thus not included.

Runtime Privacy Notice (RPN). RPN refers to the in-app privacy notice that is presented to users while they are interacting with mobile apps. The biggest advantage is that RPN makes it inevitable for users to see its content, which is the premise of defending users' "Right-to-be-Informed".

To understand what RPN looks like, we first manually interacted with 750 randomly selected popular mobile apps and recorded RPNs. As shown in Figure 1, several implementation types of observed RPNs are illustrated. In fact, the ways to implement RPNs (e.g., format, content and so on) are quite different across apps and contexts. For instance, RPNs may be presented within pop-up windows, UI text labels, and runtime permission prompts. Its content may contain at least but not limited to the processed user data along with the purpose of user data processing. Furthermore, the prompts of runtime permission requests managed by the Android framework [55] are also taken as RPNs since they also serve the purpose of informing users at the time of data collection.

Law requirements about RPN. Most of the surveyed laws have concrete provisions on the existence and quality of RPNs. Specifically, to understand the law requirements for protecting users' "Right-to-be-Informed", we first manually reviewed a number of privacy-preserving laws and the surveyed results are shown in Table 1. In particular, three aspects are involved, including the timing, content and format of provided privacy notices.

When it comes to the GDPR requirements for timing, content, and format of RPNs, they can be viewed from the following two general perspectives:

- **RPN Existence.** Under Article 13 of the GDPR [21], the data controller is obligated to provide the data subject with necessary notice information at the time when personal data are

obtained. Since user data collection typically occurs during users' interaction with mobile apps, privacy notices must be provided in the runtime of mobile apps (instead of privacy policies). Therefore, to keep mobile apps compliant with GDPR, mobile app developers (typically as the data controller [29]) must ensure the existence of relevant RPNs.

- **RPN Quality.** As required by GDPR, privacy notices should be provided *at the time when personal data are obtained.* Therefore, the timing of providing RPNs should be before or at the same time when user data is accessed or collected so that mobile users (i.e., the data subject) can be timely notified as required. Furthermore, GDPR has requirements for the content of RPNs. Particularly, an RPN should have the required notice elements (as clearly stated in GDPR [21]) to be a law-compliant one. As shown in Table 1, six necessary notice elements (i.e., IC, UR, PP, LB, SP, ER) are required. Apart from the content, GDPR also regulates the format of RPNs. Specifically, mobile app developers are required to provide them in a clear, intelligible, and easily accessible manner [19].

Research scope. In our research, we investigate the real-world practices of RPNs while following GDPR's core requirements of RPNs existence and RPN quality. Particularly, as shown in Table 2, our study mainly focuses on the notice elements that can be concretely shown to users via the app's user interface (UI) - IC, UR, TD, PP, LB, SP, ER. Additionally, we assume the notice element - TD (i.e., types of collected user data) is mandatory according to Article 13 of GDPR, because the absence of this notice element could raise user confusion about what type of user data is accessed or collected. Besides, since the identity of data controller (IC) and user rights (UR) do not change according to the concrete user data collection, they are categorized into the general notice element type. Vice versa, other required notice elements are assigned to the specific notice element type.

Table 2: Key notice elements of "Right-to-be-Informed" in GDPR.

Notice Elements	General	IC: identity of data controller UR: user rights
	Specific	TD: types of collected user data PP: processing purpose of user data LB: legal basis of processing user data SP: storage period of user data ER: entity of user data receiver

Our research does not cover two specific types of notice elements in GDPR (i.e., automated user data processing, and oversea data transmission). We do not inspect the notice of automated user data process because this information is overlapped with the processing purpose of user data (PP), another notice element that has already been covered by our research. Besides, we did not discuss notice practice regarding overseas data transmission, because this data practice can be hardly monitored from the view of mobile apps. For example, the app could first collect user data and further transfer it out on the server side.

3 KEY RESEARCH SUBJECTS

In this research, we investigate RPNs to understand the extent to which users' "Right-to-be-Informed" is protected in real-world mobile apps. To achieve this goal, the following three key research questions are raised:

- **S1: Understanding the ecosystem.** To gain a comprehensive understanding of RPNs in mobile apps, we need to identify the parties involved in the RPN ecosystem, understand their relationships, and how they impact RPN practices.
- **S2: Finding the gap(s).** Since developers may intentionally or unintentionally violate the regulations of RPN, it is necessary to tell whether any gap exists in the wild and what the gaps are like if the answer is yes.
- **S3: Pinpointing the root cause(s).** If any gap exists, why does it exist? The unveiled causes can greatly help developers update their apps and be compliant with law requirements.

Roadmap. In the following sections, we first lay out the ecosystem of RPN in §4 to understand the relationships between different parties within it. Then, we elaborate on the methodology of investigating RPN practices in §5. Next, in §6, we present the identified gaps of RPN practices in the wild from analyzing a large number of popular mobile apps. Finally, in §7, we show the root causes behind the identified gaps through a notification campaign.

4 ECOSYSTEM OF RPN

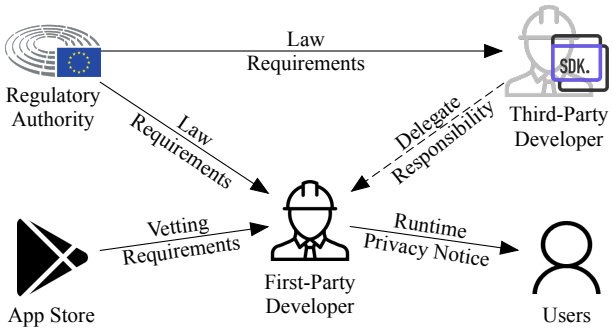


Figure 2: The ecosystem of RPN in mobile apps.

The ecosystem of RPN involves four parties: (1) regulatory authority (2) first-party and third-party developer, (3) app store and (4) mobile users. We present the whole ecosystem of RPN and the connections between these parties in Figure 2.

- **Regulatory Authority.** Regulatory authorities provide laws to guide user data processing behaviors in numerous areas, including the mobile area. In GDPR, the different responsibilities of "data controller" (normally the first-party developers) and "data processor" (normally third-party developers) [5, 29] are clearly distinguished while the data controller is typically the one responsible for fulfilling users' "Right-to-be-Informed". If any non-compliance of RPNs is detected, the data controller may face risks of app removal, fame loss, and even huge fines.

- **App Store.** App stores typically audit each mobile app submitted to them and decide whether to publish these mobile apps according to their vetting policies. Although app stores are not the only way mobile users can search and download mobile apps, they are the primary channels since they have a comprehensive range of available apps, and apps in them are typically audited to be secure and law-compliant [4, 31]. Thus, the vetting policies of app stores have a significant impact on the RPN practices in the wild.
- **First-party/Third-party developers.** Developers are in charge of the design and implementation of RPN. As shown in Figure 2, they can be further divided into first-party developers and third-party developers. Since developing mobile apps can turn into a complex and costly process, modular software development kits (SDKs for short) developed by third-party developers are often adopted to reduce the time and cost of app development. However, this may come at a privacy cost for mobile users as SDKs are accused of accessing or collecting a large number of user data, even without any transparency [42, 50, 72, 73, 86]. Therefore, third-party developers should provide necessary information to disclose their user data practices. While implementing RPNs, mobile app developers can utilize official APIs provided by mobile operating systems or consent management platform (CMP for short) SDKs. However, considering that app development is under the control of first-party developers, the responsibility of informing users of third-party SDKs' privacy-related behaviors is normally delegated to the first-party developers.
- **App Users.** Mobile app users are the destination of RPNs. Ideally, RPNs should be implemented by developers, audited by app stores, reviewed to be law-compliant by regulatory authorities, and then work as expected to protect users' "Right-to-be-Informed". Nevertheless, real situations can be complex. For instance, users may not see RPNs even if their data is accessed or collected. Besides, even when RPNs are provided, they may not comply with the vetting policies of app stores or GDPR. When these situations exist, mobile users can directly provide feedback to app developers and report to regulatory authorities or app stores if available.

Mis-alignment between Different Parties. Apart from the above four parties, their relationship with each other has a high impact on the real-world practices of RPN. To detail, while GDPR and the vetting policies of app stores both have requirements for conducting RPN practices, the stringency of their constraints is different. In the dimension of RPN quality, *GDPR requires that* RPN *should be provided at the time* when user data is obtained. In contrast, app stores vet the provision of in-app privacy notices without specifying the exact timing of providing privacy notices. Additionally, GDPR regulates that seven notice elements - IC, UR, TD, PP, LB, SP, ER are necessary to construct an RPN while app stores only require two notice elements, i.e., TD, PP. Moreover, as shown in Table 1, GDPR also regulates that the format of RPN should be clear while app stores have no relevant requirements. Therefore, compared to the regulations of GDPR, the vetting policies of app stores on RPNs are relatively lax. Given this situation, app developers face the dilemma of adhering to either GDPR or the vetting policies enforced by app stores.

To this end, we conclude that GDPR's requirements of providing proper RPNs are *stringent while the vetting policies of app stores are comparatively lax.* This fact could result in developers being uncertain about which regulations to comply with (as verified in §7.2), and users' "Right-to-be-Informed" cannot be reliably safeguarded.

5 AUTOMATED RPN ANALYSIS

As mentioned earlier, to better understand the implementation practices of RPNs, it is necessary to perform a large-scale, automated analysis over a large number of RPNs in real-world. To achieve this, we have designed and implemented RENO, an end-to-end pipeline by tackling a set of non-trivial challenges. Here we elaborate on the details of RENO.

5.1 Approach Overview

Figure 3 shows the overall workflow of RENO. First, mobile apps awaiting testing are explored and monitored in the testing environment of RENO, which outputs the triggered user interface (UI for short) and behaviors of user data collection. Meanwhile, with the triggered UIs, RENO employs an RPN detector (§5.3) to check whether RPNs are provided. If RPNs are identified, they will further be analyzed by RPN element analyzer (§5.4) to extract their contained notice elements. Finally, the triggered user data collection behaviors and RPN practices will be examined by the compliance checker (§5.5). In this way, RENO finally reports those missed RPNs or substandard RPNs.

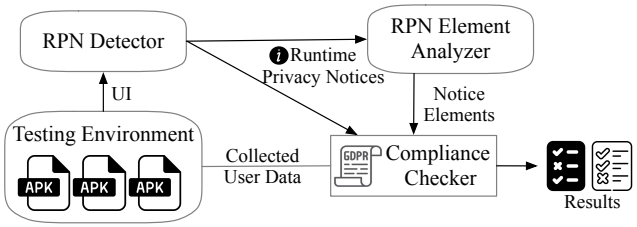


Figure 3: Overview of RENO.

5.2 Testing Environment

Basic Setup. To understand the (non-)compliance of RPN practices in a large scale of mobile apps, it is necessary to first automatically explore the app and monitor the data collection within the app. To this end, RENO sets up an automated testing environment for app exploration. The automation is based on Droidbot [62] - a state-of-the-art Android dynamic exerciser. Besides, RENO monitors user data collection behaviors by utilizing Frida [75] and MitmProxy [35]. Here, Frida is used to hook privileged APIs [24] of the Android framework that access sensitive user data. Mitmproxy is utilized to intercept and inspect the network traffic. The content (payload) of the traffic reveals part of the user personal data, as well as their intended receivers.

Capturing User Data Collection. With the above setup, RENO supports capturing and analyzing the data practices under the following four types: 1) data managed by the Android system with runtime permissions (e.g., contacts); 2) data managed by the Android system without runtime permissions (e.g., MAC address); 3) data related to user input (e.g., bank card number); 4) other personal data such as user cookies. The former two types of user personal data can be captured by monitoring specific system provided APIs [39], while the rest two are detected by analyzing the network traffic through a set of heuristics. In particular, we have collected a comprehensive list of privacy-related keywords from prior research [36, 57, 66, 69, 70], and use such keywords[2] to identify user personal data from network traffic. Due to space limitation, we leave the detailed implementations of identifying user data in Appendix C.

To evaluate the effectiveness of monitoring user data collection, we randomly selected 100 apps from our research dataset (in §6.1) and performed a manual verification. Each of the sampled apps was manually explored for 15 minutes and their user data access or collection behaviors were manually checked. By comparing the monitored results with the manual checking results, the testing environment is verified to achieve 96.08% F1-score in identifying user data collection.

5.3 RPN Detector

As mentioned earlier, identifying RPNs from a given app poses several unique challenges. Existing works mostly focused on analyzing privacy policies [37, 56, 63, 82, 89–91], which are easy to access (i.e., from the app description page in app stores) and well structured (i.e., their statements are typically presented in a complete sentence structure). However, RPNs are elusive as they are hidden inside the interactions between users and mobile apps. To address this issue, the key observation is that RPNs are typically presented in a specialized context to gain user attention, i.e., to be easily perceived. Thus, the appearance context along with its semantics could be a reliable sign of RPNs. Therefore, RENO identifies RPN through a customized binary machine learning classifier, which considers the whole text of a mobile UI page as the underlying features.

Training Dataset. To train the RPN detector, we conducted random sampling and testing on 750 popular mobile apps from the Google Play store in Germany. We manually explored each app for at least 15 minutes to ensure thoroughness. Each UI page and its corresponding layout files are collected for further analysis. Then, we take three experts working on mobile privacy to individually label whether these UIs are with RPN or not. If there is any discrepancy in the annotation results, the three experts will have a discussion to reach their final agreement. In fact, since the semantics of RPNs are very clear, most discrepancies are caused by overlooking by one of the experts. Based on the labeled data, we then constructed a training dataset consisting of 1,053 UI pages with RPNs, and 1,010 randomly selected normal UI pages. In the meantime, we perform a sentence-level labeling, by identifying the exact sentence describing RPN from the UI pages with RPNs. In total, 9,037 sentences are extracted from the selected UI pages, where 3,679 of them describe RPNs. Note that, our research did not take the types of RPNs as labels, as RENO is designed to identify law compliance issues of RPNs regardless of the RPN type.

Training Schemes and Performance Validation. During the training of the RPN detector, we tried two training schemes. The first one focuses on the granularity of UI page (i.e., taking the whole text of one UI page and its label as the training input) and the second one is at the granularity of sentence (i.e., taking one sentence and its label as the training input). We employed the pre-trained BERT [53] plus a linear layer and a softmax [87] as the binary classifier for our task. We choose BERT due to its well-known ability to capture the contextual semantic information for texts. In each training scheme, we conducted a ten-fold cross-validation to verify the effectiveness of the classifier. Specifically, the training dataset was divided into ten subsets and nine of them were used for training while the rest one was for validation. The entire process was automatically repeated ten times. Finally, the training scheme with UI page granularity is validated to be better (F1-score: 88.11%) than the other one (F1-score: 63.08%). Therefore, we choose to adopt the classifier trained with the former scheme as RPN detector.

5.4 RPN Element Analyzer

Furthermore, for those identified RPNs, it is necessary to accurately extract and interpret their contained notice elements. Prior related work mainly focuses on privacy policies and relies on the complete sentence structure of their statements to extract the corresponding key elements (e.g., data object, purpose). However, this basis does not hold true for RPNs since they are commonly seen to be presented with multiple short sentences or phrases, resulting in an incomplete and fragmented structure. To tackle this issue, we observed that the notice elements in RPNs exhibit fixed semantics and maintain consistency in both privacy policies and RPNs. Consequently, understanding the semantics of various notice elements in RPNs becomes feasible by referencing the easily accessible privacy policies. Moreover, the comprehended semantics can be leveraged to extract notice elements in RPNs more effectively.

To this end, the RPN element analyzer conducts notice element-level analysis through two successive stages: initially confirming the existence of a specific type of notice element, and upon affirmation, proceeding to extract it. The first stage is similar to the RPN detector, which is naturally a classification problem while the latter stage corresponds to the essence of named entity recognition (NER for short), which is an NLP method that extracts information from unstructured text. Particularly, six binary classifiers that respectively distinguish whether one of the six notice elements (i.e., IC, UR, PP, LB, SP, ER) exists in an identified RPN were developed. Additionally, the identification of the notice element - TD can be well achieved with the constructed keywords list as introduced in the testing environment. Then, with BIO (short for beginning, inside and outside) labeling [52], the NER-based notice element extractor for the latter stage was designed and implemented.

Training Dataset. Learning-based methodologies typically require a large amount of training data to acquire good performance. To meet this requirement, we follow the observation mentioned above and combine privacy notices extracted from the privacy policies of mobile apps in our research dataset with the training dataset of the

[2]The list of collected keywords are available in https://github.com/RenoProject2024/Reno/blob/main/privacy2keyword.json.

RPN detector. Finally, we constructed a new training dataset of 3,672 privacy notice sentences and labeled the types and content of their contained notice elements. For instance, an RPN says "To personalize our services for you, we also collect location data information when you are not using the app. You can revoke it anytime". After BIO labeling, it becomes "To [B-PP] personalize [I-PP] our [I-PP] services [I-PP] for [O] you [O] we [B-IC] also [O] collect [O] location [B-TD] data [I-TD] information [I-TD] when [O] you [O] are [O] not [O] using [O] the [O] app [O] You [B-UR] can [I-UR] revoke [I-UR] it [I-UR] any [I-UR] time [I-UR]".

Training Schemes. We trained the six binary classifiers in the same way as in RPN detector. Referring to the NER-based notice element extractor, its training scheme is few-shot learning (FSL) which requires comparatively fewer labeled training samples to fine-tune a pre-trained model to achieve competitive performance. In particular, we employed the pre-trained model - "BERT-Base uncased" [53]. To make it more specialized in RPNs, we followed one of its pre-train strategies (i.e., masked language modeling) to continue to train it on the privacy policies of mobile apps in our research dataset. With the model structure of BERT plus an additional conditional random field (CRF) [65], the NER-based notice element extractor was implemented and trained.

Performance Validation. After conducting ten-fold cross-validation, we validated that the six binary classifiers achieve F1-scores of 90.99%, 95.23%, 94.37%, 96.72%, 94.24%, and 97.88% respectively for identifying the existence of notice elements - PP, UR, IC, ER, LB, SP in an RPN. The performance of the notice element extractor was similarly validated with ten-fold cross-validation. The validation results confirmed that the notice element extractor achieves F1-scores of 91.42%, 89.77%, 90.73%, 89.24%, 81.96%, 97.67% and 90.63% respectively for extracting the content of notice elements - PP, UR, IC, ER, LB, SP, TD in RPNs. Therefore, the proposed RPN element analyzer demonstrates high performance in analyzing different notice elements of RPNs.

5.5 Compliance Checker

After identifying RPNs and extracting their notice elements, RENO needs to locate the concerned gaps between them and GDPR while considering user data collection behaviors of mobile apps. Particularly, these gaps are distributed in two perspectives mentioned before, i.e., existence and quality. On the one hand, if one app collects user data while providing no relevant RPNs, an existence gap is identified. On the other hand, if an RPN is provided but with poor content, format, or timing, then a quality gap is identified.

Map RPNs to Relevant User Data Collection Behaviors. Specifically, RENO employs the designed compliance checker to conduct a compliance analysis, which tries to map the identified user data collection with corresponding RPN behaviors and further figure out whether any gap exists. To achieve this goal, all the labeled and detected content of notice element - TD are manually analyzed to construct an ontology mapping organized by hypernym relationship[3]. The biggest advantage of doing so is that this ontology mapping can overcome the difficulty of mapping two statements with inconsistent granularity. For instance, when an identified RPN says "We collect your personal information for verifying your identity" and the collection of user email addresses is identified, these two behaviors are mapped since the user email address (hyponym) belongs to personal information (hypernym).

Check Existence Gaps. After RPNs are mapped with their relevant user data collection behaviors, the compliance checker can easily judge whether there exist the existence gaps, i.e., mobile apps collect user data while providing no relevant RPNs.

Check Quality Gaps. For detecting quality gaps, the proposed compliance checker first relies on the extraction results of the RPN element analyzer to understand whether the required notice elements are all provided. If any type of notice element is missing in an RPN, then a quality gap belonging to poor content is found.

Then, for the format requirement of RPNs, i.e., to be clear with no vagueness, the compliance checker adopts a frequency-based scheme to effectively locate the vague expressions of all these seven notice elements. Generally, for clear notice expressions, their content should be specific to their apps' concrete data processing behaviors and thus be unique. When it comes to vague notices, their expressions are common and similar to each other. For instance, instead of clarifying clear processing purposes of user data (PP) in an RPN, mobile apps are commonly seen to use similar expressions like "to work properly", "for special purposes" and so on. Therefore, the compliance checker performs a frequency analysis to pick out the commonly used expressions for each type of notice element, which are further manually cross-checked with three experts in this field to pick the vague ones. An expression is marked as ambiguous only if more than two of the three experts reach a consensus. Additionally, the compliance checker also refers to the corpus of vague words and sentences [61] to detect vague RPNs. Meanwhile, the constructed ontology mapping is used to determine if the expressions of the notice element - TD (i.e., the type of collected user data) are vague. Specifically, if the expression of TD has hyponyms in the ontology mapping, it indicates a vague expression.

Lastly, for the timing of RPNs, the compliance checker simply compares the timestamp when RPNs are provided and the moment when relevant user data collection occurs, to effectively judge whether RPNs are provided timely as required by GDPR.

Performance Validation. To verify the performance of RENO for finding gaps, we randomly sampled 99 apps from our research dataset (3 apps per category), manually explored each of them for an average of 15 minutes, and followed the requirements of GDPR to manually check their existence and quality gaps in provided RPNs. The manual verification triggered 196 RPNs and 83,392 user data collection behaviors, which are involved in 13,826 network flows and 56,067 invocations of privileged APIs. Finally, it is manually verified that RENO achieves 97.54% F1-score (95.28% precision and 99.91% recall) and 93.87% F1-score (90.26% precision and 97.78% recall) respectively for identifying the existence and quality gaps. While validating the performance of RENO and its components, we do not take into consideration the false negatives (i.e., missed RPN gaps) brought by the limited exploration ability of Droidbot since improving it in this aspect is an orthogonal research direction.

[3]Due to space limitation, the complete ontology mapping can be accessed at https://github.com/RenoProject2024/Reno/blob/main/Ontology.json

Table 3: Dataset used in our research, with 4,656 mobile apps distributed in Google Play stores of 19 EU countries.

EU Country	Austria	Belgium	Bulgaria	Croatia	Czech Republic	Denmark	Finland	France	Germany	Greece
# Apps	957	911	868	864	864	899	843	886	894	846
EU Country	Hungary	Ireland	Italy	Netherland	Poland	Portugal	Romania	Spain	Sweden	# Total
# Apps	854	869	865	865	850	844	839	869	872	4,656

6 UNDERSTANDING RPNS IN THE WILD

In this section, we aim to understand the whole picture of RPN in the wild. To achieve this goal, we apply RENO to figure out whether any gap exists between the RPN practices and GDPR.

6.1 Dataset Collection

Considering the 27 countries in the European Union (EU), we first select the most popular mobile apps in their Google Play stores according to Apptopia [18] - a well-known app analytics platform that provides app rankings across different countries. Due to network restrictions, we were unable to directly download mobile apps from Google Play stores in the 27 countries. Thus, with the list of selected mobile apps, we referred to APKCombo [3] which pulls app files (i.e., APK and OBB files) directly from Google Play Store [1]. We implement a scraper that mimics the app downloading behavior of regular users to automatically download these apps from APKCombo.

As shown in Table 3, we were able to access the popular mobile apps distributed in 19 of 27 EU countries. In particular, for each considered EU country, we selected the top 40 apps in each category (33 categories in total) and filtered out mobile apps that have no network permission or have not been updated for at least a year. By following this strategy, we finally crawled 5,123 unique mobile app ids with 4,656 apps successfully downloaded. Among the downloaded apps, 4,467 of them are unique and the remaining 189 apps are country-specific versions of 70 unique apps. Along with the downloaded mobile apps, their associated metadata (e.g., distributed in which country, link of the privacy policy, category and so on) are also obtained. Most failed-to-be-downloaded mobile apps were due to copyright infringement, since they were removed from Apkcombo as regulated by the Digital Millennium Copyright Act (DMCA) [14]. In total, 4,625 apps were successfully tested while the failed ones were due to compatibility issues.

Experiment Statistics. All experiments are conducted on six OnePlus 9 (Android 11) mobile devices and six supporting desktop computers (Windows 11). Our experiments last around 30 days with each mobile app in our research dataset tested for 30 minutes. Finally, 322 million logs of API invocation, 453,090 network flows and 1,030,529 runtime UI screenshots were collected during our empirical study, where 7,171 RPNs were identified. Since all the sampled apps for performance validation of RENO are randomly picked, the verification results can well provide a reasonable effectiveness estimation of RENO among the whole research dataset.

6.2 Non-compliance Gaps

With the analysis results of RENO upon our research dataset, we have summarized seven findings from the perspectives of existence and quality gaps, which comprehensively depict the status quo of RPN in the wild.

Existence Gaps. First, to check how serious the lack of RPNs is, we carefully analyzed the existence gaps detected by RENO. Among our research dataset, 4,615 (99.78%) mobile apps collected user data, but only 2,364 (51.22%) mobile apps provided RPNs. In total, there were 7,171 RPNs provided, where 27.41% were presented in runtime permission prompts, 20.80% were pop-ups, 23.02% were UI text labels, and 28.77% were hint pages.

More importantly, RPNs are significantly missed in real-world apps. In particular, 77.10% of user data collection lacks RPNs. Upon closer examination, the collection of user data that is managed with runtime permissions has best practices (91.61% provide RPNs) while it is worst for the collection of user data in "user input" category (11.72% provide RPNs). While collecting user data managed with runtime permissions should 100% have runtime permission prompt and thus 100% provide RPNs, the left 8.39% referred to resources that are not managed by runtime permissions to collect user data. For instance, an investigated app collected user location by utilizing an SDK that infers user location based on the connected Wi-Fi. Since the protection level of the required permission - *android.permission.ACCESS_WIFI_STATE* is normal (instead of dangerous), no runtime permission prompt was presented, which resulted in the lack of RPNs.

In addition, by randomly sampling five apps in each app category, we found that whether RPNs are provided or not has limited relevance to app functionality. Instead, mobile apps have the best RPN practices in scenarios involving runtime permission prompts, but falter in scenarios where "user input" related data is collected. The reason is that accessing or collecting user data managed by runtime permission necessarily triggers the prompts, which act as RPNs to inform users. In other scenarios, particularly when gathering data related to "user input", developers may assume that users can infer the purpose of data processing and therefore not provide RPNs.

Finding I: RPNs *are significantly missed in real-world apps.*

Second, since network requests that send out user data are strong evidence of user data collection, we further identify whether such user data collection is initiated by the first-party or a third-party. By referring to related work [59] and teasing out all identified domains that have more than 1,000 identified network flows in our research, we manually verified them to create a curated third-party domain

Table 4: The availability of privacy notice resources offered by third-party services. *Availability* refers to whether a third-party service provides notice regarding their collection of user data. *Completeness* refers to whether the provided notice is complete in terms of `IC, UR, TD, PP, LB, SP, ER`. *Source* refers to the format of provided notice. *Remind* refers to whether third-party services explicitly remind first-party developers to inform users on their behalf.

Service Domain	Availability	Completeness	Source	Remind
googleapis.com	-	×	/	✓
doubleclick.net	✓	×	privacy policy	✓
facebook.com	✓	×	privacy policy	✓
appcenter.ms	✓	×	document	×
sentry.io	✓	×	document	✓
unity3d.com	✓	✓	privacy policy	✓
adjust.com	×	×	/	×
yandex.net	✓	×	document	✓
amazonaws.com	×	×	/	✓
branch.io	✓	×	privacy policy	×

list[4], which has 27 third-party services with 33 domains. Based on this list, 383,003 network flows were identified to send out the collected user data where 56.77% of them were sent to third-party services. Furthermore, 89.50% of user data collection initiated by third-party services in network flows lack RPNs while it is 86.85% for the first-party. Additionally, regarding the lack of RPNs, third parties are responsible for 46.66% of them while the first party is responsible for the left. More importantly, it is revealed that no matter if they are popular or long-tailed third-party services, they exhibit similar bad practices of lacking RPNs.

To investigate this phenomenon, we studied whether third-party services provide easily accessible and GDPR-compliant privacy disclosures to ease the burden of first-party developers for notifying users. Table 4 shows the results of top-10 third-party service domains[4] that have the most behaviors of lacking RPNs. It can be seen that most third-party services owning these domains do provide privacy disclosures for their user data processing behaviors and actively remind first-party developers to notify users. However, their privacy disclosures have different formats and are even not complete (i.e., contain all notice elements required by GDPR). While numerous Google API services lack explicit notice information to guide first-party developers in notifying users about privacy practices upon integration, a select few utilize the "data safety" (DS) format as a privacy notice (e.g., Google Maps SDK [26]) [34]. Different from the requirements of GDPR, such DS notice practice works when users are browsing candidate mobile apps in the Google Play store (instead of the time when users are interacting with mobile apps). Besides, DS focuses more on data collection and sharing while ignoring other notice elements regulated in GDPR. Since a mobile app may integrate various third-party SDKs (around 18 on average in an Android app [23]), a first-party developer or team needs to identify what functionalities of the picked SDKs are adopted and whether they involve user data processing, and finally refer to their official documents to check if any privacy notice information is available. Such a heavy burden of providing RPNs for the data collection of third-party services can hinder first-party developers from doing so.

[4]The complete list and surveyed results can be accessed at https://docs.google.com/spreadsheets/d/1lTS_Q8msqH4j4T_GTYZa72uOCbQzLW7kglX49Up5PTE.

Finding II: *First-party developers face the dilemma where third-party developers delegate to them the responsibility of notifying users and provide no complete disclosures of user data processing.*

Third, while reviewing the identified RPNs, it is noticed that some mobile apps adopt CMP (consent management platform) SDK to help implement the required RPNs. To comprehensively understand the role that CMP SDKs play in RPN practices, an authoritative ranked CMP list provided by IAB Europe [9] and relevant researches [59] are referred to, where a CMP list containing 39 SDKs is formed and 7 of them that focus on mobile apps with available documents have been picked out. The selected CMP list includes AppConsent [79], Didomi [48], Quantcast [58], Osano [76], Clarip [44], Quadrant [38] and Sourcepoint [83].

By checking the existence of these selected CMP SDKs, we filtered out from our research dataset 53 (1.14%) mobile apps that adopt CMP SDK (which is in line with the finding of [59]). When looking into the RPN practices in these apps, it is clear that CMP SDKs benefit mobile apps in providing required RPNs since the ratio of user data collection lacking RPNs is 77.66% which drops to 38.11% when mobile apps adopt CMP SDKs.

Finding III: *While CMPs do benefit the situation of* RPN *practices in real-world mobile apps, they play a very limited role due to their low adoption rate and actual usage focus.*

Quality Gaps. After understanding the existence gaps of RPNs, we further checked the quality of identified RPNs respectively from aspects of content, format, and timing.

As shown in Table 5, the proportion of the seven types of notice elements in all identified RPNs (7,171) is calculated. Particularly, the notice element - TD is the most frequently mentioned one in identified RPNs. The top-3 notice elements in identified RPNs are respectively `TD, IC, PP`, which means mobile users are most commonly exposed to these three notice elements in RPNs. Subsequently, we evaluate the number of provided notice elements. The results are shown in Table 6, which demonstrate that most RPNs (72.43%) provide either two or three required notice elements. Besides, all identified RPNs have one or more missing notice elements. Generally, when there are two or more notice elements in RPNs, the more elements an RPN has, its proportion accounting in real-world practices is lower.

Moreover, we further looked into whether and how CMP SDKs improve the quality of the identified RPNs. As shown in Figure 4, by respectively comparing the rates of different notice elements in apps using and not using CMP SDKs, it demonstrates that the quality of RPN is better when adopting CMP SDKs. From the perspectives of mathematical statistics, mobile apps using CMP SDKs provide 3.32 notice elements in each RPN on average while this number is 2.38 for apps without adopting CMP SDKs. Therefore, CMP SDKs do benefit the situation of RPN but still not enough.

Finding IV: *Even when* RPNs *are provided, users are not guaranteed to get notification of all necessary notice elements as required by GDPR.*

Table 5: The proportion of different notice elements in the identified RPNs. Note that different instances of the same notice element type in one sentence are accounted as one.

Notice Elements	Number	Proportion
Processing Purpose of User Data (PP)	4,191	58.44%
User Rights (UR)	432	18.27%
Identity of Data Controller (IC)	1,736	73.43%
Entity of User Data Receiver (ER)	759	10.58%
Legal Basis of Processing User Data (LB)	641	8.94%
Storage Period of User Data (SP)	93	1.30%
Types of Collected User Data (TD)	5,664	78.98%

Table 6: Distribution of notice elements in RPNs.

Number of Notice Elements	Number of RPNs
1	998 (13.92%)
2	3,492 (48.70%)
3	1,702 (23.73%)
≥ 4	979 (13.65%)

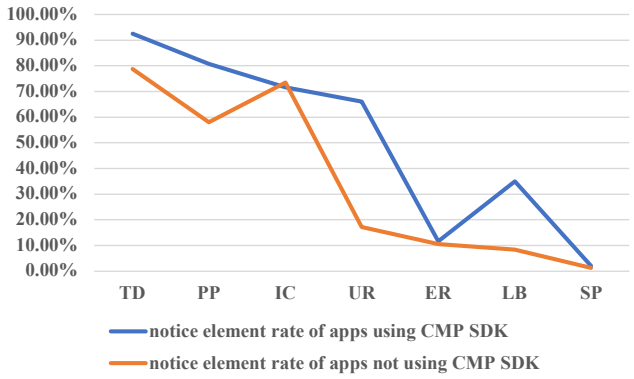


Figure 4: The presence rates of notice elements in mobile apps that adopted CMP SDKs versus those that did not.

Furthermore, according to the compliance analyzer of RENO, the concrete content of identified RPNs was checked to see if they satisfy the format requirements of GDPR, i.e., being clear with no vagueness.

Among 2,364 mobile apps that provide RPNs, 624 (26.40%) apps have 1,623 (22.63%) RPNs are with vague descriptions. Among these vague RPNs, the range of affected types of notice elements involves PP, IC, ER, TD. Table 7 presents the samples and proportions of these notice elements using vague expressions.

Finding V: *Approximately a quarter of the identified* RPNs *utilize vague expressions, which substantially reduces the transparency of user data processing.*

Finally, we carefully investigated the timing of providing RPNs. By comparing the time when user data collection happens and the moment when a relevant RPN is provided, RENO identified that 1,343(56.81%) mobile apps provide the required RPNs after user data is already collected. Among all informed collection of user data[5], 37.40% of them were conducted before mobile apps provided corresponding RPNs. The detailed "collect before providing notice" situation of different types of user data is listed in Table 8.

Finding VI: *The dark pattern of "collect before providing notice" is serious.*

Regarding the timing of providing RPNs, another type of improper RPN practice – "detailed notice after refusal" is identified. With the customized app exploration of RENO, i.e., "refusing runtime permission request first and allowing afterward", 154 mobile apps were identified to provide detailed RPNs with more notice elements only after users refuse to grant runtime permissions. Particularly, it was found that 861 runtime permission requests of user data merely relied on runtime permission prompts to inform users. Meanwhile, 158 runtime permission requests were re-requested with detailed RPNs provided after being refused.

Such phenomenon is denoted as "detailed notice after refusal". After investigating its code implementations, we found that this phenomenon is suggested in the official documents of Android. Particularly, the Android system provides a specialized API [54] to automatically help developers determine whether to show a rationale for explaining why their apps request the runtime permissions. This API only returns true when runtime permission is refused (but not forbidden, i.e., users did not click the "deny and don't ask again" button) and re-requested. Namely, the Android system assumes only in this situation, users may feel confused about why apps request certain runtime permissions, and it is the right time for developers to provide explanations. However, since this API is optional, app developers may not adopt it at all and thus completely rely on the runtime permission prompt to inform users, which only contains the notice element of TD (i.e., the types of collected user data) and IC (i.e., the identity of data controller). Besides, even if this API is adopted, the notice element suggested to be supplemented is PP, i.e., the processing purpose of user data. In fact, after permission refusal, 158 RPNs provided detailed PP as suggested and 144 of them were sufficiently clear. Thus, as GDPR requires the existence of seven notice elements, users' "Right-to-be-Informed" may still be violated since they get no sufficient notice.

Finding VII: *Instead of proactively providing qualified RPNs, a few developers adopted "detailed notice after refusal" strategy to inform users.*

Based on these findings, we have correspondingly provided compliance suggestions in Appendix A.

7 RQ3: WHY DO THE GAPS EXIST?

We responsibly notified the affected app developers, which has two main goals: first, we need to inform them of the detected non-compliance risks of RPN. Second, we would like to gain insights into understanding why the identified gaps exist.

[5]In this aspect, each type of privacy collection behavior within an app is tallied as one.

Table 7: Identified vague notice elements in our dataset.

Notice Elements	# Affected Apps	# Vague RPNs	Vague Expression Samples
Processing Purpose of User Data (PP)	68 (2.88%)	127 (1.77%)	*"for special/commercial/marketing/analytical/advertising purpose"*
Identity of Data Controller (IC)	107 (4.53%)	159 (2.22%)	*"partner and vendor"*
Entity of User Data Receiver (ER)	89 (3.76%)	168 (2.34%)	*"third party and country"*, *"partner and vendor"*
Types of Collected User Data (TD)	546 (23.10%)	1,452 (20.25%)	*"personal information"*, *"permission"*, *"device information"*

Table 8: Prevalence of "collect before providing notices" for different types of user data in collected RPNs.

User Data Type	# Noticed	Collected Before Notice
w/ Runtime Permission	1,996	493(24.70%)
w/o Runtime Permission	869	796(91.60%)
User Input	824	84(10.19%)
Others	244	98(40.16%)

7.1 Notification Campaign

In particular, we extracted the email addresses app developers submitted to the Google Play store and described GDPR requirements along with mobile apps' non-compliance behaviors in the notification emails. More importantly, the following questions were asked to try to gain insights into understanding why the gaps exist.

- Were you aware of implementing RPNs each time when user data is accessed or collected?
- Could you share insights into the specific choices in implementing RPNs?
- What strategies are you exploring to enhance your app's RPN implementations? What supports would be beneficial for you?

The notification campaign was finished before December 7, 2023. During this process, we merged the notification emails sent to the same developer and finally sent out 4,399 emails. Among them, 4,169 emails were successfully sent out. The failures are mainly due to being rejected by the recipient, remote recipients' servers being busy or down, being filtered out by the firewall, and so on. Till now, we have received 821 unique replies, where 60 of them are not automatic responses (i.e., manual responses). Most of the manual responses neither confirm nor deny our reports with similar sayings like *"We appreciate your feedback regarding GDPR compliance of our app. It was already forwarded to our developers for further analysis."*, and no further replies are received. Meanwhile, few responses denied our report and refused to provide more details. Most importantly, after filtering out these meaningless responses, we got 13 replies confirming our reports and providing meaningful feedback, which can greatly help us understand the causes behind the identified gaps.

7.2 Responses from Developers

By referring to the meaningful responses provided by mobile app developers, we summarize them as follows.

Developers Prefer to Follow App Store rather than GDPR. Two mobile app developers tend to make their app aligned with Google Play's guidelines instead of GDPR to avoid direct risks (e.g., being removed from app stores). For instance, one of our received feedback states as follows.

> **Response I**: *"While we strive to follow Google Play's guidelines for app deployment and ensure GDPR compliance, we acknowledge that there may have been instances where our implementation of the runtime privacy notice did not strictly adhere to GDPR regulations."*

Compared to regulatory authorities, the app store, being closer to app developers, offers more detailed, practical, and actionable regulations for implementing RPNs in mobile apps. Therefore, it is easy to understand that app developers prefer to follow the app store's guidelines, which results in the identified quality gaps.

Developers Consider Privacy Policy is Enough. Three app developers doubt the necessity of providing RPNs. For example, one of our received responses states as follows:

> **Response II**: *"The right to be informed under the GDPR is typically realized by making the Privacy Policy available to the users at the time of personal data collection. We at app A(anonymized) do provide the users with the link to access the Privacy Policy at all the most important steps of using the application (whenever we start collecting personal data for any new purpose), and later on, the policy is accessible at any time in the user's account".*

Nevertheless, according to Art.13 of GDPR (i.e., providing privacy notices *"at the time when personal data are obtained"* [21]) and the audit rules of Google Play store (i.e., *"You must provide an in-app disclosure of your data access, collection, use and sharing"* [27]), RPNs are necessary and their content cannot only be presented in privacy policies. In line with the spirit of GDPR, RPNs should be provided at each time user data is accessed or collected instead of only at the "most important steps of using the application". Actually, it is a balance among development cost, user experience and notice effect, which most app developers find hard to achieve.

Developers are Working on the Improvements. The rest feedback acknowledged the potential issues on RPN, in the meantime, they emphasized that they are working to make their mobile apps compliant with GDPR with their best effort. For instance, one response states as follows:

> **Response III**: *"GDPR will be implemented within the app in the second half of this year and all subsequent apps will strictly comply with it".*

As can be seen, app developers who acknowledged our reports either were preparing an updated app version or had a clear schedule to make their apps compliant with GDPR.

7.3 Dilemma in RPN Compliance

Through multiple rounds of communication with developers, it was found that they face a dilemma of balancing user experience and law compliance. Regarding the discrepancies between GDPR and RPN practices in the real world, their existence has its rationality. In particular, app developers need to consider user experience during app development and it may have conflicts with law-compliance-related implementations. One typical example is the frequency and timing of providing RPNs. As regulated in GDPR, privacy notices should be provided *at the time when personal data are obtained* [21]. However, considering the scale and frequency of modern mobile apps' accessing or collecting user data, a large number of (or frequently provided) RPNs can annoy mobile users and consequently bring a negative user experience. Based on this dilemma, RENO judges whether RPNs are provided before or at the same time when user data is collected. If the answer is yes, then the mobile app complies with the law.

8 DISCUSSION

Ethics Consideration. We carefully manage our research activities to ensure that they stay within legal and ethical boundaries. In particular, all the tested mobile apps were signed in with our researchers' accounts. RENO only monitors network requests to confirm whether any user data (belonging to our researchers) is sent out. Besides, the identified RPNs do not contain concrete user data values such as a specific user name. Therefore, our research did not involve data from other users and caused no harm. Besides, all communication emails during the notification campaign were kept confidential (they can only be accessed by our researchers). Before our research, we consulted the IRB staff in our institution, the research was approved by our IRB and classified as with "minimal risk".

Alternative Schemes of Implementing RENO. Instead of improving the techniques that RENO adopts, it is possible to substitute them with other candidate ones. One promising scheme is to utilize the advantage of a large language model (LLM for short) to help RENO better identify and analyze RPNs. Nevertheless, such an idea is dropped after a thorough analysis. On the one hand, the success of LLM relies on big computing power and big data, which cost too much to start from scratch. On the other hand, while referring to open-sourced or commercial LLM models is practical, it has been verified [43] that the performance of the two most widely used LLMs - GPT-3.5 and GPT-4 can vary greatly over time. Thus, if adopted, their unstable performance can significantly affect RENO in practice. Besides, we also test ChatGPT [22] by using 100 RPNs to perform prompt engineering [30] and compare its performance with RENO in identifying RPN (90.00% vs 92.67%), judging whether a notice element exists (80.00% vs 94.90%), and extracting notice elements (82.16% vs 90.20%). The comparison results show that RENO is noticeably better than prompted ChatGPT.

Limitations. As demonstrated in this paper, RENO can be directly deployed for compliance checks in real-world scenarios and achieve large-scale detection through parallel deployment. However, RENO has limitations inherited from its adopted techniques, which can bring false negatives and false positives. For instance, the limited exploration ability of Droidbot [62] can cause RENO to miss RPNs, resulting in an underestimation of actual RPN practices in the wild. Meanwhile, the false positives of employed technologies (e.g., NER) can be propagated to RENO as well. We acknowledge that achieving a perfect balance between false positives and false negatives is challenging. However, since RENO provides non-compliant RPNs with contextual information such as UI screenshots and data collection behaviors, eliminating false positives is quite feasible with little human effort. Therefore, in real-world deployment, RENO could be tuned towards tolerating more false positives and maintaining minimal false negatives. Moreover, RENO is validated to achieve qualified performance, which is sufficient to understand RPN practices in our research. Further improving the performance of these adopted techniques would be orthogonal to our research.

Additionally, for evaluating RPN quality, RENO has limitations when applied in other areas instead of EU. For instance, it cannot evaluate conspicuity or identify contradictory content in RPNs, vital for CCPA[7], COPPA[8] and PIPL[33] compliance. Alternatives can be user studies that identify inattentive RPN patterns for detection and existing work[36] offers guidance on handling contradictory content. Besides, developers may provide simplified RPNs with links to more detailed information like privacy policies. However, RENO will not assess whether the data behind these links complies with GDPR because clicking on them will take users to websites, which is beyond the scope of RPN. Regarding the application scenario of RENO, we have provided further discussion in Appendix B due to space limitation.

9 RELATED WORK

Our research is closely related to topics of privacy compliance check on web and mobile platforms, such as cookie notices [67, 74, 85], privacy policies [37, 63, 82, 89–91], privacy labels [60], and consent management [59, 72, 73].

Compliance Check on Web Platforms. Particularly, for the Web platform, Degeling et al. [46] monitored the prevalence of privacy policy among popular websites in the EU, which found that 84.5% websites in Europe have privacy policies and estimated that 62.1% of them present cookie consent notices. Regarding cookie consent notices, Utz et al. [85] studied various designs of cookie notices and their influence on users' choices by conducting a study with real website visitors. Furthermore, attracted by the cookie consent notice interfaces, Matte et al. [67] and Midas et al. [74] studied cookie banners implemented by CMPs and revealed that dark patterns (i.e., interface designs that nudge users to make privacy-unfriendly choices) along with implied consent are ubiquitous. However, different from these works, our research focuses on mobile platforms and RPNs (instead of privacy policies) to investigate whether users' "Right-to-be-Informed" is well protected.

Compliance Check on Mobile Apps. When it comes to the mobile platform, one area of research is the identification of potential GDPR violations in privacy policies or labels, e.g., their privacy-related statements are inconsistent with apps' real behaviors [37, 60, 63, 82, 89–91]. In many cases, the findings of these works cannot be equated to RPN compliance, due to their fundamental requirements and application scenarios. For example, privacy policies are not timing/contextual sensitive, since they are available upon app publication or when the app launches, regardless of when

user data is collected. Another line of research [59, 72, 73] investigated the effectiveness of consent management in mobile apps, which revealed various violations against GDPR, e.g., deceiving users into accepting all data sharing. In contrast, this paper studies the real practices of RPNs in the wild (instead of judging whether consent choice is provided and works as expected) and reveals gaps between them and GDPR. Pertaining to permission prompts in mobile apps, prior works [40, 41, 49, 64, 80, 84] have explored factors influencing user decisions (i.e., grant or not), whereas our work reveals the legal compliance of RPNs.

10 CONCLUSION

This paper presents a comprehensive exploration of the RPN ecosystem, enhancing the community's insights into this domain. In order to comprehend the prevalence of RPNs in real-world mobile apps, we introduce an automated pipeline - RENO, enabling the assessment of disparities between RPN practices and GDPR standards. Through a large-scale experiment, we uncover widespread gaps and multiple noteworthy findings. Responsibly notifying affected developers and analyzing their feedback allows us to elucidate the root causes behind these gaps. We believe that our study can serve as a valuable resource for developers, aiding them in aligning their apps with legal regulations and concurrently enhancing the transparency of private data practices for app users.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of the paper. This work was supported in part by the National Key Research and Development Program (2021YFB3101200) and National Natural Science Foundation of China (62172104, 62172105, 61972099, 62102093, 62102091). Zhemin Yang was supported in part by the Funding of Ministry of Industry and Information Technology of the People's Republic of China under Grant TC220H079. Min Yang is the corresponding author, and a faculty of Shanghai Institute of Intelligent Electronics & Systems, and Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China.

REFERENCES

[1] 2023. About us - Apkcombo.com. https://apkcombo.com/about.
[2] 2023. Amended Act on the Protection of Personal Information. https://www.ppc.go.jp/files/pdf/APPI_english.pdf.
[3] 2023. Apkcombo Apk Downloader. https://apkcombo.com/.
[4] 2023. App Store Review Guidelines. https://developer.apple.com/app-store/review/guidelines/.
[5] 2023. Art. 4 GDPR Definitions. https://gdpr-info.eu/art-4-gdpr/.
[6] 2023. California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa.
[7] 2023. CCPA - Right to Know. https://oag.ca.gov/privacy/ccpa#sectionc.
[8] 2023. Children's Online Privacy Protection Rule ("COPPA"). https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa.
[9] 2023. CMP List - IAB Europe. https://iabeurope.eu/cmp-list/.
[10] 2023. Code of Virginia Code - Chapter 53. Consumer Data Protection Act. https://law.lis.virginia.gov/vacode/title59.1/chapter53/.
[11] 2023. Consumers' Right to Know What Personal Information is Being Collected. https://thecpra.org/#1798.110.
[12] 2023. Cybersecurity Law of the People's Republic of China. http://www.cac.gov.cn/2016-11/07/c_1119867116.htm.
[13] 2023. Data Protection Act 2018. https://www.legislation.gov.uk/ukpga/2018/12/part/3/chapter/2.
[14] 2023. Digital Millennium Copyright Act. https://en.wikipedia.org/wiki/Digital_Millennium_Copyright_Act.
[15] 2023. Duty to provide information. https://www.edoeb.admin.ch/edoeb/en/home/datenschutz/grundlagen/informationspflicht.html.
[16] 2023. Federal Data Protection Act (BDSG). https://www.gesetze-im-internet.de/englisch_bdsg/englisch_bdsg.html.
[17] 2023. Fedlex. https://www.fedlex.admin.ch/eli/cc/2022/491/en.
[18] 2023. Free app store rank data for Google Play. https://apptopia.com/store-insights/top-charts/google-play.
[19] 2023. GDPR - Right to be Informed. https://gdpr-info.eu/issues/right-to-be-informed/.
[20] 2023. General Data Protection Regulation. https://gdpr-info.eu/.
[21] 2023. Information to be provided where personal data are collected from the data subject. https://gdpr-info.eu/art-13-gdpr/.
[22] 2023. Introducing ChatGPT. https://openai.com/blog/chatgpt.
[23] 2023. Mobile App Tools (SDKs) Simplified: How to Choose the Best For Your App. https://appguardians.com/blog/mobile-sdks-simplified-how-to-choose-the-best-for-your-app/..
[24] 2023. Package Index | Android Developers. https://developer.android.com/reference/packages.
[25] 2023. Personal Data Protection Act 2012 - Singapore Statutes Online. https://sso.agc.gov.sg/Act/PDPA2012.
[26] 2023. Prepare for Google Play's data disclosure requirements. https://developers.google.com/maps/documentation/android-sdk/play-data-disclosure?hl=en.
[27] 2023. Preview: User Data. https://support.google.com/googleplay/android-developer/answer/13316080.
[28] 2023. Privacy Act 1988. https://www.legislation.gov.au/Details/C2021C00452.
[29] 2023. Privacy and Security in Firebase. https://firebase.google.com/support/privacy?hl=en#firebase_support_for_gdpr_and_ccpa.
[30] 2023. Prompt Engineering Guide. https://www.promptingguide.ai/zh/introduction/tips.
[31] 2023. Providing a safe and trusted experience for everyone. https://play.google.com/about/developer-content-policy/?hl=en.
[32] 2023. The Data Protection, Privacy and Electronic Communications (Amendments etc) (EU Exit) Regulations 2019. https://www.legislation.gov.uk/uksi/2019/419/schedule/1.
[33] 2023. The Personal Information Protection Law of the People's Republic of China. https://www.gov.cn/xinwen/2021-08/20/content_5632486.htm.
[34] 2023. Understand app privacy security practices with Google Play's Data safety section. https://support.google.com/googleplay/answer/11416267?hl=en&visit_id=638308615428898834-2971702574&p=data-safety&rd=1.
[35] Cortesi Aldo, Hils Maximilian, and Raumfresser. 2021. Mitmproxy - an interactive HTTPS proxy. https://mitmproxy.org/.
[36] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. 2019. PolicyLint: investigating internal privacy policy contradictions on google play. In 28th USENIX security symposium (USENIX security).
[37] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. 2020. Actions speak louder than words: Entity-Sensitive privacy policy and data flow analysis with PoliCheck. In 29th USENIX Security Symposium (USENIX Security).
[38] Appen. 2024. Quadrant | App Monetisation and Consent Management Platform SDK. https://www.quadrant.io/app-monetisation-and-consent-management-platform.
[39] Kathy Wain Yee Au, Yi Fan Zhou, Zhen Huang, and David Lie. 2012. Pscout: analyzing the android permission specification. In 2012 ACM Conference on Computer and Communications Security (CCS).
[40] Rawan Baalous and Ronald Poet. 2020. Factors Affecting Users' Disclosure Decisions in Android Runtime Permissions Model. In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).
[41] Rebecca Balebako, Florian Schaub, Idris Adjerid, Alessandro Acquisti, and Lorrie Cranor. 2015. The impact of timing on the salience of smartphone app privacy notices. In 5th annual ACM CCS workshop on security and privacy in smartphones and mobile devices.
[42] Ravi Bhoraskar, Seungyeop Han, Jinseong Jeon, Tanzirul Azim, Shuo Chen, Jaeyeon Jung, Suman Nath, Rui Wang, and David Wetherall. 2014. Brahmastra: Driving Apps to Test the Security of Third-Party Components. In 23rd USENIX Security Symposium (USENIX Security).
[43] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is ChatGPT's behavior changing over time? arXiv preprint arXiv:2307.09009 (2023).
[44] Clarip. 2024. Universal Consent Management Platform - Clarip. https://www.clarip.com/consent-management.
[45] Federal Trade Commission et al. 2013. Mobile privacy disclosures: Building trust through transparency. USA: Federal Trade Commission (2013).
[46] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. 2018. We value your privacy... now take some cookies: Measuring the GDPR's impact on web privacy. arXiv preprint arXiv:1808.05096 (2018).
[47] Kenan Degirmenci. 2020. Mobile users' information privacy concerns and the role of app permission requests. International Journal of Information Management 50 (2020), 261–272.

[48] DIDOMI. 2024. MAKE PRIVACY PART OF YOUR STRATEGY. https://www.didomi.io/.
[49] Yusra Elbitar, Michael Schilling, Trung Tin Nguyen, Michael Backes, and Sven Bugiel. 2021. Explanation beats context: The effect of timing & rationales on users' runtime permission decisions. In 30th USENIX Security Symposium (USENIX Security).
[50] Álvaro Feal, Julien Gamba, Narseo Vallina-Rodriguez, Primal Wijesekera, Joel Reardon, Serge Egelman, Juan Tapiador, et al. 2020. Don't accept candies from strangers: An analysis of third-party SDKs. In Computers, Privacy and Data Protection Conference (CPDP).
[51] Geoffrey A. Fowler. 2022. I tried to read all my app privacy policies. It was 1 million words. https://www.washingtonpost.com/technology/2022/05/31/abolish-privacy-policies/.
[52] Geeksforgeeks. 2024. Inside-outside-beginning (tagging). https://www.geeksforgeeks.org/nlp-iob-tags/.
[53] Google. 2023. TensorFlow code and pre-trained models for BERT. https://github.com/google-research/bert.
[54] Google. 2024. Request Runtime Permissions | Android Developers. https://developer.android.com/training/permissions/requesting?hl=en#explain.
[55] Google. 2024. Runtime Permissions | Android Open Source Project. https://source.android.com/docs/core/permissions/runtime_perms?hl=en.
[56] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In 27th USENIX Security Symposium (USENIX Security 18).
[57] Jianjun Huang, Zhichun Li, Xusheng Xiao, Zhenyu Wu, Kangjie Lu, Xiangyu Zhang, and Guofei Jiang. 2015. SUPOR: Precise and scalable sensitive user input detection for android apps. In 24th Usenix Security Symposium (USENIX Security).
[58] InMobi. 2024. Foster Trust. Unlock Revenue. https://www.quantcast.com/products/choice-consent-management-platform/.
[59] Simon Koch, Benjamin Altpeter, and Martin Johns. 2023. The OK Is Not Enough: A Large Scale Study of Consent Dialogs in Smartphone Applications. In 32nd USENIX Security Symposium (USENIX Security).
[60] Simon Koch, Malte Wessels, Benjamin Altpeter, Madita Olvermann, and Martin Johns. 2022. Keeping privacy labels honest. Proceedings on Privacy Enhancing Technologies 4, 486-506 (2022), 2–2.
[61] Logan Lebanoff and Fei Liu. 2018. Automatic detection of vague words and sentences in privacy policies. arXiv preprint arXiv:1808.06219 (2018).
[62] Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. 2017. Droidbot: a lightweight ui-guided test input generator for android. In 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C).
[63] Shuang Liu, Baiyang Zhao, Renjie Guo, Guozhu Meng, Fan Zhang, and Meishan Zhang. 2021. Have you been properly notified? automatic compliance analysis of privacy policy text with GDPR article 13. In International World Wide Web Conference 2021 (WWW).
[64] Xueqing Liu, Yue Leng, Wei Yang, Wenyu Wang, Chengxiang Zhai, and Tao Xie. 2018. A large-scale empirical study on android runtime-permission rationale messages. In 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).
[65] lonePatient. 2023. Chinese NER(Named Entity Recognition) using BERT(Softmax, CRF, Span). https://github.com/lonePatient/BERT-NER-Pytorch.
[66] Sunil Manandhar, Kaushal Kafle, Benjamin Andow, Kapil Singh, and Adwait Nadkarni. 2022. Smart Home Privacy Policies Demystified: A Study of Availability, Content, and Coverage. In 31st USENIX Security Symposium (USENIX Security).
[67] Célestin Matte, Nataliia Bielova, and Cristiana Santos. 2020. Do cookie banners respect my choice?: Measuring legal compliance of banners from iab europe's transparency and consent framework. In 2020 IEEE Symposium on Security and Privacy (S&P).
[68] Aleecia M McDonald and Lorrie Faith Cranor. 2008. The cost of reading privacy policies. Isjlp 4 (2008), 543.
[69] Yuhong Nan, Min Yang, Zhemin Yang, Shunfan Zhou, Guofei Gu, and XiaoFeng Wang. 2015. Uipicker: User-input privacy identification in mobile applications. In 24th Usenix Security Symposium (USENIX Security).
[70] Yuhong Nan, Zhemin Yang, Xiaofeng Wang, Yuan Zhang, Donglai Zhu, and Min Yang. 2018. Finding Clues for Your Secrets: Semantics-Driven, Learning-Based Privacy Discovery in Mobile Apps.. In Network and Distributed System Security Symposium (NDSS).
[71] CET News. 2020. Many people accept privacy policies without reading them, study reveals. https://thepaypers.com/payments-general/many-people-accept-privacy-policies-without-reading-them-study-reveals--1245901.
[72] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. 2021. Share First, Ask Later (or Never?) Studying Violations of GDPR's Explicit Consent in Android Apps. In 30th USENIX Security Symposium (USENIX Security).
[73] Trung Tin Nguyen, Michael Backes, and Ben Stock. 2022. Freely given consent? Studying consent notice of third-party tracking and its violations of GDPR in android apps. In 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS).
[74] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark patterns after the GDPR: Scraping consent pop-ups and demonstrating their influence. In 2020 CHI conference on human factors in computing systems(CHI).
[75] Oleavr. 2021. Frida binary instrumentation toolkit. https://frida.re.
[76] Osano. 2024. The Intuitive Data Privacy Platform for Simplifying Compliance | Osano. https://www.osano.com/.
[77] Shidong Pan, Zhen Tao, Thong Hoang, Dawen Zhang, Tianshi Li, Zhenchang Xing, Sherry Xu, Mark Staples, Thierry Rakotoarivelo, and David Lo. 2024. {A New Hope}: Contextual Privacy Policies for Mobile Applications and An Approach Toward Automated Generation. arXiv preprint arXiv:2402.14544 (2024).
[78] Harvard Business Review. 2015. Customer Data: Designing for Transparency and Trust. https://hbr.org/2015/05/customer-data-designing-for-transparency-and-trust.
[79] SFBX. 2024. AppConsent®, the transparency-based consent management platform. https://appconsent.io/en.
[80] Bingyu Shen, Lili Wei, Chengcheng Xiang, Yudong Wu, Mingyao Shen, Yuanyuan Zhou, and Xinxin Jin. 2021. Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model. In 30th USENIX Security Symposium (USENIX Security).
[81] Ravi Inder Singh, Manasa Sumeeth, and James Miller. 2011. Evaluating the readability of privacy policies in mobile environments. International Journal of Mobile Human Computer Interaction (IJMHCI) 3, 1 (2011), 55–78.
[82] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D Breaux, and Jianwei Niu. 2016. Toward a framework for detecting privacy policy violations in android application code. In 38th International Conference on Software Engineering (ICSE).
[83] Sourcepoint. 2024. PRIVACY TECHNOLOGY FOR DIGITAL MARKETING. https://sourcepoint.com/.
[84] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negrón-Arroyo, Christopher Thompson, Serge Egelman, and David Wagner. 2014. The effect of developer-specified explanations for permission requests on smartphone user behavior. In SIGCHI Conference on Human Factors in Computing Systems (CHI).
[85] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. 2019. (Un) informed consent: Studying GDPR consent notices in the field. In 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS).
[86] Jice Wang, Yue Xiao, Xueqiang Wang, Yuhong Nan, Luyi Xing, Xiaojing Liao, JinWei Dong, Nicolas Serrano, Haoran Lu, XiaoFeng Wang, et al. 2021. Understanding malicious cross-library data harvesting on android. In 30th USENIX Security Symposium (USENIX Security).
[87] Wikipedia. 2024. Softmax function - Wikipedia. https://en.wikipedia.org/wiki/Softmax_function.
[88] Maximiliane Windl, Niels Henze, Albrecht Schmidt, and Sebastian S Feger. 2022. Automating contextual privacy policies: Design and evaluation of a production tool for digital consumer privacy awareness. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. 1–18.
[89] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. 2016. Can we trust the privacy policies of android apps?. In 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 538–549.
[90] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel R Reidenberg, N Cameron Russell, and Norman Sadeh. 2019. Maps: Scaling privacy compliance analysis to a million apps. Proc. Priv. Enhancing Tech. 2019 (2019), 66.
[91] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven Bellovin, and Joel Reidenberg. 2016. Automated analysis of privacy requirements for mobile apps. In 2016 AAAI Fall Symposium Series.

A COMPLIANCE SUGGESTIONS

To help mobile app developers ensure GDPR compliance, we extracted the following five suggestions from the findings and good practices identified in our research.

- **Timely Provide RPN.** To comply with GDPR and avoid user confusion about the correspondence between data collection behaviors and RPN practices, it is suggested to timely provide RPNs right before relevant data collection occurs.
- **Avoid Redundant RPN Content.** General notice elements that remain the same for all users (e.g., the identity of data controller (IC) including the contact details of the controller and the controller's representative) are suggested to be provided once. The main objective here is to ensure that each

essential notice element is notified at least once and unnecessary redundancy is avoided to make RPN concise to comply with GDPR.

- **Make RPN Clear and Organized.** To assure that users of different ages and backgrounds can easily understand RPNs, RPNs should be organized and presented by using clear and plain language to well inform users.
- **Adopt CMP SDKs.** Since CMP SDKs are shown to improve RPN practices, first-party developers are encouraged to integrate CMP SDKs to help implement RPNs and ease the implementation burden.
- **Cooperate to Improve Transparency.** Third-party SDK developers are highly recommended to provide easily accessible and function-level notice information for their owned SDKs' privacy-sensitive behaviors, which can effectively ease the burden of first-party developers.
- **Refer to Automated Generation of RPNs.** We suggest employing automated techniques [77, 88] that analyze privacy policies, consider contextual information, and generate candidate RPNs. Developers can then check these candidates and implement the final RPNs in their apps.
- **Enhance RPN Requirements.** Considering the significant role that app stores play in the RPN ecosystem, their vetting policies in EU countries should align with GDPR to avoid confusion for app developers.

B APPLICATION SCENARIOS OF RENO

Generally, both regulatory authorities and app stores can effectively rely on the high performance of RENO to vet whether mobile apps comply with GDPR requirements. Furthermore, app developers deeply understand their own app's privacy collection practices and RPN behaviors. Hence, they can easily verify GDPR compliance by referencing the core requirements outlined earlier in this paper.

Moreover, while GDPR applies to all areas that involve user data processing in the EU, RENO is specialized in checking compliance issues of RPNs in mobile apps and thus extra efforts may be needed to port it to other platforms. For instance, to check the status of "Right-to-be-Informed" in web platform, the testing environment of RENO needs to be updated with suitable techniques, e.g., dynamic web testing, to trigger and monitor the runtime behaviors of web apps. Despite these limitations, when deployed in areas other than mobile platforms, the core components of RENO that detect and analyze RPNs can still be referred to perform compliance checks upon users' "Right-to-be-Informed".

C IDENTIFICATION SCHEME OF PRIVACY COLLECTION

For the first two types of user data managed by the Android system, access to them can be precisely monitored by the app-level instrumentation. More importantly, a three-fold privacy recognition scheme is employed to identify user data that leaves the user's device. First, keyword matching is applied to identify user data that is sent out. Second, regular expression (regex for short) is employed to match values of user data that have fixed formats, e.g., the values of email addresses should be like `user@mail.server.name` and thus can be matched with a regex. Third, a predefined value matching is conducted on user data that may be encoded or encrypted before being sent out. In particular, the predefined value list includes the Google Advertiser ID, Android ID, device ID, IMEI, email addresses, phone number, and their encoded or hashed values (with MD5, SHA1, SHA256, 46ESAB, BASE64). Thus, the common obfuscation or encodings applied to the collected user data can be well handled.