ClearCommPrivacy: Communicating App Privacy Behavior in Android

Elijah Neundorfer
Columbus State University
Columbus, Georgia, USA
neundorfer_elijah@columbusstate.edu

Alfredo J. Perez
Columbus State University
Columbus, Georgia, USA
perez_alfredo@columbusstate.edu

ABSTRACT

Informing users of Android apps' privacy behavior is crucial to maintain transparency. In the past, approaches have been developed to communicate app privacy behavior based on frameworks that require extensive APIs, new permission models, entirely new Operating Systems (OS), and/or third-party plugins/tools to assist developers. In this work, we present ClearCommPrivacy, a User Interface (UI) template for Android apps to convey privacy/permission information in an app by a developer familiar with an app's privacy behavior using a standardized code template and two XML files. We present the design of ClearCommPrivacy and some basic evaluation results.

CCS CONCEPTS

• **Human-centered computing** → **Mobile devices**; • **Security and privacy** → **Usability in security and privacy**; • **Software and its engineering** → Integration frameworks.

KEYWORDS

Usable Privacy, Privacy Policy, Android, Permissions, Terms of Service

ACM Reference Format:
Elijah Neundorfer and Alfredo J. Perez. 2022. ClearCommPrivacy: Communicating App Privacy Behavior in Android. In *2022 ACM Southeast Conference (ACMSE 2022), April 18–20, 2022, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3476883.3520231

1 INTRODUCTION

The Android permission model protects users by preventing apps to access certain data and device functions such as device's location, cameras and microphone, and access to shared storage without first obtaining a user's or device's permission [5, 8, 14, 19, 21]. A number of Android permissions are considered dangerous and can violate user's privacy (or device operation) [8, 14]. In addition to these technical issues with the Android platform (which may be similar in other platforms), a second issue with permissions is the legal aspect of privacy. Privacy policies and terms of service are known to be inscrutable and very long in many cases, hiding apps/company's privacy practices in layers of legal language [6, 13, 18]. It would take take 244 hours per year to read all the policies on each unique website that an average user visits [13].

Privacy policies and notices present users information about how their personal data is being used, but they are not always clearly presented [16, 17], thus generating significant privacy problems: (1) malicious developers can easily take advantage of users through their confusion and willingness to trust their apps; (2) non-malicious developers can accidentally acquire more information that a user may not willing to share; and (3) non-malicious developers can accidentally collect more data than they intend to as they are unfamiliar with best practices in privacy [12]. Techniques for accessible privacy policies and communication have been proposed in the past [12, 13, 20]. However, these techniques may require new programming frameworks and/or entirely new operating systems. Some developers posit using these new privacy models without considering how they may be legally binding, and some of these techniques require extensive development time to function properly. Typically the communication of privacy behavior is not a priority [12], thus resulting in an impaired communication about the privacy behavior of an app (or a connected device [10]).

Third-party groups have dedicated resources to evaluate apps' privacy behaviour (after they have been developed) to determine users' risk when they use these apps. For example, the CHIMPS Lab at Carnegie Mellon University has built a tool, called PrivacyGrade, which analyzes the permissions each app requests, the data it collects, and why it collects this data, and then gives each app a letter grade corresponding to its risk [11]. However, a major problem with this approach is that third-party groups can only speculate on some pieces of data. For example, the team behind PrivacyGrade speculates that Instagram requests users' location to attach their location on public social media posts [11]. This is most likely true, but the team behind PrivacyGrade notes that they do not know for sure.

While there have been some changes in the Google Play market (the largest of the Android app markets) for developers to disclose privacy policies and app data usage [7] (in part to comply with General Data Protection Regulations (GDPR) [16]), such policies do not modify the current Android app permission model based on XML permissions and pop-up user notifications/consent (for permissions considered dangerous). Thus, while the metadata about privacy behavior is available to users before installation time, it may not be seen afterwards. Finally, third-party privacy evaluation tools require the app to have enough demand that to warrant an external evaluation. Privacy evaluation takes a significant resources and is not always accurate, and therefore it is unrealistic to rely on these methods for privacy decisions.

ACMSE 2022, April 18–20, 2022, Virtual Event, USA

ACM ISBN 978-1-4503-8697-5/22/04…$15.00
https://doi.org/10.1145/3476883.3520231

In this work we present ClearCommPrivacy, a User Interface (UI) template to simplify the work placed on developers to present and disclose an app's privacy information in the Android platform. ClearCommPrivacy presents privacy information to users in a friendly and readable manner through the automatic organization and presentation of metadata provided by a developer.

2 TOOL DESIGN

We designed ClearCommPrivacy to inform users about four groups of metadata related to an app's privacy information: (1) what permissions are being requested by the app and what data is being accessed using those permissions; (2) why the developer wants this data; (3) whether or not the data can be accessed while the app is in the background; and (4) whether the data remains solely on the app or is shared with either the company owning the app or third party companies. Two components are used to present this data to the user: (1) a graphical user interface (called Privacy Behavior UI); (2) a software template implemented (called Developer-Facing Template).

2.1 Privacy Behaviour UI Design

ClearCommPrivacy's UI presents an app's privacy behavior by splitting it in five parts: (1) the relevant permissions; (2) a description of the data accessed; (3) the reason for accessing the data; (4) the state of the app at the time of the access (i.e., foreground or background); and finally (5) whether or not the data is shared beyond the app. We based the UI on the model proposed by the Brandeis team [9]. To display this data, the privacy behavior UI includes two layers. The first layer displays a list of data the app accesses. Each of these data entries is grouped according to the relevant permission group. We present an example of this layer in Figure 1. This interface can also display a list of data accesses for permissions that were either requested in the app but not described by the developer, or data accesses for permissions that were described by the developer but not requested by the app. The second layer of the privacy behavior UI includes a detailed description for each data access. For example, when a user taps "Device Location" as listed in Figure 1, the user will be presented with Figure 2. In this example, the interface gives the user a description as to why the app is accessing location data, whether this data leaves the application, and if it is accessed while the app is not in use.

The UI can provide a link to any relevant privacy policies or terms of service, and it warns the user if a permission requested did not provide any relevant metadata. The UI is meant to be standardized so users can see a consistent UI design and structure across multiple apps. Because ClearCommPrivacy makes use of the CardView and RecyclerView UI classes, the minimun SDK needed is Android SDK 28 (Android 9.0). As of December 2021, more than 70% of Android smart phone devices globally run Android 9.0 or above [22].

2.2 Developer-facing Template Design

For the developer, our first goal was to create a template that makes use of standard tools for Android development without installing additional plugins (i.e., using only Android Studio and Java/Kotlin). Our secondary goal for ClearCommPrivacy was that a developer familiar with an app's privacy behavior could easily adapt

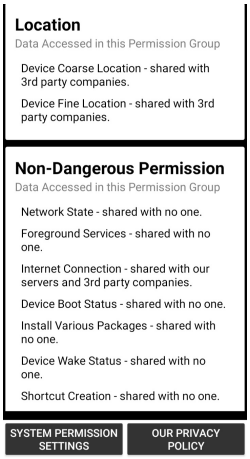

Figure 1: ClearCommPrivacy's Main UI Graphical User Interface for DuckDuckGo

Figure 2: App's Access Explanation for Location Data Using ClearCommPrivacy's Privacy Behavior UI in DuckDuckGo

ClearCommPrivacy for their needs. The UI presented in the past section requires the developer to import a set of classes (Figure 3), adding them to the correct package, updating their app's manifest and build.gradle files. The developer then uses the files to provide metadata for each permission that the app requests as explained in the previous section. When the app is compiled, ClearCommPrivacy will verify whether the permissions listed in the app's package manager match the permissions the developer has stated being used. The tool highlights permissions that were not detailed by the developer in the generated UI. Finally, the tool will compile all this information to be presented in the end-user privacy behavior UI previously described.

We created a Java class called *DataAccessDescriptor* to facilitate customization. Instances of this class hold the metadata that explains the privacy behavior of the app for an specific permission. For example, if an app accesses the device location using the ACCESS_FINE_LOCATION permission in two different pieces of code,

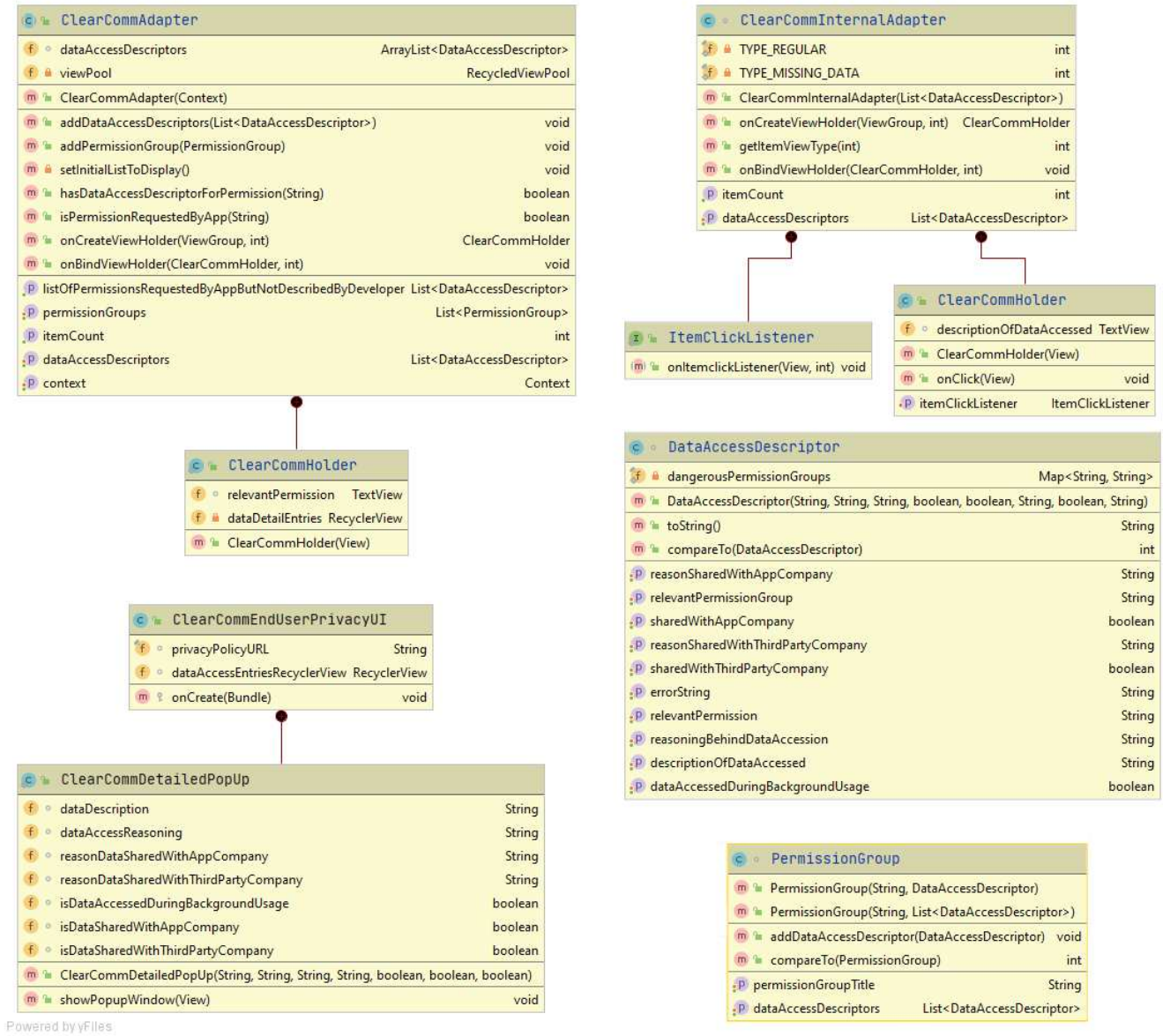

Figure 3: ClearCommPrivacy UML Class Diagram

the developer will create only one instance of *DataAccessDescriptor*. However, if an app accesses both the device location and the device speed, the developer will create two instances. If the developer wants the user to read the app's privacy policy, the developer supplies a string with a URL referencing the privacy policy. When this string is a valid URL, the UI shows a button in the app's UI to navigate the user to the privacy policy. Figure 4 presents a illustration of the utilization of the *DataAccessDescriptor* class and how the data in the instances populate the UI.

When an app using ClearCommPrivacy starts the privacy behaviour activity, the *DataAccessDescriptor* instances are sorted by Android permission groups that the user is familiar with (e.g., Body Sensors, Calendar, Call logs, Camera, Contacts, Location, Microphone, Phone, Physical Activity, SMS, and Storage). Any permission not sorted in these groups is presented to the user as part of a "non-dangerous permission" group. For permissions listed within the instances of *DataAccessDescriptor* that do not match permissions requested to the OS by the app, the UI presents a warning informing the user that the permission described was never requested. For any permissions requested to the Android OS that are not listed within the instances of *DataAccessDescriptor*, the UI presents a warning to the user that their data is being collected but the developer has not provided any metadata information (Figure 5).

Figure 4: Visibility of DataAccessDescriptor Fields in the Privacy Behavior UI

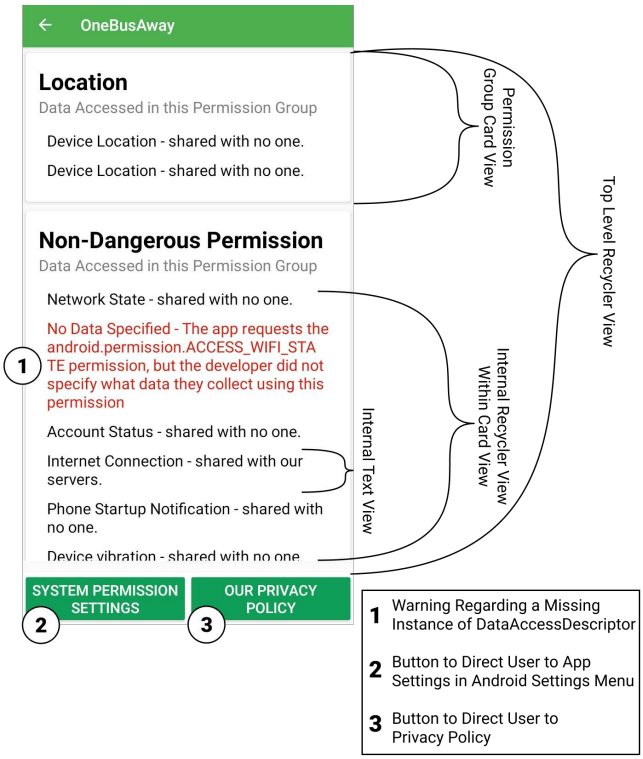

Figure 5: An Annotated View of the Privacy Behavior UI for the OneBusAway App

3 EVALUATION

We evaluated ClearCommPrivacy by measuring how quickly can it be adapted in an app. To accomplish this, we took five open source apps available in GitHub which make use of a variety of Android permissions and different UI designs. In this section we describe how we successfully implemented the privacy behavior UI for four of the five apps, and the time it took to implement ClearCommPrivacy in each app.

We cloned each project repository from GitHub. Before making any changes to these apps, we executed the apps in a Samsung Galaxy S9 device running Android 10 to ensure that the app worked as expected. To implement ClearCommPrivacy, we used two different computers: an ASUS computer running Microsoft Windows 10 Home Edition for three apps, and a MacBook Pro running macOS 11.0.1. In both computers Android Studio was used.

For the installation and implementation, we followed the instructions provided in a GitHub repository for ClearCommPrivacy. We took two steps, first installing ClearCommPrivacy into the app, and then completing the implementation for the privacy behavior UI. We tracked the total time measured from the moment the tester began downloading the ClearCommPrivacy compressed zip file from the GitHub repository, until the time the app compiled and ran properly with all instances of DataAccessDescriptor and the privacy policy link properly implemented. We also collected data about the amount of time to complete each part of the implementation. We show in Table 1 a breakdown for these times for each evaluated app. In this table the column *Install Time* corresponds to the time it took to download the app from GitHub and get it running/installed in the phone, the column *Impl. Time Part #1* corresponds to the amount of time it took the tester to add ClearCommPrivacy to the app and the column *Impl. Time Part #2* corresponds to the time it took to implement the *DataAccessDescriptor* for privacy behavior.

Of the five apps, we successfully installed and implemented ClearCommPrivacy in four. We failed to adapt the 1Sheeld Android App [1] because of backward compatibility issues in Android Studio with regards to the gradle plugin and external libraries. For the rest of the other four apps, we had no issues on adapting the apps in Android SDK 28 and Androidx. On average, it took us around 46 minutes to implement ClearCommPrivacy in each of these four apps. In each of these implementations, it is assumed whoever uses ClearCommPrivacy already knows or have access to a document that describes the privacy behavior of the app in advance. Thus for our tests, the tester inferred information based on the permissions and any documentation from the app developer and/or used placeholder information.

Table 1: Results of Implementing ClearCommPrivacy Tool in Five Open-source Android Apps

Mobile App	Install Time	Impl. Time Part #1	Impl. Time Part #2	Total Time
One Bus Away [15]	11m, 35s	27m, 5s	20m, 15s	58m, 55s
CallMeter [2]	25m, 29s	7m, 51s	15m, 53s	49m, 13s
PSLab [4]	19m, 30s	3m, 8s	13m, 47s	36m, 25s
DuckDuckGo [3]	9m, 20s	6m, 41s	26m, 26s	42m, 27S
1Sheeld Android App [1]	N/A	N/A	N/A	N/A
Average	16m, 28.5s	11m 11.25s	19m 5.25s	46m, 45s

4 CONCLUSION AND FUTURE WORK

In this work we have presented the design and implementation of ClearCommPrivacy, a UI template for Android that allows developers to easily communicate the privacy behavior of their apps to end-users. As future work we plan to evaluate the usability of the UI template for users and developers, continue modifying ClearCommPrivacy to support different languages (internationalization), as well as to develop an application that a privacy engineer could use to generate the privacy behavior descriptors in advance to simplify the process in integrating the privacy behaviors in an app using ClearCommPrivacy.

ACKNOWLEDGMENTS

This research was partially supported by the National Science Foundation under grant award No. 1950416.

REFERENCES

[1] 1sheeld. 2021. *1Sheeld Android App.* https://github.com/Integreight/1Sheeld-Android-App
[2] Felix Bechstein. 2021. *Call Meter 3G.* https://github.com/felixb/callmeter
[3] DuckDuckGo. 2021. *DuckDuckGo Android.* https://github.com/duckduckgo/Android
[4] FOSSASIA. 2021. *PSLab Android App.* https://github.com/fossasia/pslab-android
[5] Yanick Fratantonio, Chenxiong Qian, Simon P. Chung, and Wenke Lee. 2017. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Jose, USA, 1041–1057. https://doi.org/10.1109/SP.2017.39
[6] Google Inc. 2020. *Terms of Service; Didn't Read.* https://tosdr.org/
[7] Google Inc. 2021. *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
[8] Google Inc. 2022. *Android Permissions Overview.* https://developer.android.com/guide/topics/permissions/overview
[9] Aniruddh Iyer, W Chung, Qian Wang, and Ally Liu. 2019. *Brandeis Project.* https://www.aniruddh-iyer.com/brandeis
[10] Yeşem Kurt Peker, Gabriel Bello, and Alfredo J. Perez. 2022. On the Security of Bluetooth Low Energy in Two Consumer Wearable Heart Rate Monitors/Sensing Devices. *Sensors* 22, 3 (2022), 988.
[11] CMU CHIMPS Lab. 2014. *PrivacyGrade.* http://privacygrade.org/
[12] Tianshi Li, Yuvraj Agarwal, and Jason I. Hong. 2018. Coconut: An IDE Plugin for Developing Privacy-friendly Apps. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 178 (Dec 2018), 35 pages.
[13] Aleecia M. McDonald and Lorrie Faith Cranor. 2008. The Cost of Reading Privacy Policies. *Isjlp* 4 (2008), 543. https://kb.osu.edu/handle/1811/72839
[14] Ramon P. Medina, Elijah B. Neundorfer, Radhouane Chouchane, and Alfredo Perez. 2018. PRAST: Using Logic Bombs to Exploit the Android Permission Model and a Module Based Solution. In *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, Nantucket, USA, 1–8.
[15] OneBusAway. 2021. *OneBusAway for Android.* OneBusAway. https://github.com/OneBusAway/onebusaway-android
[16] Alfredo J. Perez and Sherali Zeadally. 2021. Recent Advances in Wearable Sensing Technologies. *Sensors* 21, 20 (2021), 6828.
[17] Alfredo J. Perez, Sherali Zeadally, and Jonathan Cochran. 2018. A Review and an Empirical Analysis of Privacy Policy and Notices for Consumer Internet of Things. *Security and Privacy* 1, 3 (2018), e15.
[18] Alfredo J. Perez, Sherali Zeadally, and Nafaa Jabeur. 2018. Security and Privacy in Ubiquitous Sensor Networks. *Journal of Information Processing Systems* 14, 2 (2018), 286–308.
[19] Anthony Peruma, Jeffrey Palmerino, and Daniel E. Krutz. 2018. Investigating User Perception and Comprehension of Android Permission Models. In *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*. Gothenburg, Sweden, 56–66.
[20] Irene Pollach. 2007. What's Wrong with Online Privacy Policies? *Commun. ACM* 50 (Sep 2007), 103–108. https://www.researchgate.net/publication/220421895_What's_wrong_with_online_privacy_policies
[21] Wook Shin, Sanghoon Kwak, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka. 2010. A Small But Non-negligible Flaw in the Android Permission Scheme. In *2010 IEEE International Symposium on Policies for Distributed Systems and Networks*. Washington D.C., USA, 107–110.
[22] Statistica Ltd. 2021. *Mobile Android Operating System Market Share by Version Worldwide from January 2018 to December 2021.* https://www.statista.com/statistics/921152/mobile-android-version-share-worldwide/