Privacy Analysis of COVID-19 Contact Tracing Apps in the EU

Samuel Wairimu[1(✉)] and Nurul Momen[1,2]

[1] Karlstad University, Universitetsgatan 2, 651 88 Karlstad, Sweden
{samuel.wairimu,nurul.momen}@kau.se
[2] Blekinge Institute of Technology, Karlskrona, Sweden

Abstract. This paper presents results from a privacy analysis of COVID-19 contact tracing apps developed within the EU. Though these apps have been termed advantageous, concerns regarding privacy have become an issue that has led to their slow adoption. In this empirical study, we perform both static and dynamic analysis to judge apps' privacy-preserving behavior together with the analysis of the privacy and data protection goals to deduce their transparency and intervenability. From the results, we discover that while the apps aim to be privacy-preserving, not all adhere to this as we observe one tracks users' location, while the other violates the principle of least privilege, data minimisation and transparency, which puts the users' at risk by invading their privacy.

Keywords: Privacy · COVID-19 · Contact tracing apps

1 Introduction

The global spread of COVID-19 resulted in governments taking extreme measures to prevent further spread of the pandemic within their borders. In the EU, the imposition of these measures, which include partial to total lock-down of cities or the entire country, has seen the restriction of fundamental human rights and freedoms (e.g., liberty), and a significant decline of the economy [23]. For example, EU member states such as Spain, Greece and Portugal whose economies mostly depend on tourism -contributing to over 15% of there respective GDPs- will be highly affected by measures introduced as a way of reducing the spread of COVID-19 [10]. Hence, to ease these restrictions, support manual contact tracing in the context of public health, and allow the return to a new normal, several EU member states have followed suit in the development and rolling-out of contact tracing apps (e.g., France - `StopCovid France` and Spain - `Radar COVID`), while others are on the process of developing one (e.g., Belgium[1] and Luxembourg[2]).

[1] https://www.brusselstimes.com/all-news/belgium-all-news/health/120349/belgian-contact-tracing-app-will-be-ready-in-september/, Accessed 07.07.2020.
[2] https://today.rtl.lu/news/luxembourg/a/1514009.html, Accessed 07.07.2020.

M. Asplund and S. Nadjm-Tehrani (Eds.): NordSec 2020, LNCS 12556, pp. 213–228, 2021.
https://doi.org/10.1007/978-3-030-70852-8_13

Nevertheless, with the said advantages that come with the use of contact tracing apps, there have been concerns regarding privacy which slow down their adoption. In the common EU toolbox for member states, the adoption of these apps by users depends on privacy preservation and trustworthiness [8]. Additionally, in the guidelines adopted by the European Data Protection Board (EDPB), contact tracing apps should be compliant with the GDPR and privacy legislation [9]. It is in this regard that we investigate the privacy of contact tracing apps deployed in the EU member states with the aim of determining if they are privacy friendly. As such, we analyse the `AndroidManifest.xml` files of these apps with a focus on permissions declared in relation to their respective frameworks outlined in Table 2. We measure permission usage with and without user interaction to gain an insight into the apps' actual permission access behaviour. Finally, we look into the privacy and data protection goals to assess the privacy aspect of these apps in terms of their transparency and intervenability.

Research Questions: With a number of studies discussing the privacy aspects of contact tracing apps across EU, for instance [18,20]; our interest is driven by critiquing their behaviour and data protection expectations empirically. To accomplish these, we set out to answer the following questions:

1. Do the apps violate the Principle of Least Privilege (PoLP)? According to Saltzer and Schroeder [22], a program needs to function with the least set of privileges in order to avoid any form of malicious interaction. Hence, we assess these apps to identify whether they operate with the least set of privileges (permissions) by measuring their actual permission access in relation to the app's core functionality.
2. How do these apps behave during runtime, i.e., with and without user interaction? While static analysis is used in determining declared permissions and permission levels, we monitor the apps during runtime by measuring permissions access patterns, which provides an insight into how they actually behave.
3. Can a person be identified based on the permissions accessed by these apps during runtime? According to the principles of data minimisation (Art. 5(1)(a)) and purpose limitation (Art. 5(1)(b)) of the GDPR, data collections should be kept at a minimum and for a specific purpose respectively. Violating these principles could lead to the amassing of personal data that potentially allows for linkability and the identification of a person [25]. As such, we identify apps that collect more data than required for their core functionality and assess whether a user could be identified through such data.
4. Is the privacy and data protection goals of transparency and intervenability respected? With the acknowledgement that there are three privacy and data protection goals [16], we identify and assess the goals of transparency and intervenability as they can be inspected from the end-user side. Hence, this bit provides the answer to whether these apps are open in terms of data processing, and if the rights of the data subject are implemented.

To answer the questions, we adapt different assessment metrics that provide an insight and a comparison into the privacy of these apps deployed across the EU member states.

Outline: The rest of the paper is organised as follows: Sect. 2 discusses the background of contact tracing apps. Section 3 discusses the methods, which includes the inclusion criteria for app selection and different analytical approaches for privacy analysis. Section 4 discusses comprehensive results of the analysis while Sect. 5 provides the discussion, limitations, and conclusion of the study.

2 Background

Contact tracing is a key procedure when it comes to preventing the spread of a highly contagious infection. This process requires quick identification of individuals who have come into close contact with an already identified case of the said infection. Conventionally, the process of contact tracing is based on manual tracking where individuals suspected of being in close contact to a confirmed case are identified, and a contact list is constructed for immediate follow-up. However, in certain cases, this process is not only labor intensive and marred with privacy concerns due to direct identification of infected individuals, but is also reliant on human memory, which more than often leads to inaccuracies [1,21]. Manual contact tracing can easily take place where a deadly contagion is contained rather than widely spread to a point of overwhelming the authorities [1,4]. However, with the current global spread of COVID-19, manual contact tracing has become more arduous; hence the need for apps that can construct a digital record of ephemeral proximity identifiers and instantly notify users if they have come into close contact with an individual who has previously tested positive for COVID-19 [11]. While contact tracing apps are not meant to replace manual tracing [9], their uses have been termed as an advantage as they would allow a smooth exit strategy, including the return to normalcy of the fundamental human rights and freedoms that had been restricted to reduce the spread of the disease.

These apps work by building a digital record of identifiers derived from proximity data, which is obtained from either Bluetooth Low Energy (BLE) or location data. The latter, which has already been put in use by certain countries across Asia (e.g., Taiwan and South Korea), uses a comprehensive time-stamped list of GPS locations obtained from users' mobile devices. While this approach seems to work in the mentioned countries, such solutions cannot be embraced by European Citizens as they are regarded invasive in terms of privacy [1]; indeed, the use of location based services could be used to determine both the identity of the person and their surroundings [12]. As such, several contact tracing apps that utilise BLE have been developed as their use has been found to be more effective, or rather suitable, in detecting contacts between people rather than the use location data [3,6]. The BLE technology depends on the exchange of identifiers between nearby devices via a Bluetooth connection. In the context of COVID-19, the proximity and period of exposure between people influences the

probability of an infection [1]. It is in the same context that the use of proximity tracing apps has become convenient as the exchange of identifiers between two or more close devices could be used to notify a user if they have been exposed to the infection, without sacrificing their privacy. As a result, in their guidelines, EDPB highlight and support the idea of using apps that do not require access to location data as proximity data is considered sufficient in tracing COVID-19 cases [9].

Hence, several privacy-preserving contact tracing apps that rely on proximity data and are compliant with the GDPR have been proposed and considered within EU member states [8]. These apps leverage a number of frameworks, for example, ROBERT (ROBust and privacy-presERving proximity Tracing), among others [1]. Recently, Google and Apple released the ExposureNotification API framework; a system which facilitates in alerting users of the possibility of having been potentially exposed to COVID-19[3]. These frameworks strive to be privacy-preserving and compliant with the data protection regulation. As a result, these frameworks are currently being leveraged when it comes to developing contact tracing apps across EU member states (see Table 2). Nevertheless, developing contact tracing apps from different frameworks results in apps seeking different goals and having contrasting designs, which could possibly lead to a number of privacy violations. For example, [4] discusses an attacker model where an adversary can violate a user's privacy by deanonymizing their IDs with the intention of tracing new cases. Therefore, we analyse and compare the privacy of contact tracing apps deployed in several EU member states in relation to their respective frameworks, with the aim of investigating whether they are privacy-friendly or privacy invasive.

3 Methods

In this section, we define the inclusion criteria, which is relevant in determining which apps are to be included in the study, followed by the assessment methodology, which we follow to answer the aforementioned questions. We limit our study to the Android platform due to its large user base and open source nature.

3.1 Inclusion Criteria

Several EU member states have already developed and rolled-out contact tracing apps; nonetheless, it is worth knowing that not every app was eligible for inclusion in our study. During the initial app installation phase, it was noted that not all apps could be installed and run on the test device based on a number of reasons, which include but are not limited to: the requirements of citizen's personal data and the unavailability of apps in the official app store, in this case Google Play Store. Following this, we defined inclusion criteria that guided us

[3] https://developer.apple.com/documentation/exposurenotification, Accessed 09.07. 2020.

Table 1. Inclusion criteria for app selection

Criteria	Description
Criterion 1	The app should not ask for registration details (e.g., Phone Number)
Criterion 2	The app should be available for installation in the country of study
Criterion 3	The app should be available in official stores, i.e., Google Play Store
Criterion 4	App's functionality - the app should be used for contact tracing purposes

in determining which apps could be included in our study. Table 1 shows the criteria followed in selecting and installing the apps that were deemed eligible for our study. A majority of the released apps were asking for citizen's registration details such as phone numbers with a country code so as to access the app's core functionality (e.g., `eRouška - Czechia Republic`). As a result, we focused on apps that did not require registration of personal details for its use. Moreover, a number of apps (e.g., `ProteGO Safe - Poland`) were unavailable within the country where the study was being conducted (i.e., outside of their origin country); hence, such apps were automatically excluded from our study. In addition to this, we targeted official contact tracing apps that had been released by public authorities and published in official app stores. One of the recommendations outlined by EDPB is that public officials should provide links to their respective official contact tracing apps so as to prevent users from installing third-party apps, which might pose significant risks to their privacy [9]. As such, we followed the links provided to download apps as per the criteria provided. Finally, with the apps having different functionalities, for example self-diagnosis as in the case of Greece `DOCANDU Covid Checker`, we focused only on apps whose core functionality is contact tracing. As a result, we were able to install and run a total of 7 apps, each from a different EU member state as indicated in Table 2.

3.2 Assessment Methodology

The study design followed in order to provide an in depth privacy analysis of the apps has three different assessment metrics: Static Analysis, Dynamic Analysis (with and without user interaction) and, Privacy and Data Protection Goals Analysis.

Static Analysis: During the development of an Android app, it is mandatory for a developer to include an `AndroidManifest.xml` file within the app's APK. It is in this file that the developer declares, inter alia, the app's package name, build-version code, its principle components, etc., that the app needs for a particular purpose. Of importance is the declaration of the permissions that an app needs in order to access sensitive system resources (e.g., GPS) and user's personal information such as location. By declaring these permissions, the Android Operating System ensures that users' privacy are safeguarded by permitting secure access to sensitive resources [15]. Further, a developer is recommended to declare

Table 2. List of apps collected in conjunction with their respective frameworks

Apps #	Framework	Country
Stopp Corona	Apple/Google - ExposureNotification API	Austria
CovTracer	Safe Paths (MIT-led project)	Cyprus
Smitte—stop	Apple/Google - ExposureNotification API	Denmark
StopCovid France	ROBERT	France
Immuni	Apple/Google - ExposureNotification API	Italy
Apturi Covid	Apple/Google - ExposureNotification API	Latvia
Corona-Warn-App	Apple/Google - ExposureNotification API	Germany

the least privileged set of permissions required for the app's functionality [24]. As such, we extract the manifests from the apps APK files and analyse them with the intention of gaining an insight into the apps' protection levels through the evaluation of the permissions declared. Additionally, we investigate whether the permissions declared correspond to the apps' functionalities in relation to their underlying framework specifications.

Dynamic Analysis: Based on permissions declared, an app is able to request access to required resources during runtime. These permissions, when granted by a user, access resources in a manner and frequency that a user is unaware of. As such, several studies for instance [17,19], shed some light on this by vetting the runtime behaviour of android apps by analysing how often they access sensitive resources and their actual permission access pattern. Hence, we adopt and apply a similar approach on the apps in order to inspect the frequency in which they access resources and if they portray uncalled for behaviour during resource utilisation by analysing their actual permission access patterns. Further, we uncover whether the apps adhere to PoLP, *"that is, each app, by default, has access only to components that it requires to do its work and no more"*[4], by comparing the actual accessed permissions to the apps core functionality. We base this analysis on data gathered from two phases, that is, with and without user interaction; both collected on separate occasions for a period of six days. To accomplish data collection, we use A-Ware, a prototype tool introduced by [19] that runs as a service and uses AppOpsCommand[5] to extract and log in the accessed resources. The tool logs in resource events that the apps previously accessed and records them in a predefined format that is later saved and accessed as a JSON file (see sample log below).

[4] https://developer.android.com/guide/components/fundamentals.html, Accessed 28.07.2020.

[5] https://android.googlesource.com/platform/frameworks/base/+/android-6.0.1_r25/cmds/appops/src/com/android/commands/appops/AppOpsCommand.java, Accessed 28.07.2020.

```
{"Package":"com.netcompany.smittestop_exposure_notification
    ","Permission":"READ_EXTERNAL_STORAGE","Timestamp":"Wed
    Jul 08 11:05:41 EDT 2020"}
```


In addition to the aforementioned analysis, we investigate whether the actual permissions accessed and recorded during runtime using A-Ware could be used to identify a user through linkability, that is, *"if too much linkable information is combined"* [25]. For example, if the apps access location data through the `ACCESS_COARSE_LOCATION` or `ACCESS_FINE_LOCATION` permissions, one could be able to directly infer the whereabouts, area or address of an app user. To observe this, we adopt a model introduced by [13] that visualises the identity of person by mapping it to permission accessed by a particular app.

Privacy and Data Protection Goals Analysis: In the protection goals for privacy engineering, Hansen et al. [16] describe unlinkability, transparency and intervenability as the three privacy and data protection goals that complement the CIA (confidentiality, integrity and availability) triad. While all these are important aspects, we focus on the transparency and the intervenability goals as they can inspected from the end-user. By analysing the goal of transparency, we assess how open these apps are in terms of data processing. On the other hand, assessing the intervenability goal aims at analysing if the data subject rights have been implemented from the end user perspective [16]. To achieve this, we investigate these goals by relating them to the GDPR and in relation to the apps privacy policies which we extracted and archived.

4 Results

This section presents the main findings of our analysis.

4.1 Manifest Analysis

Essentially, for an app to perform as required, it normally needs access to certain resources from either the user or the system. These resources are conventionally accessed through permissions, and depending on which resources the app requires, the permissions can either be granted automatically or explicitly through user's approval. The permissions requested, which act as protection mechanisms for user privacy, are of three levels[6]: *Normal*, *Signature* and *Dangerous*. *Normal* permissions are granted automatically (i.e., during installation of the app) as they access resources that pose little threat to the users privacy. Like normal permissions, *Signature* permissions are automatically granted at install time, however, they only access permissions signed by the same certificate. Finally, *Dangerous* permissions are exclusively granted by the user at run-time as they access sensitive resources that pose a high risk to the user's privacy.

[6] https://developer.android.com/guide/topics/permissions/overview#normal-dangerous, Accessed 28.07.2020.

Permissions Declared per App: Table 3 shows the permissions declared in the manifest files of the apps indicated in Table 2. Overall, the apps declare a total of 64 permissions, which are grouped into two permission levels. Of these permissions, 82.81% are Normal and 17.19% are Dangerous. Taking each app in isolation, it can be noted that `CovTracer` requests a large number of permissions (20 in total - with 35% covering dangerous permissions), followed by `StopCovid France`, which requests a total of 11 permissions - with 27.3% covering dangerous permissions. On the other hand, `Corona-Warn-App` requests only one dangerous permission by declaring the use of `CAMERA`.

Table 3. Permissions declared within each app's `AndroidManifest.xml` file. Y is used in this context to indicate the permissions requested per app

Permissions	Stopp corona	Cov tracer	Smitte stop	Stop covid france	Immuni	Apturi covid	Corona-warn-app
BLUETOOTH	Y	Y	Y	Y	Y	Y	Y
BLUETOOTH_ADMIN		Y		Y			
INTERNET	Y	Y	Y	Y	Y	Y	Y
RECEIVE_BOOT_ COMPLETED	Y	Y	Y	Y	Y	Y	Y
ACCESS_NETWORK_ STATE	Y	Y	Y	Y	Y	Y	Y
WAKE_LOCK	Y	Y	Y	Y	Y	Y	Y
FOREGROUND_ SERVICE	Y	Y	Y	Y	Y	Y	Y
ACCESS_LOCATION_ EXTRA_COMMANDS		Y					
READ_SYNC_SETTINGS		Y					
WRITE_SYNC_SETTINGS		Y					
ACCESS_WIFI_STATE		Y					
AUTHENTICATE_ACCOUNTS		Y					
REQUEST_IGNORE_ BATTERY_OPTIMIZATIONS				Y			
WRITE_EXTERNAL_ STORAGE		Y					
ACCESS_COARSE_ LOCATION		Y		Y			
ACCESS_BACKGROUND_ LOCATION		Y					
ACCESS_FINE_ LOCATION		Y		Y			
GPS		Y					
ACTIVITY_RECOGNITION		Y					
CAMERA				Y			Y
RECEIVE		Y				Y	
BIND_GET_INSTALL_ REFERRER_SERVICE		Y				Y	

Interesting Observations in Relation to the Frameworks: The frameworks indicated in Table 2 endeavour to preserve users' privacy according to their documentations. Hence, apps that leverage these frameworks use the privacy specifications highlighted within the frameworks' documentations. For example, the ExposureNotification API leveraged by `Stopp Corona`, `Smitte|stop`,

`Immuni`, `Apturi Covid`, and `Corona-Warn-App` specifies the use of a decentralised BLE technology for proximity identification and exchange of identifiers between nearby devices [2], thus alerting users of potential exposure to COVID-19 with minimal privacy risk. Hence, based on its specifications[7] and the Google COVID-19 Exposure Notifications Service Additional Terms [14], these apps are required to declare and use normal permissions only, excluding the use of `BLUETOOTH_ADMIN`, other permissions such as Signature, Privileged or Special permissions, and any runtime permissions (unless granted their use by Android), for example `STORAGE`.

Nevertheless, `Corona-Warn-App` is seen to declare a dangerous permission (i.e., `CAMERA`), although its usage is explicitly stated in the privacy policy as a feature required for scanning QR codes for test registration. On the other hand, the ROBERT framework specifies the use of a centralised BLE technology for proximity tracing in fighting COVID-19 by measuring risk exposures between users [5]. Hence, `StopCovid France`, which leverages the framework, should declare the use of `BLUETOOTH`, among other normal permissions, for the purpose of detecting when users are in close proximity. However, it can be noted that the app requests for dangerous permissions (i.e., `CAMERA`, `ACCESS_FINE_LOCATION` and `ACCESS_COARSE_LOCATION`). While the reason for accessing `CAMERA` has been pointed out in the privacy policy as a feature needed to scan a QR code to self report whether a user has tested positive for COVID-19, the reason for accessing location is not mentioned.

Contrary to the above-mentioned frameworks that use BLE technology for proximity tracing, Safe Paths leverages the ubiquitous use of mobile devices to trace and reduce the spread of COVID-19 by allowing users to decentrally log in their time-stamped GPS locations [21] and voluntarily share these data with other users in an event where one is tested positive. As seen in Table 3, `CovTracer`, which utilises the framework, declares not only normal permissions, but dangerous ones such as `ACCESS_COARSE_LOCATION` and `ACCESS_FINE_LOCATION`, which when granted accesses the location of the user. While the developers of this app explicitly state the use and advantage of mobile location data for contact tracing[8], this goes against the general legal analysis highlighted by EDPB that states that *"contact tracing apps do not require tracking the location of individual users. Instead, proximity data should be used."* [9]. Furthermore, a number of studies, for example [4,5], show adversarial models on how public health officials or authorities can use the gathered contact tracing information for other purposes. For instance, while the Safe Paths framework enables health officials with ways of redacting location trails of diagnosed carriers[9], such data could be used for other intentions such as re-identification of users with the purpose of inferring their contact graphs. In addition, even though CovTracer targets users whose movements are not restricted at the present time,

[7] https://developers.google.com/android/exposure-notifications/exposure-notifications-api, Accessed 16.08.2020.
[8] https://covid-19.rise.org.cy/en/, Accessed 28.07.2020.
[9] https://www.media.mit.edu/projects/safepaths/overview/, Accessed 22.10.2020.

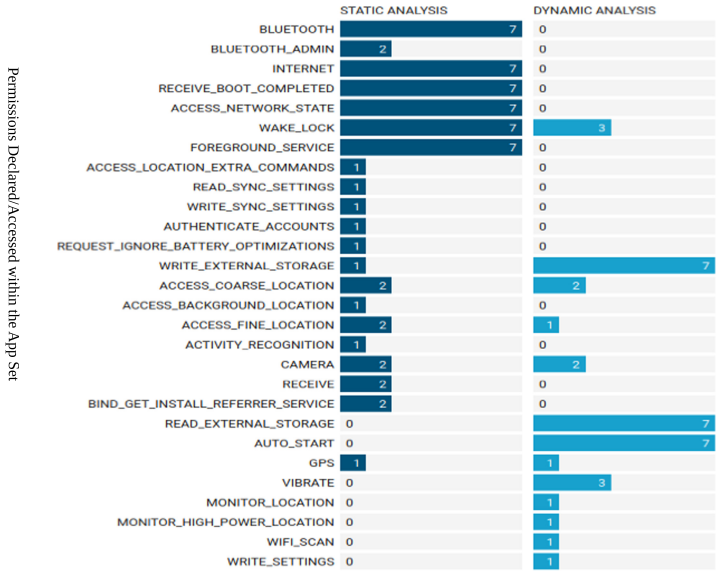

Fig. 1. The chart shows the permissions declared (static analysis) versus the permissions accessed (dynamic analysis) by the apps. The numbers within the bars indicate the number of apps that declare and access that particular permission.

for instance police officers, their data could still be used for other purposes as previously mentioned. As such, this poses a risk in terms of privacy for the users of this app.

4.2 Dynamic Analysis

Figure 1 shows the comparison between what the apps declare in their manifests and what they actually access during runtime. As an assumption, a user would expect the apps to behave in a transparent manner by using permissions that it has actually requested and in a fashion that would not endanger their privacy. However, the actual permission access pattern differs from what has been declared or from what has been mentioned in the apps' privacy policies. Figure 2 shows the visualised results obtained from our app set with and without the user interaction phase respectively. The graphs show the frequency at which the apps accessed the permissions during the period of study. For instance, it can be noted that the frequency at which the apps access permissions without user interaction is slightly less compared to the user interaction phase. Of interest is that apps which leverage the ExposureNotification API tend to access the `READ_EXTERNAL_STORAGE` and `WRITE_EXTERNAL_STORAGE` permissions and at a higher frequency in both phases even though these

permissions were not requested. According to the Android developers[10], *"Android 4.4 (API level 19) or higher apps do not need to request any storage related permissions to access app-specific directories within external storage"*; hence, this can explain this behaviour of these apps that leverage the ExposureNotification framework. Further, [14] highlights that developers should not request any runtime permissions such as `STORAGE` expect in a case where Android Developers have authorised their use; this is because the ExposureNotification API accesses the on-device storage for the purposes of storing the ephemeral proximity identifiers required for contact tracing[11]. Regardless of this, it can be noted that two apps, that is, `Corona-Warn-App` and `StopCovid France` requested access to the use of `CAMERA` which was granted during the user interaction phase when we acted like a user who intended to scan a QR code for test registration or wanted to self report a positive case respectively. Having been granted this permission, it was noted that the *camera* feature was constantly accessed by both apps even with the user having ceased to use the QR functionality. Nevertheless, it can be assumed that the use of this runtime permission was exclusively authorised by Android as its use is relevant in reporting and slowing the spread of COVID-19. Further, `Corona-Warn-App` is shown to access a special permission, that is `WRITE_SETTINGS`, which [14] prohibits the developers leveraging ExposureNotification API from using, and which the user has to grant exclusively if the app aims for API level 23 or higher[12].

A closer look at both graphs also indicate that one of the app, that is `StopCovid France`, is violating PoLP as it accesses a permission that it does not require for its core functionality. This is because the privacy policy mentions that the app uses BLE in tracing and notifying users if they have come into close contact with a positive case or are at risk of COVID-19. In spite of this, however, it can be noted that even though the app's core functionality depends on BLE technology, it still accesses location data through the `ACCESS_COARSE_LOCATION` permission, which provides the approximate location of a user. This does not only violate PoLP, but also contradicts the Commission Nationale de L'informatique et des Libertés (CNIL) opinion, which explicitly states that the app does not track users' location [7] but instead uses BLE functionality for contact tracing. Further, access to location without obvious justification poses a high risk to users privacy as the use of location data could be used to infer the location of a user and their surroundings [12]. As such, one can deduce from the analysis that the use of this app could potentially lead to invasion of privacy. In addition, the issue of under-privilege permissions, that is, the use of permissions that have not been declared in the manifest, arises here. This is because some apps (i.e., `CovTracer`, `StopCovid France` and `Immuni`) fail to declare the use of `VIBRATE`

[10] https://developer.android.com/training/data-storage/app-specific#external, Accessed 23.10.2020.

[11] https://developers.google.com/android/exposure-notifications/exposure-notifications-api, Accessed 16.08.2020.

[12] https://developer.android.com/reference/android/Manifest.permission#WRITE_SETTINGS, Accessed 03.08.2020.

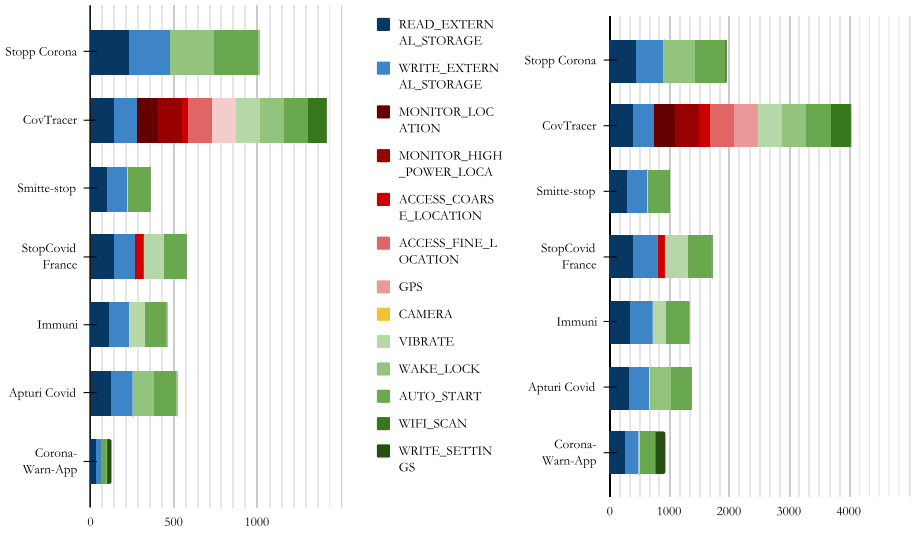

Fig. 2. Permission usage by contact tracing apps for a period of six days: The left bar graph shows permission usage without user interaction while the right bar graph shows the permission usage with user interaction. Of interest, is the access to location by `StopCovid France`, which supposedly uses BLE technology for contact tracing and `CovTracer`, which accesses location when the phone rests (without user interaction). The left graph also shows that the apps are (very) active when the phone rests.

permission in their manifests but access it regardless. However, the use of the vibrate permission or its declaration in the manifest file can be omitted by using the performHapticFeedback() function of a View thus vibrating once to deliver response on a user action[13].

Extrapolation of Permission Access Usage: According to Hansen et al. [16], the inability to distinguish a user in a large data set is associated with data minimisation and purpose limitation. As such, it can be argued that when too much information is collected, which goes beyond the app's specified purpose, it could lead to the identifiability of the user. For instance, Fig. 3 shows the user identities derived from the permissions accessed by `StopCovid France` through the use of the aforementioned partial identity model. As indicated, the app violates PoLP by accessing components that go beyond its core functionality. Through this, the app violates the principle of data minimisation and purpose limitation by collecting location data, which goes against its core functionality as indicated in its privacy policy. Further, with the ROBERT framework allowing the collected data to be stored centrally, the "honest-but-curious" government could be able to infer the whereabouts of the user in question, together with

[13] https://stackoverflow.com/questions/56213974/androids-performhapticfeedback-vs-vibrator-documentation-and-use, Accessed 27.10.2020.

their address as shown in Fig. 3, which could further be used to deduce their contact graph [5] or create a hot spot mapping.

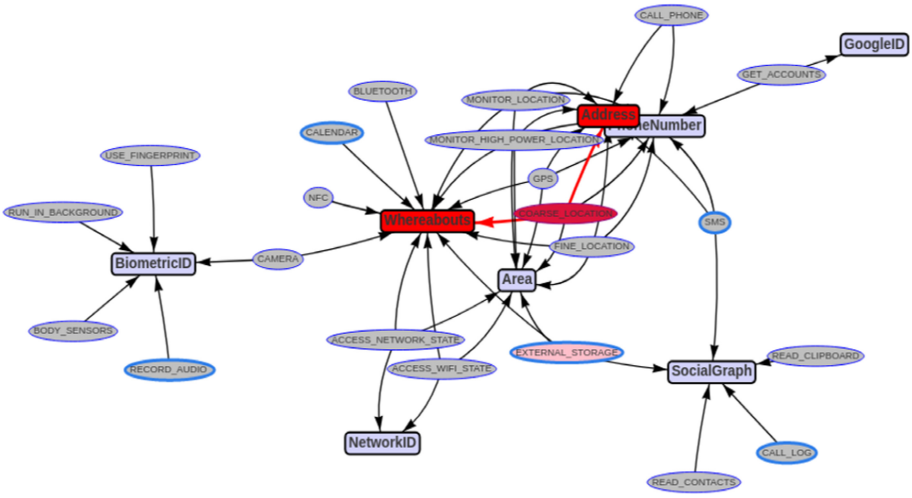

Fig. 3. Derivation of identity attributes from permission accessed by StopCovid France. Identity attributes are highlighted in dark red color (rectangle), with the permissions access contributing to this highlighted in maroon (ellipse) (Color figure online)

4.3 Privacy and Data Protection Goals Analysis:

As highlighted earlier, the property of transparency ensures that the end user is aware of the entire process concerning their personal data, which includes what data is being processed [16]. In the context of the contact tracing apps, it is expected that such information is documented in the apps' privacy policies where users can learn what data is being considered for processing. Considering the principle of data minimisation (Art. 5 (1)(c) GDPR), the apps are required to collect information that are relevant and necessary for their functionality. On the other hand, the principle of purpose limitation (Art. 5 (1)(b) GDPR), limits the collection of data to only specific, definite and lawful purposes. Hence, in terms of transparency, `StopCovid France` is the only app within the app set that violates these two principles as it can be noted in its privacy policy that it does not mention location as a category of data to be processed.

In the context of intervenability, it can be noted that majority of the apps give users control over their personal data via the apps user interface. This promotes users trust towards the use of the app. Art. 6 (1)(a) GDPR mentions consent as a basis for lawful processing of personal data with the conditions specified in Art. 7. From the analysis, it can be noted that 71% of the apps request users consent as the legal basis for processing users data, which includes

data collection, use or any form of disclosure. This empowers users with control over their personal data as they have been granted with the right to withdraw their consent at any given time thus preventing further processing. However, two of the apps within the app set, i.e., `StopCovid France` and `Immuni` quote Art. 6 (1)(e) - which mentions public interest as the basis for lawful processing. With this being one of the acknowledged basis for lawful processing, Art. 6 (3)(a) permits EU member states to impose such a law, which leaves users with limited control over there personal data. Further analysis of these two apps indicate that users cannot exercise their rights. On one hand, `StopCovid France` assures users privacy by stating that personal data processed are pseudonymized; hence, Art. 15, 16 and 18 cannot be exercised. However, the user has the right to erasure (Art. 17) as they can delete data on both their device and the central server by uninstalling the app. On the other hand, users using `Immuni` cannot exercise the rights on Art. 15-20 as the re-identification of users is impossible due to data anonymisation. Notwithstanding, the user, under Art. 21, has the right to object the processing of their data by uninstalling the app, which gradually deletes the data on the central server over a period of fourteen days. Despite these, both apps, like the rest of the other apps, comply with the right for a user to lodge a complaint and contact the Data Protection Officer (DPO) if need be. While this is the case, it can be interpreted that such little control for the users to exercise their rights undermines the respect for user privacy; however, we assume that the user privacy is being backed up by the implementation of security measures. Inspection of the remaining apps indicate that users have the right to exercise Art. 15-20 via the app's interface.

5 Discussion and Conclusion

Having analysed all the apps in Table 2, we present the following findings:

- The EDPB, under the general legal analysis, point out that contact tracing apps need not trace users using location as proximity data is considered sufficient [9]. However, it can be noted that `CovTracer` tends to use location to track users instead of proximity data, even though the developers of this app specify the use of location data. This can also be noted from `StopCovid France` - where the app utilises location data by accessing the `ACCESS_COARSE_LOCATION`.
- `StopCovid France` not only violates PoLP by accessing more than is required for its core functionality when it accesses location data, but also violates the principle of data minimisation and purpose specification which cause the app to collect more than it requires. This data could be used in ways that the user least expects. For example, developers of the ROBERT framework, which the app leverages, document an adversarial model that indicate how the authority could use centrally gathered data for other purposes, for example, re-identification of users [5].

In regards to our contribution and based on the findings in this research, we note that, the identified contact tracing apps in Table 2 play an important role when

it comes to curbing the spread of the pandemic. All apps provide privacy policies that explain clearly to the users what kind of data the apps collect and how they use these data. Further, the apps leverage privacy preserving frameworks that ensure privacy of users. However, while this is the case, it was noted that two apps tend to go against the EDPB recommendations. For example, `StopCovid France`, which leverages a privacy preserving framework that specifies the use of BLE, uses location on top of proximity data without actually being transparent to the users. The use of location data has been shown to be of high risks to users in the context of contact tracing apps, as such, the privacy of users does not need to sacrificed in order to slow the spread of the virus. This applies to `CovTracer` as well, which uses location data for the purposes of contact tracing.

Limitations: Conventionally, apps are dynamic in nature. As such, the reliability of this study would be questionable as the results would lack reproducibility. This would include the results of the apps behaviour with and without user analysis, which would ultimately affect the visualisation of partial identity graphs. In addition, we consider the possibility of false-positives in our data set as we observed instances whether the apps accessed permissions which had not been declared in their `AndroidManifest.xml` files, for example, `VIBRATE`.

Conclusion: In summary, a user would expect apps within the EU Member states to be privacy friendly due to the strong data protection rules. However, while a majority of the apps tend to be privacy friendly, a few are not. For example, `StopCovid France` tends to access course location which gives the approximate location of a user, and goes against PoLP. On the other hand, `CovTracer` does not follow on the EDPB recommendations, which highlight that an app should not track a user using location data, but instead proximity data using BLE. Hence, certain measures need to be taken when developing these apps. For instance, the developers need to follow the guidelines issued by EDPB that highlight the general legal analysis for contact tracing apps. Further, the principles relating to processing of personal data need to be followed.

Acknowledgement. This research is funded by the DigitalWell Research Project from Region Värmlad, Sweden.

References

1. Aisec, F.: Pandemic contact tracing apps: DP-3T, PEPP-PT NTK, and ROBERT from a privacy perspective. IACR Cryptol. ePrint Arch. **2020**, 489 (2020)
2. Apple&Google: Exposure notification: Bluetooth® specification v1.2 (2020)
3. Bell, J., Butler, D., Hicks, C., Crowcroft, J.: Tracesecure: towards privacy preserving contact tracing. arXiv preprint arXiv:2004.04059 (2020)
4. Brack, S., Reichert, L., Scheuermann, B.: Decentralized contact tracing using a DHT and blind signatures. IACR Cryptol. ePrint Arch. **2020**, 398 (2020)
5. Castelluccia, C., et al.: Robert: robust and privacy-preserving proximity tracing (2020)

6. Cho, H., Ippolito, D., Yu, Y.W.: Contact tracing mobile apps for COVID-19: privacy considerations and related trade-offs. arXiv preprint arXiv:2003.11511 (2020)
7. CNIL: Publication of CNIL's opinion on the French "contact tracing" application known as "STOPCovid" (2020)
8. EC: ehealth network: mobile applications to support contact tracing in the EU's fight against COVID-19 - common EU toolbox for member states, version 1.0 (2020)
9. EDPB: Guidelines 04/2020 on the use of location data and contact tracing tools in the context of the COVID-19 outbreak (2020)
10. Fernandes, N.: Economic effects of coronavirus outbreak (COVID-19) on the world economy. Available at SSRN 3557504 (2020)
11. Ferretti, L., et al.: Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. Science **368**(6491), eabb6936 (2020)
12. Fritsch, L.: Profiling and location-based services (LBS). In: Hildebrandt, M., Gutwirth, S. (eds.) Profiling the European Citizen, pp. 147–168. Springer, Dordrecht (2008). https://doi.org/10.1007/978-1-4020-6914-7_8
13. Fritsch, L., Momen, N.: Derived partial identities generated from app permissions. Open Identity Summit 2017 (2017)
14. Google: Google COVID-19 exposure notifications service additional terms (2020)
15. Hammad, M., Bagheri, H., Malek, S.: Determination and enforcement of least-privilege architecture in android. In: 2017 IEEE International Conference on Software Architecture (ICSA), pp. 59–68. IEEE (2017)
16. Hansen, M., Jensen, M., Rost, M.: Protection goals for privacy engineering. In: 2015 IEEE Security and Privacy Workshops, pp. 159–166. IEEE (2015)
17. Hatamian, M., Momen, N., Fritsch, L., Rannenberg, K.: A multilateral privacy impact analysis method for android apps. In: Naldi, M., Italiano, G.F., Rannenberg, K., Medina, M., Bourka, A. (eds.) APF 2019. LNCS, vol. 11498, pp. 87–106. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21752-5_7
18. Klonowska, K.: The COVID-19 pandemic: two waves of technological responses in the European Union (2020)
19. Momen, N., Pulls, T., Fritsch, L., Lindskog, S.: How much privilege does an app need? Investigating resource usage of android apps (short paper). In: 15th Annual Conference on Privacy, Security and Trust (PST), pp. 268–2685. IEEE (2017)
20. Ponce, A.: COVID-19 contact-tracing apps: how to prevent privacy from becoming the next victim. ETUI Research Paper-Policy Brief 5 (2020)
21. Raskar, R., et al.: Apps gone rogue: maintaining personal privacy in an epidemic. arXiv preprint arXiv:2003.08567 (2020)
22. Saltzer, J.H., Schroeder, M.D.: The protection of information in computer systems. Proc. IEEE **63**(9), 1278–1308 (1975)
23. Vaudenay, S.: Centralized or decentralized? The contact tracing dilemma. IACR Cryptol. ePrint Arch. **2020**, 531 (2020)
24. Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., Zhang, X.: Exploring permission-induced risk in android applications for malicious application detection. IEEE Trans. Inf. Forensics Secur. **9**(11), 1869–1882 (2014)
25. Wuyts, K., Scandariato, R., Joosen, W.: LIND(D)UN privacy threat tree catalog. CW Reports (2014)