computers

MDPI

Article

PriADA: Management and Adaptation of Information Based on Data Privacy in Public Environments

Hugo Lopes [1], Ivan Miguel Pires [2,3], Hector Sánchez San Blas [4], Raúl García-Ovejero [4] and Valderi Leithardt [1,5,6,*]

[1] Departamento de Informática, Universidade da Beira Interior, 6201-001 Covilhã, Portugal; lopeshma@gmail.com
[2] Instituto de Telecomunicações, Universidade da Beira Interior, 6201-001 Covilhã, Portugal; impires@it.ubi.pt
[3] Computer Science Department, Polytechnic Institute of Viseu, 3504-510 Viseu, Portugal
[4] Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca, Spain; hectorsanchezsanblas@usal.es (H.S.S.B.); raulovej@usal.es (R.G.-O.)
[5] COPELABS, Universidade Lusófona de Humanidades e Tecnologias, 1749-024 Lisboa, Portugal
[6] VALORIZA, Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal
* Correspondence: valderi@ipportalegre.pt

Received: 15 August 2020; Accepted: 26 September 2020; Published: 28 September 2020

Abstract: The mobile devices cause a constant struggle for the pursuit of data privacy. Nowadays, it appears that the number of mobile devices in the world is increasing. With this increase and technological evolution, thousands of data associated with everyone are generated and stored remotely. Thus, the topic of data privacy is highlighted in several areas. There is a need for control and management of data in circulation inherent to this theme. This article presents an approach to the interaction between the individual and the public environment, where this interaction will determine the access to information. This analysis was based on a data privacy management model in open environments created after reading and analyzing the current technologies. A mobile application based on location by Global Positioning System (GPS) was developed to substantiate this model, which considers the General Data Protection Regulation (GDPR) to control and manage access to each individual's data.

Keywords: data privacy; mobile devices; environment privacy; General Data Protection Regulation (GDPR)

1. Introduction

Mobile devices have become an everyday element in each individual's daily life and have passed a use for enjoyment has to be fundamental to their daily activity. It is used not only for leisure but also is used for work due to the significant development that has suffered in recent years for its technological breakthrough. That is why each machine is a private information source about the person who owns it and who surrounds it. As these data are considered to belong to the individual, collection and treatment cannot be carried out without the individual's consent. Based on this assumption, the processing and collection of private data in a non-consensual way damages and violates our privacy and may damage it [1]. The sharing of any data depends on the individual's perception and willingness to share such private data, respecting each individual's privacy preferences, which is the study's primary motivation [2].

Because mobile devices do not work in isolation, if not to the use of interconnected telephone networks, all information generated is too. Therefore, information security is always essential because all the data inserted in the networks will be available for life [3,4]. Privacy and security are critical for systems related to healthcare or other subjects [3,5].

The privacy and security in mobile information systems are essential because the mobile devices are equipped with several sensors that may capture sensitive data related to the user, such as location, previous communications, and other sensors data capable of identifying the different users [6,7]. Still, the privacy of the users is always crucial in the different developed systems.

The scope of this work is the analysis of all the environmental situations with the technology involved together with the guarantee of the correct treatment of their data. Thus, the main motivation of this work consists in the creation of a data privacy management model based on the creation in public environments. It includes the comparison and study of the state of the art considering the existing solutions.

The novelty of this study is related to the use of mobile devices to manage the privacy and security of users in public environments. Currently, there is a dispersion of public hotspots that allows different types of people to access network services, therefore, the implementation of these mechanisms is essential since other types of people can use these systems.

As a result of this theoretical study of existing data privacy management models in the market, a mobile application has been created which improves certain aspects of previous models. This mobile application uses the location of the Global Positioning System (GPS) and the time of the device as the basis for determining access to information in each defined location, thus implementing a model of management and privacy of public data. The system was only tested in controlled environments to demonstrate the reliability of the mobile application and would soon be tested with real users. This paragraph ends the introduction. Section 2 presents the background of the proposed solution. The materials and methods are presented in Section 3, showing the requirements and details of the implementation in Sections 4 and 5. The validation of the solution was offered in Section 6. Finally, the discussion and conclusions of this research are presented in Section 7.

2. Background

Data privacy has had a vast prominence in society. Several approaches are taken to realize the dream of one day. There could be a world in which there is a real state of privacy for the individual. For such privacy to exist, it is necessary to consider aspects such as the individual's behavior, existing technologies, political, economic, and social limits [8]. Mobile devices are among the most significant sources of information about each individual, as they reflect habits, tastes, and characteristics related to each one. Considering that mobile devices have such data, there is a need to control and manage how it is done [2]. Several studies have been done about the privacy and security of Internet of Things technologies, because it is a crucial subject where the security is essential as it mainly uses distributed and fog computing [9–12] and the comparison of several algorithms [13,14]. The use of mobile applications is also expanding, and the privacy of the data inserted on them should be controlled [15,16]. Several literature reviews have been performed with the recent years, where we detailed some of the systems related to this project [17–20].

2.1. Comparison with Prior Work

For the elaboration of the privacy management model in public environments, we analyzed the related work developed between 2017 and 2020.

The authors of [21] presented an Android platform intending to be customizable and secure. However, when connected to the Internet, the user cannot be sure that his device is safe from external agents' malicious attempts. Considering this factor and since smartphones are part of many people's daily lives, they are, consequently, a source of private information, since all users keep photos, videos, messages, among other types of data on your device. After analysis, it is inferred the absence of a scoring system that measures the degree of disclosure and privacy of an application and that the best way to prevent leakage of private information is to pay attention to requests for permission by the application and not to press the button "Accept" without any consideration.

In [22] is presented an analysis of the frequency of events on screen in Android applications. This analysis uses the concept of local differential privacy (LDP), which is based on global differential privacy (DP) to preserve user data. Differential privacy is a study in the area of statistics and data analysis that uses hash tables, subsampling (subsampling), and addition of noise (noise injection) to allow collective learning (crowdsourced learning) with the aim of to keep user data private. In the DP model, the responses (data inputs in the application) entered by the user are provided to a trusted entity to be treated. After performing the DP analysis, the problem is that the natural results are provided to customers (application producers) who perform unreliable analyzes. The LDP model has an additional constraint to the previous process. Even if a customer has access to the analysis responses in the database, he/she will not be able to carry out any analysis/learning of his/her data, as this model does not aggregate the raw data centrally.

In [23], a study is made of the analysis libraries existing on the Android operating system. We also analyze privacy policies in some applications and the effect that actions derived from collecting user data through libraries have on applications. It was found that some libraries operate under specific permissions that the user gives to each application. Grants, such as internet access, Wi-Fi status, access to location, files in memory, and device settings, are considered for library data collection. Although the application can operate on specific data, it does not mean that the collection process carried out by some analysis libraries is a correct practice. Some of these libraries deliberately use these permissions to obtain non-essential private data, that is, depending on the library that a particular application uses, for a given set of authorized permissions, that library can collect, or not, more data than what is strictly necessary. Due to this collection and consequent data leak, the mobile application: ALManager was created. This application uses the exposed structure to manage analysis libraries to reduce threats to user's private data. This management is done in two ways, first allowing the user to examine the information collected by the analysis libraries. It then allows the user to specify which applications can collect data through libraries and block specific libraries in other applications.

The authors of [24] categorize the permissions into invasive (i.e., access to personal information, camera, microphone, Bluetooth, and location) and generic, validating this classification with the Naive Bayes algorithm, thus constituting an evaluation model. With probabilities, this model determines, within a set of permissions, how many are invasive to privacy and the extent to which the requested authority may be harmful in the future. Specific applications can be considered and advised as unsuitable for installation and use.

In [25], an analysis is made of the practices carried out by the producers of Android applications related to the existing privacy policies. It was found that many applications collect and process users' private data. However, there is difficulty identifying whether the implementation of the application code is under the application's privacy policy. The GATOR framework (GUI analysis structure) structure was adapted, and the detection of privacy violations based on hierarchical mapping was developed. This mapping is how the application interface is created. Thus, the collection and processing of information by the application when entering data by the user (e.g., in a specific text field), must conform to the application claims to handle.

The authors of [8] deals with innovation in the way of using large amounts of data (Big Data) related to the Internet of Things (IoT). Consequently, there is an aim to improve technological products, services, and other general improvements in society's capabilities. This innovation is associated with some data privacy and environmental security problems. The heart of the problem is related to collecting, using, and managing big data at the intersection of security and privacy requirements. Here, innovation is reflected in new ways of using large amounts of data, often inappropriately and that compromise users' privacy and the security of systems.

In [26], the authors propose a method to reduce this problem. Using the PAU (Privacy Assessment with Uncertain Consideration), an analysis (Machine Learning) was made to data in the communications history, access data in the cloud, and circulates in real-time. After the examination, an algorithm was created that, combining these methods, could improve the accuracy of the evaluation related to the data in circulation. This simulation using the algorithm can serve as a basis for creating more secure intelligent systems that preserve data.

In [27], there was a need to control how users' private information was treated, taking into account the location, called the ubiquitous environment. For this, a privacy model was created that manages personal data in these environments. In this model, a complicated relationship is made between the characteristics of the situation, users, application, communication, and available services. A prototype was developed based on middleware structured in layers to represent the model. It enables the necessary control and management of the environments.

In [28], the authors investigate location-based privacy protection technologies and put together some recent studies representing approaches to this subject. An analysis of the research was carried out with the final objective of evaluating the existing solutions and highlighting possible research directions for future investigations.

In [29], the authors propose a new application scenario called context-sensitive friend discovery based on mobile sensing. Contextual attributes such as location, climate, and temperature are used to improve existing friend discovery schemes. However, data privacy becomes a primary concern for consumers when accepting this application. It was also proposed a context-sensitive friend discovery scheme that preserves privacy to solve this problem, where the user's confidential data are well protected.

In [30], it was proposed an algorithm based on the implementation of t-closeness model to ensure data availability without revealing data private individuals. Thus, processed data can not only obtain the data necessary to provide privacy protection effect, it also satisfies the data request using data availability.

The authors of [31] proposed a system that combines the measurement of soap levels, room capacity, distances, temperature, and humidity with sensors embedded in Long Range Wide Area Network (LoRaWaN) devices. The proposed approach was tested in a real environment, concluding that it presents reliable results to manage and control airport toilets.

Taking into account the fast development of Internet of Things (IoT) technologies, the authors of [32] proposed a lightweight server-aided data monitoring scheme (SIM) to implement distributed computing with sensors to perform data monitoring. However, the authors did not mention details about the different environments captured.

A comparison was obtained between the model performed and the related works considered. In Table 1, we can see that the related works are related to data privacy considering the following approaches, such as user, application, generalized environment, and public environment. Thus, the following definitions will be found:

- Address: the work addresses the requirement addressed;
- Not address: the work does not address the requirement;
- Not described: No information was found about the requirement addressed;
- Under Development: The requirement is still under development. It is usually pointed out frequently in tests, validations, results obtained, or future work.

Table 1. Relation between data privacy approaches and the different studies.

Study	Data Privacy Approaches			
	User	**Application**	**Generalized Environment**	**Public Environments**
May et al. [19]	Address	Address	Not described	Not described
Zhang et al. [20]	Address	Address	Not described	Not described
Liu et al. [21]	Address	Address	Not described	Not described

Table 1. *Cont.*

Study	Data Privacy Approaches			
	User	Application	Generalized Environment	Public Environments
Kesswani et al. [22]	Address	Address	Not described	Not described
Wang et al. [23]	Address	Not address	Not described	Not described
Sollins et al. [6]	Address	Not address	Address	Not described
Feng et al. [24]	Not described	Address	Not described	Not described
Leithardt et al. [25]	Address	Address	Address	Not address
Yan et al. [28]	Address	Address	Not described	Not described
Zhuo et al. [29]	Address	Address	Not described	Not described
Hao et al. [30]	Address	Address	Not described	Not described
Sales Mendes et al. [31]	Address	Address	Address	Not address
Zhao et al. [32]	Address	Address	Not described	Not described
This study	Address	Address	Address	Address

After the analysis is carried out, it is found that there is a continuous search in the fight for the privacy of users' data, both at the scientific and academic level, where it is necessary to take into account all possible approaches. Several essential factors related to this theme, including issues related to user behavior, application behavior, and the environment's context, are decisive.

Finally, it is possible to verify that none of the described works focused on managing data privacy in public environments, which is the main contribution of this work.

2.2. Comparison with Other Solutions

The MoveWithMe [33] arises from the need to hide real location data. It uses a data simulation algorithm that, automatically and in real-time, makes location requests to service providers for fictitious location data and accurate data. Due to this strategy, the algorithm present in the application makes it difficult to identify the service provider, since he cannot distinguish which groups of data are real.

The Priser [34] is a model that uses Bluetooth 4.0, known commercially as Bluetooth Low Energy (BLE). BLE was developed for communication between devices, widely used in the Internet of Things (IoT). The possibility of using a Bluetooth sensor and a GPS receiver was evidenced to check and ensure the user's presence in the environment.

The ShiftRoute [35] is a new Privacy Protection Location Mechanism designed for map services on smartphones. This mechanism allows a user to establish a route between two points without revealing any vital location information, that is, done strategically: the points at each end are changed to nearby points, without providing accuracy. Adjacent points are considered Points of Interest (POIs). These serve as a reference for the real locations.

The SieveDroid [36] was created to prevent unwanted transmissions of private data and prevent degradation of the application's functionalities. This framework was designed with the following features: control over Private Data Transmission (PDT) in the Android system and reveals which sensitive operations of a given application are from the generation of a Private Data Usage (PDU) graph of logs (the process of printing or saving information about the activity of the application code) at run time. After generating the chart, privacy control filters are created.

The UbiPri [27] is a prototype for representing the middleware model for controlling and managing privacy in ubiquitous environments. This prototype uses location-based services for mobile devices.

The PISA [37] is a mobile application used to increase the awareness of privacy risks, stimulating the user to reflect the dangers on data sharing and extraction to promote the education and security of the users with intuitive interfaces.

Thus, Table 2 shows the implementation of local privacy, mobile devices, and geolocation in the different solutions available in the literature.

Table 2. Relation between the implemented features and the applications.

Applications	Approaches		
	Local Privacy	**Mobile Devices**	**Geolocation**
MoveWithMe [33]	Yes	Yes	Yes
Priser [34]	Yes	Yes	Yes
ShiftRoute [35]	Yes	Yes	Yes
SieveDroid [36]	No	Yes	No
UbiPri [27]	Yes	Yes	Yes
PISA [37]	Yes	Yes	No
This study	Yes	Yes	Yes

The solutions presented in Table 2 are distinct between them, but they have the common purpose of contributing to users' data privacy. The data privacy approach in the environment is highlighted in each solution, where these solutions generally use location-based services for mobile devices.

3. Materials

3.1. Definition of Management Model for Privacy in Public Environments

The data privacy management model in the Public Environment is where categories of public environments will be defined to determine access to information. Thus, the criterion that allows data to be stored and processed by the model's representative application is the individual's consent and authorization to provide their data. The previous concept covers all data that circulates through the European network and Internet access. Thus, based on the comparison of the state-of-the-art related to the literature and existing implemented solutions, a model was developed for the data privacy management in open environments with the definition of the following categories:

- Unrestricted Public Environment: Environment without time restrictions or access control;
- Temporarily Unrestricted Public Environment: Time-restricted environment without access control;
- Public Environment of Semi-Restricted Access: Environment without time restriction but with access control;
- Public Restricted Access Environment: Time-restricted environment with access control.

Table 3 presents some examples of public environments related to different categories that are generally attributed. The assignment of a class to a given environment depends on legal factors and rules inherent to it. These legal factors and regulations are essential to determine the access to the ground but are not the focus of this study. It should be something to deepen in the future.

Table 3. Relation between types and environment categories.

Categories	Public Environments			
	Unrestricted Public Environment	**Temporarily Unrestricted Public Environment**	**Public Environment of Semi-Restricted Access**	**Public Restricted Access Environment**
Garden	Yes	No	No	No
Public highway	Yes	No	No	No
Square	Yes	No	No	No
Shopping center	No	Yes	No	No
Gallery	No	Yes	No	No
Parking	No	Yes	No	No
Trade point	No	No	No	Yes
Service	No	No	Yes	No
Institution	No	No	No	Yes

3.2. Definition of Individual Profiles

The proposed model focuses on the interaction between the individual and his/her environment. Thus, the following purposes of personal profiles were considered:

- Levels 1, 2 or 3: the user only has access to the information given by her/his environment;
- Level 4: the Administrator is a user that can access the information provided by the category of his/her environment. This user can also perform operations on the data as well as the users.

Regarding the mentioned profiles, it will be the environment to determine the access to information for all of them.

3.3. Definition of Types of Information

A direct relationship was made with the environment's category to define the types of information that a user can access. Thus, as presented in Table 4, the following levels were defined:

- Level 1: information given by the Unrestricted environment.
- Level 2: information provided by the Unrestricted and Temporarily Unrestricted environment.
- Level 3: information provided by the Unrestricted, Temporarily Unrestricted, and Semi-Restricted environment.
- Level 4: information given by the Unrestricted, Temporarily Unrestricted, Semi-Restricted, and Restricted environment.

Table 4. Relation between the access level of information and environment categories.

Categories	**Public Environments**			
	Level 1	**Level 2**	**Level 3**	**Level 4**
Unrestricted	Yes	No	No	No
Temporarily Unrestricted	Yes	Yes	No	No
Semi-Restricted	Yes	Yes	Yes	No
Restricted	Yes	Yes	Yes	Yes

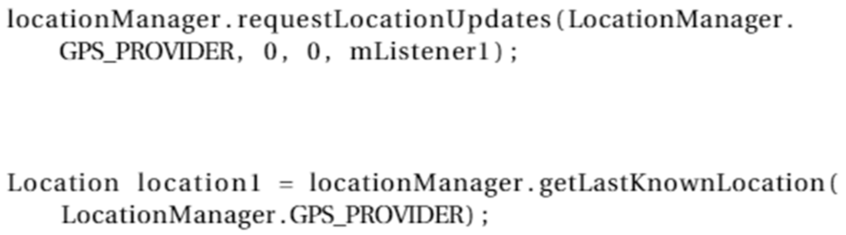
It was considered that the more restricted an environment is, the more sensitive is the information associated with it. It determines the type of information the user will have access. Level 4 of data is classified as the most restricted of the types of information defined.

This model focuses on the public environment since it is where there is the most significant interaction between people, and since it is where there is the most considerable sharing of information. The models studied in previous works focus on the behavior of the user and the application. Still, it is necessary to join a third approach to better control the privacy of this data, i.e., the environment where the user and the application interact. Works like [1,2] also approach the environment, but do not specify the public setting's interaction. There is no categorization and what variables to consider in this interaction, giving even more relevance to this study.

4. Methods

Android Studio IDE as well as Java language were used to create the application. For the correct functioning of the location-based application, essential methods were used:

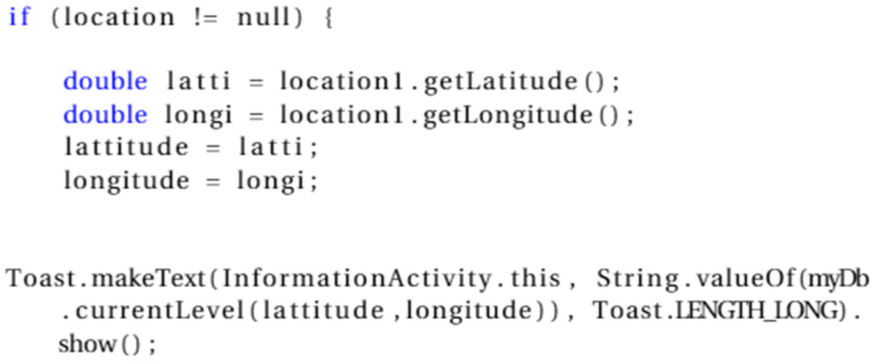
- LocationManager—this class provides access to the system's location services (see Figure 1). These services allow the application to obtain periodic updates of the device's location or launch an intent specified by the application when the device enters a given area. This class requires access to the application manifest, which explains the application's request for internet access and the device's location. In conclusion, this class uses a service and a content provider for its operation and application operation.



```
locationManager.requestLocationUpdates(LocationManager.
    GPS_PROVIDER, 0, 0, mListener1);


Location location1 = locationManager.getLastKnownLocation(
    LocationManager.GPS_PROVIDER);
```

Figure 1. LocationManager class call.

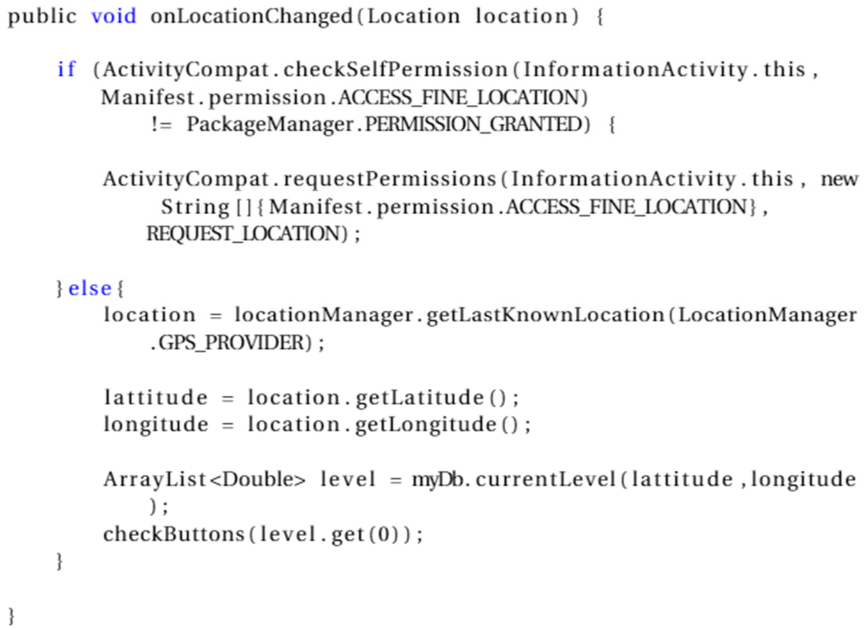
- getLocation—This function allows you first to access the device’s location (see Figure 2) through the coordinates of longitude and latitude obtained using the function currentLevel of the activity DataBaseHelper, the permission level of the location where the user is returned from the comparison with the existing locations in the database.



```
if (location != null) {

    double latti = location1.getLatitude();
    double longi = location1.getLongitude();
    lattitude = latti;
    longitude = longi;


Toast.makeText(InformationActivity.this, String.valueOf(myDb
    .currentLevel(lattitude,longitude)), Toast.LENGTH_LONG).
    show();
```

Figure 2. GetLocation method.

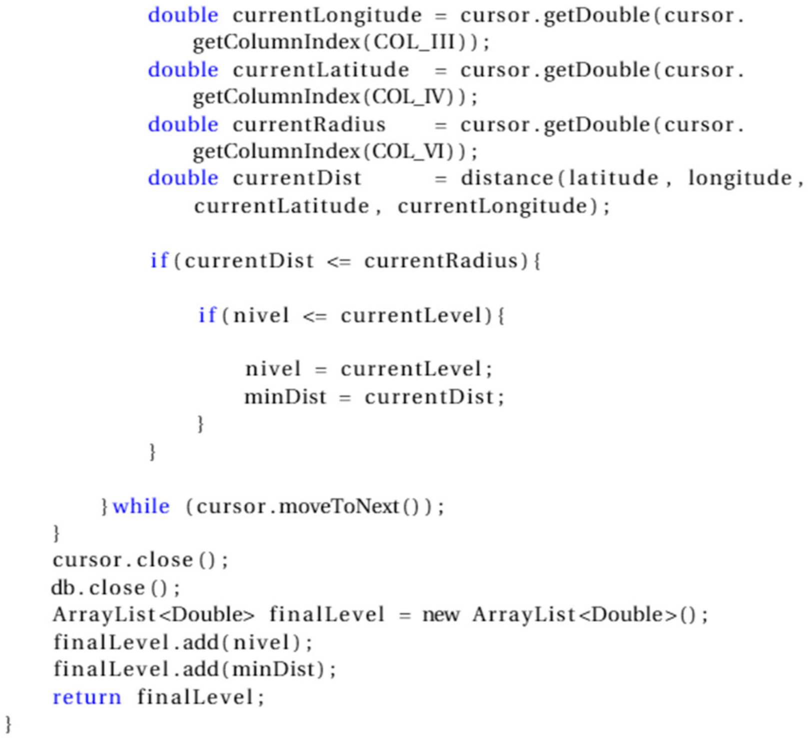
- requestLocationUpdates—This method receives as parameters a minimum distance, in meters, from the required displacement of the device, a minimum time in milliseconds, and the name of the LocationListener to make consecutive calls to the onLocationChanged function (see Figure 3).



```
public void onLocationChanged(Location location) {

    if (ActivityCompat.checkSelfPermission(InformationActivity.this,
        Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {

        ActivityCompat.requestPermissions(InformationActivity.this, new
             String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            REQUEST_LOCATION);

    }else{
        location = locationManager.getLastKnownLocation(LocationManager
            .GPS_PROVIDER);

        lattitude = location.getLatitude();
        longitude = location.getLongitude();

        ArrayList<Double> level = myDb.currentLevel(lattitude,longitude
            );
        checkButtons(level.get(0));
    }

}
```

Figure 3. OnLocationChanged method.

- currentLevel—This function aims to scroll through the table (using the Cursor class) that contains all location points given by longitude and latitude where access to application data is allowed. In turn, each location has an environment category and a maximum radius that delimits the access area. This function will receive the location of the device given by the getLocation function. In the currentLevel function, the distance function is called to calculate the distance. A comparison is made between the calculated distance, the user's location, and the distance radius belonging to a given site, which is in the database. If this computed distance is less than or equal to the radius belonging to a location, the environment's category level is returned. This level will be used by the onLocationChanged function, which will call the checkButtons function to determine the application layout change (see Figure 4).



```
            double currentLongitude = cursor.getDouble(cursor.
               getColumnIndex(COL_III));
            double currentLatitude  = cursor.getDouble(cursor.
               getColumnIndex(COL_IV));
            double currentRadius    = cursor.getDouble(cursor.
               getColumnIndex(COL_VI));
            double currentDist      = distance(latitude, longitude,
               currentLatitude, currentLongitude);

            if(currentDist <= currentRadius){

                if(nivel <= currentLevel){

                    nivel = currentLevel;
                    minDist = currentDist;
                }
            }

        }while (cursor.moveToNext());
    }
    cursor.close();
    db.close();
    ArrayList<Double> finalLevel = new ArrayList<Double>();
    finalLevel.add(nivel);
    finalLevel.add(minDist);
    return finalLevel;
}
```

Figure 4. CurrentLevel method.

- Distance—This function is auxiliary to the currentLevel function. Its main objective is to calculate the distance between a longitude and latitude coordinate point at which the user is and a coordinate point existing in the database, referring to a specific location.
- checkTime—This method allows an element of the user interface to be touched to act. In this method, a variable of the Calendar class was created, called currentTime, which will be used to access the current system time. Considering the defined day time, each of the existing buttons may have an access time, depending on the category of the environment. If this access time matches the device's time, data access will be allowed. A call is made to the database to obtain the data associated with each table.

The main objective was to control access to data, taking into account each user's location, as each environment provided access to different types of data. Also, each set had a specific data access time so that there was a real correspondence with each public space's time.

This control was designed to reduce uncontrolled access and information leakage of personal data. It can be applied in various real everyday scenarios, whether in the different business sectors, including health, education, or even public entertainment.

This study may serve as a basis for the creation/improvement of control and data management systems. However, in this work, the focus is on public environments does not mean that a study cannot be made for other types of environments. One of the reasons that the focus is on public settings was many variables under investigation for greater precision and consistency in the tests, results, and conclusions. Also, one of the objectives of future research could be to address the control and management of private environments and each individual's environment.

5. Application

5.1. Functional Requirements

After authentication, the user would have the possibility to view information, manage users and manage the existing data. Regarding the user management, it also included the option of working users. Finally, information management consisted of the possibility to collect different information.

The main functional requirements were:

- A user with level 4 or Administrator can manage other users and application data;
- A user from level 1 to 3 can check the application data;
- The data query has access mechanisms by location and time.

5.2. Non-Functional Requirements

The non-functional requirements were:

- The user must be registered to perform authentication;
- Only a user with level 4 can manage other users and application data;
- For any user to consult any application data, they will have to authorize the location permission;
- The mobile application requires access to the device's location;
- Installing the application on the mobile device requires 4 Megabytes plus the space of data stored by the mobile application;
- The minimum version of the Application Programming Interface (API) for the application was 23, which corresponds to Android version 6.0 (Marshmallow).

5.3. Implementation

The implementation of the Application Layout was carried out using a purely guiding outline. The implementation decisions were influenced by the different needs, which led to the constant modification of this outline. The main goal was to make operations logical and straightforward. In implementing the layout, the eXtensible Markup Language (XML) was used, using the Android Studio IDE. For the back end, the Java language was used.

5.3.1. Registration

It was necessary to create a record where the fields name, password, description, and a profile level (from level 1 to 4) were filled into the user to have access control to the login application. When an attempt was made to register a user, in turn, the fields were checked before being entered into the local database to ensure that there were no repeated users. The methods used for this purpose were called insertData and addData. The first was in the activity that controlled the DataBaseHelper Database, while the second was in the Registration activity, called RegisterActivity. In the RegisterActivity, it was also possible to edit or delete users. The updateData and updateUser functions were used to modify the user's data. The first was in the RegisterActivity activity and the second in the DataBaseHelper. Finally, to be able to delete a user, the functions deleteData and deleteUser were used. The first was found in the class assigned to the registration of the user RegisterActivity and the second in the activity DataBaseHelper, aimed at controlling the database.

5.3.2. Authentication

Regarding the implementation of the authentication process, presented in Figure 5, it included the verification of the fields introduced when registering the user. This process took place in the LoginActivity activity. The functions used for this purpose were called checkLogin and validate. The first was in the class that controlled the database called DataBaseHelper, while the second was located in the Login activity called LoginActivity.

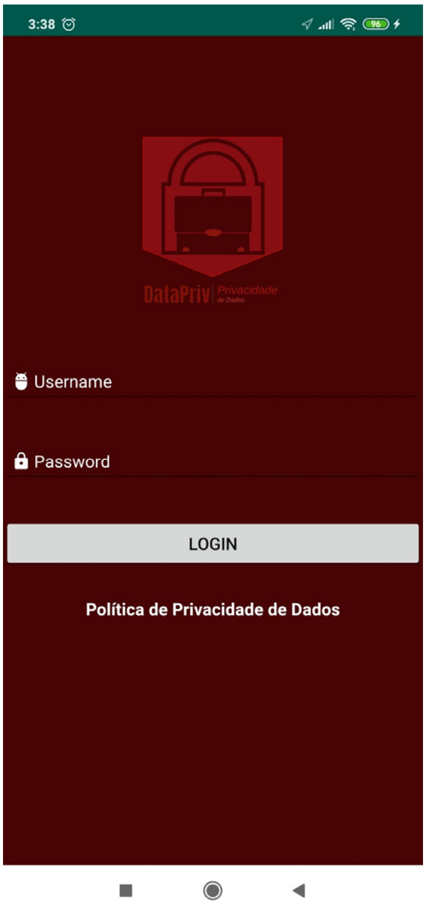

3:38
DataPriv Privacidade de Dados
Username
Password
LOGIN
Política de Privacidade de Dados

Figure 5. Login screen.

5.3.3. Data Management

In the AdminActivity activity, the users with level 4 could manage other users, insert, edit, and remove data in the application and access the data query activity. The UserActivity for the remaining levels also allowed connection to the activity where the information was contained. When the user entered InformationActivity for the first time and had the GPS turned off, they were activated. If the user refused, he was prevented from proceeding and redirected to the previous activity, where he will have to repeat the process. Finally, suppose the user allowed access, and the Global Positioning System (GPS) was active. In that case, the application obtained the current location given by latitude and longitude, allowing it to remain in the activity. After getting the device's location, the application only allowed access to the data if the user was within an existing location in the database, at a maximum distance (radius in meters) defined for each location point. If the user checked this condition, considering the environment's category, he would see different types of data, as visible/invisible buttons, that had been implemented for this purpose. Finally, for each type of information belonging to a specific category of environment, there may be a need to be accessed at a time defined in the application. This check was done using the device's date/time using the Calendar class.

Figure 6 shows the management screen related to the users and information. Next, Figure 7 shows the data retrieved by the mobile application. In Figure 7, it is also possible to verify that the user was in a Restricted Access category location, approximately where he/she is 10 m from the point defined by latitude and longitude in the database. Finally, Figure 8 shows the working machine used connected in real-time to the mobile testing device.

However, the security to protect the users' information, performing the following procedures with AES (Advanced Encryption Standard) algorithm:

- assisted the encrypt and decrypt functions of the data inserted by the different users;
- encrypted any information chosen before being inserted in the database;
- deciphered any existing data in the database when viewing the content or checking the fields in the screens.

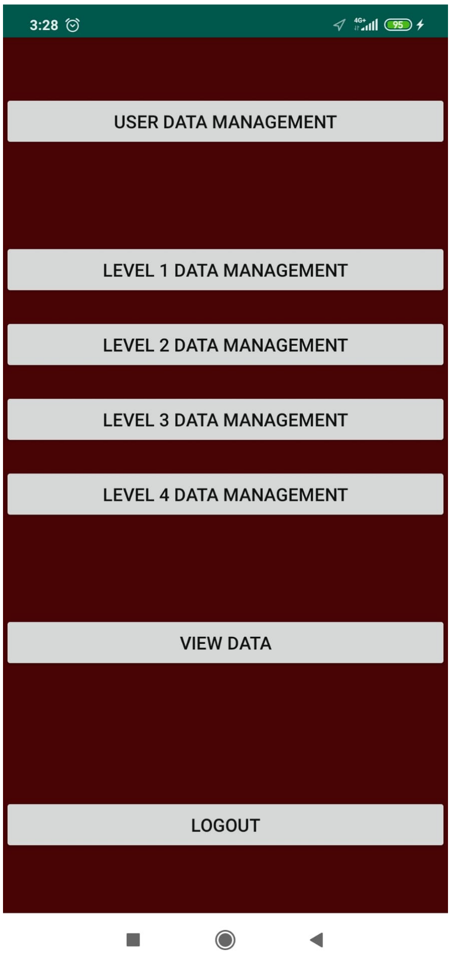
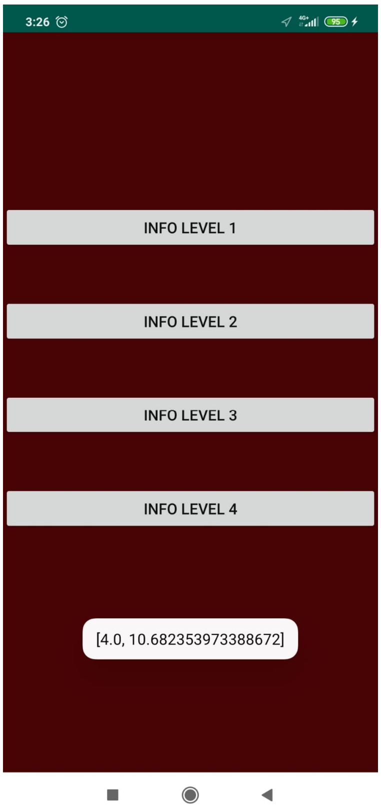


Figure 6. Management of information and users.



[4.0, 10.682353973388672]

Figure 7. Information retrieved.

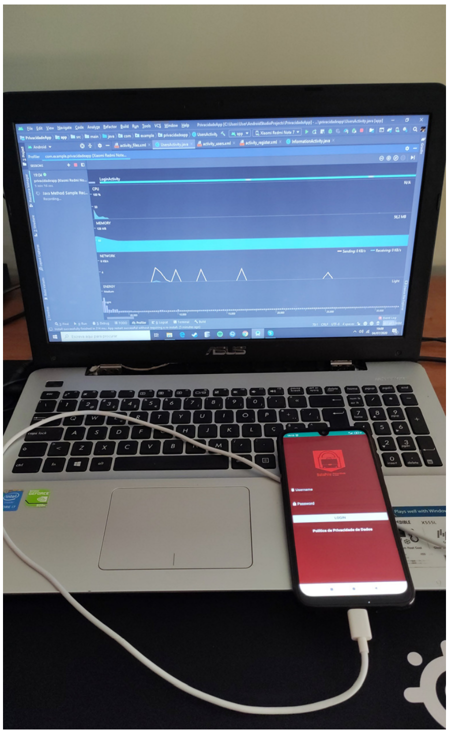


Figure 8. Testing environment of the mobile application.

In Figure 8, the user was in a location with level 4, so four buttons for accessing information can be seen. It was also possible to view the distance to the location point defined in the database.

In Figure 8, it was possible to see the operation of the application in real-time. On the screen of the working machine used, it was possible to observe the application's general monitoring graph and on the mobile testing device the login on the mobile application.

6. Validation

6.1. Test Cases

Some of the application tests were done through real situations in different public environments. For a better understanding, the main tests performed were:

- Registration/Authentication: Users with varying levels of permission were created to test the functioning of the user registration/authentication. In the first attempt to register a new user, the field "name" was placed equal to that of an existing one in the database, and the application prevented registration. In the second attempt to register a new user, the "name" field was placed differently from an existing one in the database, and the application successfully registered. After the user was introduced, a user verification was performed in the database when the authentication attempt was made. In the first attempt, the user existed in the database, and the application saved the permission level, so the user was redirected to the activity corresponding to his permission level. In the second attempt, as the user's existence in the database was not verified, he was prevented from entering any activity;
- Location: The coordinates of an "A" location have been inserted away from the device's position. Next, a maximum distance delimited to the bound and ensured that the user could only consult the data within an area defined by the application. As this distance did not reach the device's current location, he was not allowed to view any data access button. To test the contrary case, the user walked towards the location "A" defined by the application. As the user walked towards location "A", the application would automatically update its location and check if its place was within the maximum distance from location "A". As there was a match, the buttons became visible, and the corresponding data could be viewed depending on the location's permission. For the tests described above to be possible, first, the application asked for permission to access the device's location, it was refused, and the user was prevented from proceeding and redirected to the previous activity corresponding to his permission level. The same attempt was made again, and when requesting access to the device's location, it was accepted, and the GPS was not active, so he was asked for permission to activate it. The user refused and was immediately prevented from proceeding and redirected to the previous activity where he had to repeat the process previously described until all conditions were met. In the last attempt, the conditions were all checked, and the user managed to remain in the information query activity;
- Data consultation time: To ensure that the user could only consult the data at a specific time (day of the week and time), for an information (information buttons), different consultation times were introduced. The test was done for two different kinds of information. In the first, a consultation time was added outside the time the device. The second type of information was within the consultation time imposed by the device. As the first type of information was outside the application's schedule, it prevented the user from consulting information. Finally, as the second type of data was within the limit imposed by the application, it allowed data to be asked.

6.2. Performance Tests

Android Profiler [38] was used to test the application's performance. It is a tool for monitoring the real-time performance of the application provided by Android Studio [39]. This tool focuses on monitoring the application in the following aspects:

- Energy consumption through the Energy Profiler [40];
- Memory allocation from Memory Profiler [41];

- CPU (Central Processing Unit) activity through the CPU Profiler [42];

Figure 9 shows the global chart that evaluates each of the existing components in the Android Profiler [38] mentioned above.

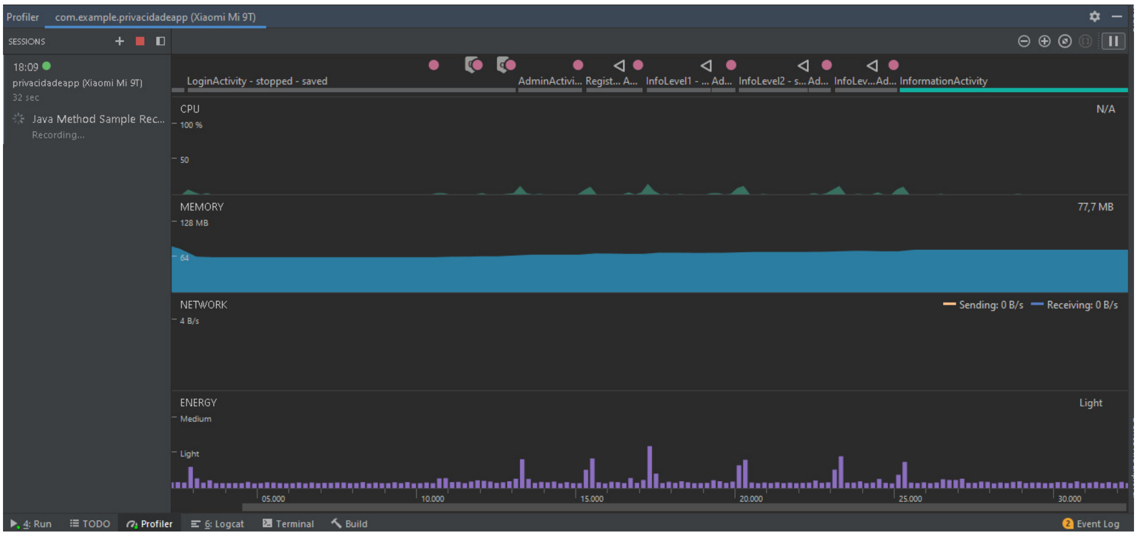

Profiler
com.example.privacidadeapp (Xiaomi Mi 9T)
SESSIONS
18:09
privacidadeapp (Xiaomi Mi 9T)
32 sec
Java Method Sample Rec...
Recording...
LoginActivity - stopped - saved
AdminActivi... Regist... A... InfoLevel1 - ... Ad... InfoLevel2 - s... Ad... InfoLev...Ad... InformationActivity
CPU
N/A
100 %
50
MEMORY
77,7 MB
128 MB
64
NETWORK
Sending: 0 B/s
Receiving: 0 B/s
4 B/s
ENERGY
Light
Medium
Light
05.000
10.000
15.000
20.000
25.000
30.000
4: Run
TODO
Profiler
6: Logcat
Terminal
Build
2 Event Log

Figure 9. General chart of application performance at run time.

The previous figure shows the general chart of performance tests performed on the data privacy management model's representative application in public environments.

It is possible to see low energy consumption, low processor usage, and a small memory allocation. Thus, it is possible to verify that, in the worst case, the application used 8% of the device's processing capacity, presenting an average usage between 2% to 3%.

7. Discussion and Conclusions

This research's motivation consisted of the development of a data privacy management model focused on the public environment, since current models could be improved. For this, a mobile application was created that uses the GPS location and the device's time as a basis to determine access to information in each site.

This model, together with other application models related to data privacy, can form a model considered complete that analyzes all situations that occur in the environment in which any technology is inserted and therefore guarantee users a correct treatment of their data.

Based on the comparison of state of the art referring to Table 1, it can be seen that the elaborated work contributes to an improvement in data privacy since, in addition to making the relationship between the individual and information, it addresses the public environment in which the individual is inserted. In this way, with the categorization carried out, it is possible to determine which are the most critical public environments where there should be a different treatment of each individual's information. From this categorization, it is also possible to define what type of data will be made available in each environment, thus directly forming the environment and information. In this way, we will see an improvement in managing the individual's privacy in public settings. The implementation referring to the section where software engineering is described using case diagrams and activity diagrams has as focus the use of GPS and permission to access the user's location.

From the test scenarios within a set of public environments, it reflects the application's functioning and response to changes in context. Thus, it is possible to obtain real results on the interaction with the individual.

As this theme's study reached more significant proportions, there was a need to study the central theme in more depth. Consequently, it was concluded that despite the application functioning well, issues such as GPS accuracy and energy consumption derived from the use over a long time could limit its operation.

Finally, we can conclude through tests performed using the Android Profiler tool [31] that satisfactory results were obtained in terms of computational resources and energy consumption.

In the future, it would be interesting to make some improvements in terms of precision and energy consumption using Bluetooth Low Energy [34], using a database connected to a server for better treatment, exchange, and management of information between the database and the application. In addition to the points mentioned, issues such as application security and the use of Artificial Intelligence algorithms could bring improvements in the application's operation.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, writing—original draft preparation, writing—review, project administration and editing: H.L. and V.L., visualization, writing—review and editing: I.M.P., R.G.-O. and H.S.S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by FCT/MEC through national funds and co-funded by FEDER-PT2020 partnership agreement under the project UIDB/50008/2020. This work was partially supported by Fundação para a Ciência e a Tecnologia under Project UIDB/04111/2020.

Acknowledgments: This work is funded by FCT/MEC through national funds and when applicable co-funded by FEDER-PT2020 partnership agreement under the project UIDB/50008/2020. (*Este trabalho é financiado pela FCT/MEC através de fundos nacionais e cofinanciado pelo FEDER, no âmbito do Acordo de Parceria PT2020 no âmbito do projeto UIDB/50008/2020*). The authors also like to acknowledge the collaboration of Computer lab 7 of Instituto de Telecomunicações - IT Branch Covilhã. This article is based upon work from COST Action IC1303-AAPELE—Architectures, Algorithms and Protocols for Enhanced Living Environments and COST Action CA16226–SHELD-ON—Indoor living space improvement: Smart Habitat for the Elderly, supported by COST (European Cooperation in Science and Technology). More information in www.cost.eu.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Standard
API	Application Programming Interface
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
DP	Differential Privacy
GDPR	General Data Protection Regulation
GPS	Global Positioning System
GUI	Graphical User Interface
IoT	Internet of Things
IDE	Integrated Development Environment
LDP	Local Differential Privacy
PAU	Privacy Assessment with Uncertain Consideration
PDT	Private Data Transmission
PDU	Private Data Usage
POI	Points of Interest
XML	Extensible Markup Language

References

1. Leithardt, V.R.Q.; Geyer, C.F.R.; Silva, J.M.S. *Controle e Gerenciamento de Privacidade de Dados*; Novas Edições Acadêmicas: Lisbon, Portugal, 2019; ISBN 978-3-8417-1533-3.
2. Leithardt, V.; Santos, D.; Silva, L.; Viel, F.; Zeferino, C.; Silva, J. A Solution for Dynamic Management of User Profiles in IoT Environments. *IEEE Latin Am. Trans.* **2020**, *18*, 1193–1199. [CrossRef]
3. Sousa, P.S.; Sabugueiro, D.; Felizardo, V.; Couto, R.; Pires, I.; Garcia, N.M. mHealth Sensors and Applications for Personal Aid. In *Mobile Health*; Adibi, S., Ed.; Springer Series in Bio-/Neuroinformatics; Springer International Publishing: Cham, Switzerland, 2015; Volume 5, pp. 265–281, ISBN 978-3-319-12816-0.

4. Marques, G.; Pires, I.M.; Miranda, N.; Pitarma, R. Air Quality Monitoring using Assistive Robots for Ambient Assisted Living and Enhanced Living Environments through Internet of Things. *Electronics* **2019**, *8*, 1375. [CrossRef]
5. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Rodríguez, N.D. Validation techniques for sensor data in mobile health applications. *J. Sens.* **2016**, *2016*. [CrossRef]
6. Barsocchi, P.; Calabrò, A.; Crivello, A.; Daoudagh, S.; Furfari, F.; Girolami, M.; Marchetti, E. A Privacy-By-Design Architecture for Indoor Localization Systems. In *Quality of Information and Communications Technology*; Communications in Computer and Information Science; Shepperd, M., Brito e Abreu, F., da Rodrigues Silva, A., Pérez-Castillo, R., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 1266, pp. 358–366, ISBN 978-3-030-58792-5.
7. Kaaniche, N.; Laurent, M.; Belguith, S. Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey. *J. Netw. Comput. Appl.* **2020**, 102807. [CrossRef]
8. Sollins, K.R. IoT Big Data Security and Privacy Versus Innovation. *IEEE Internet Things J.* **2019**, *6*, 1628–1635. [CrossRef]
9. Katsikas, S.; Gkioulos, V. Security, Privacy, and Trustworthiness of Sensor Networks and Internet of Things. *Sensors* **2020**, *20*, 3846. [CrossRef]
10. Lim, J.-H.; Kim, J.-W. Privacy-Preserving Aggregation of IoT Data with Distributed Differential Privacy. *J. Korea Soc. Comput. Inf.* **2020**, *25*, 65–72. [CrossRef]
11. Affonso Souza, C.; César de Oliveira, C.; Perrone, C.; Carneiro, G. From privacy to data protection: The road ahead for the Inter-American System of human rights. *Int. J. Hum. Rights* **2020**, 1–31. [CrossRef]
12. Guo, X.; Wang, W.; Huang, H.; Li, Q.; Malekian, R. Location Privacy-Preserving Method Based on Historical Proximity Location. *Wirel. Commun. Mobile Comput.* **2020**, *2020*, 1–16. [CrossRef]
13. Saraiva, D.A.F.; Leithardt, V.R.Q.; de Paula, D.; Sales Mendes, A.; González, G.V.; Crocker, P. PRISEC: Comparison of Symmetric Key Algorithms for IoT Devices. *Sensors* **2019**, *19*, 4312. [CrossRef]
14. Qi, L.; Hu, C.; Zhang, X.; Khosravi, M.R.; Sharma, S.; Pang, S.; Wang, T. Privacy-aware Data Fusion and Prediction with Spatial-Temporal Context for Smart City Industrial Environment. *IEEE Trans. Ind. Inf.* **2020**. [CrossRef]
15. Sophus Lai, S.; Flensburg, S. A proxy for privacy uncovering the surveillance ecology of mobile apps. *Big Data Soc.* **2020**, *7*. [CrossRef]
16. Lutz, C.; Hoffmann, C.P.; Ranzini, G. Data capitalism and the user: An exploration of privacy cynicism in Germany. *New Media Soc.* **2020**, *22*, 1168–1187. [CrossRef]
17. Ismagilova, E.; Hughes, L.; Rana, N.P.; Dwivedi, Y.K. Security, Privacy and Risks within Smart Cities: Literature Review and Development of a Smart City Interaction Framework. *Inf. Syst. Front* **2020**. [CrossRef]
18. Yang, P.; Xiong, N.; Ren, J. Data Security and Privacy Protection for Cloud Storage: A Survey. *IEEE Access* **2020**, *8*, 131723–131740. [CrossRef]
19. Anusha Linda Kostka, J.E.; Vinila Jinny, S. Data Security and Privacy Protection in Cloud Computing: A Review. In *Intelligence in Big Data Technologies—Beyond the Hype*; Advances in Intelligent Systems and Computing; Peter, J.D., Fernandes, S.L., Alavi, A.H., Eds.; Springer: Singapore, 2020; Volume 1167, pp. 253–257, ISBN 9789811552847.
20. Chanal, P.M.; Kakkasageri, M.S. Security and Privacy in IoT: A Survey. *Wirel. Pers. Commun.* **2020**. [CrossRef]
21. May, Z.E.; Kaffel Ben Ayed, H.; Machfar, D. State of the art on Privacy Risk Estimation Related to Android Applications. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; IEEE: Tangier, Morocco, 2019; pp. 889–894. [CrossRef]
22. Zhang, H.; Latif, S.; Bassily, R.; Rountev, A. Introducing Privacy in Screen Event Frequency Analysis for Android Apps. In Proceedings of the 2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM), Cleveland, OH, USA, 30 September–1 October 2019; IEEE: Cleveland, OH, USA, 2019; pp. 268–279. [CrossRef]
23. Liu, X.; Liu, J.; Zhu, S.; Wang, W.; Zhang, X. Privacy Risk Analysis and Mitigation of Analytics Libraries in the Android Ecosystem. *IEEE Trans. Mob. Comput.* **2020**, *19*, 1184–1199. [CrossRef]
24. Kesswani, N.; Lyu, H.; Zhang, Z. Analyzing Android App Privacy with GP-PP Model. *IEEE Access* **2018**, *6*, 39541–39546. [CrossRef]

25. Wang, X.; Qin, X.; Hosseini, M.B.; Slavin, R.; Breaux, T.D.; Niu, J. GUILeak: Tracing privacy policy claims on user input data for Android applications. In Proceedings of the 40th International Conference on Software Engineering, Gothenburg, Sweden, 27 May–3 June 2018; ACM: Gothenburg, Sweden, 2018; pp. 37–47. [CrossRef]
26. Feng, X.; Wang, L. PAU: Privacy Assessment method with Uncertainty consideration for cloud-based vehicular networks. *Future Gener. Comput. Syst.* **2019**, *96*, 368–375. [CrossRef]
27. Leithardt, V.R.Q. UbiPri: Middleware Para Controle e Gerenciamento de Privacidade em Ambientes Ubíquos. Ph.D. Thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, 2015. Available online: https://lume.ufrgs.br/handle/10183/147774 (accessed on 28 September 2020).
28. Yan, Y.; Gai, K.; Jiang, P.; Xu, L.; Zhu, L. Location-based Privacy-Preserving Techniques in Connected Environment: A Survey. In Proceedings of the 2019 IEEE International Conference on Smart Cloud (SmartCloud), Tokyo, Japan, 10–12 December 2019; IEEE: Tokyo, Japan, 2019; pp. 156–162. [CrossRef]
29. Zhuo, G.; Yang, H. Privacy-preserving context-aware friend discovery based on mobile sensing. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 12–14 January 2018; IEEE: Las Vegas, NV, USA, 2018; pp. 1–5. [CrossRef]
30. Hao, G.; Ya-Bin, X. Research on privacy preserving method based on T-closeness model. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; IEEE: Chengdu, China, 2017; pp. 1455–1459. [CrossRef]
31. Sales Mendes, A.; Jiménez-Bravo, D.M.; Navarro-Cáceres, M.; Reis Quietinho Leithardt, V.; Villarrubia González, G. Multi-Agent Approach Using LoRaWAN Devices: An Airport Case Study. *Electronics* **2020**, *9*, 1430. [CrossRef]
32. Zhao, M.; Ding, Y.; Wu, Q.; Wang, Y.; Qin, B.; Fan, K. Privacy-Preserving Lightweight Data Monitoring in Internet of Things Environments. *Wirel. Pers. Commun.* **2020**. [CrossRef]
33. Kang, J.; Steiert, D.; Lin, D.; Fu, Y. MoveWithMe: Location Privacy Preservation for Smartphone Users. *IEEE Trans. Inf. Forensic Secur.* **2020**, *15*, 711–724. [CrossRef]
34. Silva, L.A.; Valderi, R.Q.L.; Rudimar, S.D.; Silva, J.S. Priser—Utilização De Ble Para Localização E Notificação Com Base Na Privacidade De Dados. *Rev. Eletrônica Argent-Bras. Tecnol. Inf. Comun.* **2018**. [CrossRef]
35. Zhang, P.; Hu, C.; Chen, D.; Li, H.; Li, Q. ShiftRoute: Achieving Location Privacy for Map Services on Smartphones. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4527–4538. [CrossRef]
36. Huang, J.; Xiong, Y.; Huang, W.; Xu, C.; Miao, F. SieveDroid: Intercepting Undesirable Private-Data Transmissions in Android Applications. *IEEE Syst. J.* **2020**, *14*, 375–386. [CrossRef]
37. Toresson, L.; Shaker, M.; Olars, S.; Fritsch, L. PISA: A Privacy Impact Self-assessment App Using Personas to Relate App Behavior to Risks to Smartphone Users. In *HCI International 2020—Posters*; Communications in Computer and Information Science; Stephanidis, C., Antona, M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 1226, pp. 613–621, ISBN 978-3-030-50731-2.
38. Measure App Performance with Android Profiler. Available online: https://developer.android.com/studio/profile/android-profiler (accessed on 11 July 2020).
39. Android. The Platform Pushing what's Possible. Available online: https://www.android.com/ (accessed on 10 July 2020).
40. Inspect Energy Usage with the Energy Profiler. Available online: https://developer.android.com/studio/profile/energy-profiler (accessed on 10 July 2020).
41. View Java Heap and Memory Allocations with Memory Profiler. Available online: https://developer.android.com/studio/profile/memory-profiler (accessed on 10 July 2020).
42. Inspect CPU Activities with the CPU Profiler. Available online: https://developer.android.com/studio/profile/cpu-profiler (accessed on 11 July 2020).