

FAIR: Fuzzy Alarming Index Rule for Privacy Analysis in Smartphone Apps

Majid Hatamian^{1*}, Jetzabel Serna¹, Kai Rannenber¹ and Bodo Igl²

¹ Chair of Mobile Business and Multilateral Security
Goethe University Frankfurt
Frankfurt am Main, Germany

{majid.hatamian, jetzabel.serna, kai.rannenber}@m-chair.de

² RheinMain University of Applied Sciences
Wiesbaden, Germany
bodo.igler@hs-rm.de

Abstract In this paper, we introduce an approach that aims at increasing individuals' privacy awareness. We perform a privacy risk assessment of the smartphone applications (apps) installed on a user's device. We implemented an app behaviour monitoring tool that collects information about access to sensitive resources by each installed app. We then calculate a privacy risk score using a fuzzy logic based approach that considers type, number and frequency of access on resources. The combination of these two concepts provides the user with information about the privacy invasiveness level of the monitored apps. Our approach enables users to make informed privacy decisions, i.e. restrict permissions or report an app based on resource access events. We evaluate our approach by analysing the behaviour of selected apps and calculating their associated privacy score. Initial results demonstrate the applicability of our approach, which allows the comparison of apps by reporting to the user the detected events and the resulting privacy risk score.

Keywords: smartphone apps; privacy; usability; beacon alarming; privacy risk score; fuzzy logic

1 Introduction

Security and privacy have always been a serious concern in the field of information technology in diverse applications such as computer networks, wireless communications, etc [1]. This is even more serious when it comes to smartphone apps since they provide context-sensitive services to the users. The impressive prosperity of the Android Operating System (OS) has become even more evident with its domination over the smartphone market (with a share of 87.6% in 2016

* The authors would like to thank: A. Paterno, D. Mattes, D. Wowniuk, M. Duchmann, M. Krapp, and R. Dieges for providing the app. This research work has received funding from the H2020 Marie Skłodowska-Curie EU project "Privacy&Us" under the grant agreement No 675730.

Q2 [2]). Its prevalence, and openness characteristics have facilitated the development of apps with access to a multiplicity of sensitive resources, resulting in highly personalised and context-sensitive services that benefit users' online interactions and consequently their daily lives. However, and not surprisingly, it has also become the main target of a number of security and privacy related attacks (e.g., 97% of malicious mobile malware [3] targets Android). Furthermore, the huge proliferation of over-privileged apps also poses important privacy risks on the users, who are often unaware of such risks [4].

In this regard, Android's permission model has evolved from a binary one to a more advanced one, which is based on the principle of least privilege [5]; this model enables users to selectively grant/deny specific permissions to each of the installed apps even in run time. However, despite the recent advances, many users continue to ignore those warnings and blindly grant most permissions to apps as they request them [6]. Paradoxically, they continue to express discomfort once they realise that their data are being collected without their informed consent. As shown in [4], this behavior is mostly because of users not being fully aware of the associated privacy risks, partly because of the lack of appropriate information about the use of resources; i.e. which resources are being accessed by which apps and with which frequency; and secondly, a poor understanding of what the privacy consequences are. Thus, in this paper, we propose an app behaviour monitoring tool called '*Beacon Alarming*' system, which makes users aware of the extent to which an app is accessing sensitive resources. In particular, it shows which resources are accessed with what frequency, and whether the user was actually interacting with the app or not. Contrary to the state-of-the-art approaches based on static analysis of code, our system does not require the instrumentation of the Android OS. Secondly, we propose '*FAIR*' as a new method for privacy risk assessment in smartphone apps by using fuzzy logic while considering resource access. FAIR relies on the existence of the beacon alarming system. To the best of our knowledge, this is the first time that fuzzy logic is used as a decision-making method for privacy risk assessment in smartphone apps. We further elaborate the proposed approach with a user-friendly GUI by mapping the Android sensitive resources name to a more descriptive definition to make it more persuasive and easy to understand for the users. Additionally, once users are aware of the privacy issues, the GUI allows users to perform two different actions, either block permissions, or report an app. Finally, we empirically validate the functionality of our proposed approach through initial experiments that monitor real apps behaviour.

The rest of this paper is organized as follows. In Section 2 we review the existing work in the literature. Section 3 introduces the proposed architecture of the monitoring tool called beacon alarming. In Section 4 we introduce the proposed method for privacy risk assessment of smartphone apps called FAIR. Section 5 examines and evaluates the functionality of the proposed approach. Finally, we present the main conclusions and point out our future research direction in Section 6.

2 Related Work

Several efforts have been done to improve the user awareness of privacy and help them to make informed decisions [7–9]. These approaches are based on the advantages of including privacy facts in app descriptions in the app stores. Although, it was believed that this would enable users to make more rational decisions before downloading an app. These approaches could not efficiently operate. This is due to the fact that, during installation, users usually pay limited attention to permission screens and have poor understanding of what the permissions mention. In [10], the authors introduced a method to make smartphone apps more privacy-friendly through automated testing, detecting and analysing privacy violations. They suggested the use of an automated privacy-testing system to efficiently explore an app’s functionality, logging relevant events at multiple levels of abstraction as the app executes, and using these logs to accurately characterise app behavior. Although this is an interesting method, there is no fine-grained formulation for their proposed privacy-testing system, as well as no practical implementation. There are also some approaches based on fine-grained control over permissions and majority voting recommendations [11–13]. These approaches enable users to turn on and off the access to sensitive data or functionality (e.g. SMS, camera, microphone, contacts, etc.) on an app-by-app basis to determine whether they feel comfortable granting it or not. In fact, in such solutions, a privacy control approach is provided to enable selectively granting, denying or confining access to specific permissions on a certain app. This of course is inline with our research and as well with the most recent Android’s permission model. Nevertheless, such solutions must be complemented with additional mechanisms that will first enable users to better understand the behavior of apps and the privacy implications. Following this direction, the authors in [14], proposed to identify permission hungry apps by considering the set of permissions declared by apps in the Apps store, and making a comparison of the commonly used permission in order to make users aware of apps asking for rare or too many permissions. Authors in [15] explored the privacy behavior of apps based on the analysis of data flows which required the instrumentation of Android.

Our approach complements previous research in the sense that, the beacon alarming component analyses app privacy-related behaviour providing more scalability as it does not require to modify the Android OS, therefore, no redistribution/installation of a customise OS is required. Our component is privacy-preserving as all information is processed locally (rule-based engine) and does not leave the user’s device. Furthermore, our approach does not analyse the data flows, therefore, mechanisms, such as proxies monitoring users communications are not required. . We, in turn, focus on providing users with only relevant privacy-related information of apps using more understandable indicators. We encourage users to report privacy aggressive practices of apps based on access to individual resources. The FAIR component advances the state of the art in the calculation of a privacy score using fuzzy logic as a decision-making approach and additionally improves the scalability.

3 Beacon Alarming: Log monitoring tool

In this section we introduce the methodology that we followed for the implementation of our proposed monitoring tool called *Beacon Alarming* [16]. This tool reads the logs generated by AppOps - a privacy manager tool which was introduced by Google in Android 4.3 and which is now inaccessible, unless the device is rooted [17]. We monitored the permission requests done by each selected app. Afterwards, we implemented a user awareness component. The access events were analysed and communicated to the user by our awareness module. The outcome of the monitoring tool (it is worth mentioning that our monitoring tool does not require any root access, modification to the Android OS, etc. See Section 3.1). We also use the results obtained from beacon alarming as the input for FAIR component which is described in detail in Section 4.

3.1 Data collection

The goal of this process was to collect data about the accesses to the device resources that each of the selected apps had done, in particular those privacy related. To this end, we focused on the Android permissions classification. Generally, permissions are classified as 'normal' and 'dangerous' [5].

1. Normal: There are permissions that do not pose much risk to the user's privacy or the device's operation. Thus, the system *automatically grants* these permissions.
2. Dangerous: There are permissions that could potentially affect the user's privacy or the device's normal operation. Therefore, the system asks the user to *explicitly grant* these permissions.

We implemented a module that is able to monitor access events to both normal and dangerous permissions. Our tool was designed in such a way that we could select which app to be monitored. Thus, the data collection was done by our tool which read the logs generated by the Android's AppOps manager and collected those entries related to the selected apps and privacy related permissions. As an important note, we identified that the root access is only needed to access the AppOps management system, e.g. to tell the system to deny access to one of the operations that is controlled by AppOps. As a result, we found that in order to view the AppOps logs, there is no need to root the device, and they are accessible to any app with debugging privilege. In order to collect the logs, a timer event is sent to the `PermissionUsageLogger` service periodically. When it is received, the logger queries the AppOps service running on the device for a list of apps that have used any of the operations we are tracking. We then check through that list and for any selected app that has used an operation recently, we store the time at which that operation was used in a local data base. These entries are also used to get a usage count.

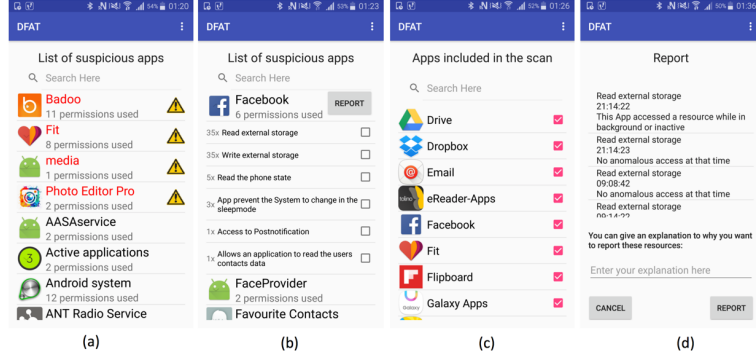


Fig. 1. The proposed beacon alarming (a) list of suspicious apps (b) details of accesses (c) selectively choosing apps to be monitored, and (d) reporting based on the app’s behaviour.

3.2 User interface and communication

In order to provide users with a better understanding of which resources were accessed with what frequency by different apps, we implemented a privacy awareness module (See Fig. 1). In this module users could select which apps to be monitored, and as a result the module displays a summary of apps and resources accessed including the corresponding timestamps. To increase the usability aspects, we mapped/translated the permissions from those defined by Android to a common language definition. Furthermore, we implement a rule-based mechanism that analysed the behaviour of the app in order to inform the user about unexpected behaviours. An example of these rules can be seen in Fig. 2 (due to space limitations, we refrained from representing of all the rules). Finally, to encourage users to take actions when potential privacy risks were detected, our module provided the interfaces to either restrict a permission or to report a resource and therefore, raise awareness of misconduct behaviors and access to sensitive data.

Restrict permissions We provided the interface for direct access to the permission manager system in Android, which allows users to revoke/grant permissions for any app.

Report permissions We developed a semi-automatic reporting tool, where users could select to report an app based on the resource that implied privacy risks. With this tool we aim to simplify the app reporting task and encourage users to report privacy related issues.

3.3 Formalisation of resource accesses

Based on the classification introduced in Section 3.1, we initially consider the set of permissions $\mathcal{P} = \{p_1, \dots, p_n\}$ consisting of two subsets $\mathcal{NP} = \{np_1, \dots, np_m\}$

```

1 #When the display is off and critical resource was used, but
  without the case of taking a phone call
2 if((criticalResources.contains(resource)) && (screenState ==
  0) && !(closeToObject == 0) && !(resource.equals("
  RECORD_AUDIO"))){
3   results.add("1");
4   results.add("Screen was off and critical Resource was used
  ");
5   return results;
6 }

```

Fig. 2. An example of rules for app privacy invasiveness detection.

and $\mathcal{DP} = \{dp_1, \dots, dp_u\}$ where \mathcal{NP} and \mathcal{DP} show the level of permission which is either normal or dangerous, respectively. We also introduce the set of apps by $\mathcal{A} = \{a_1, \dots, a_w\}$. Let $F_{a_i} = \{f_1, \dots, f_e\}$ be the set of features for each app a_i , where $1 \leq i \leq w$. Each f_j ($1 \leq j \leq e$) consists of ordered pairs $\{(p_j, np_k | dp_l)\}$. We determine each feature as an informative element regarding each app. As a result, the set of features related to all app is defined as $\{F_{a_1}, \dots, F_{a_w}\}$, where F_{a_1} represents the feature F_{a_1} associated with app a_1 . Moreover, we denote the used permission p_k ($1 \leq k \leq n$) in \mathcal{P} by app a_i ($1 \leq i \leq w$) in \mathcal{A} while the level of permission $np_g \in \mathcal{NP}$ ($1 \leq g \leq m$) or $dp_f \in \mathcal{DP}$ ($1 \leq f \leq u$) by $\mathcal{L}_{a_i, p_k} = (a_i, p_k)$. Additionally, we formulate the problem by x_{a_i, p_k} as follows:

$$x_{a_i, p_k} = \begin{cases} 1 & \text{if } \mathcal{L}_{a_i, p_k} = (a_i, p_k). \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where if there permission p_k is used by app a_i (\mathcal{L}_{a_i, p_k}), then $x_{a_i, p_k} = 1$, otherwise $x_{a_i, p_k} = 0$.

4 FAIR: Fuzzy Alarming Index Rule

This section introduces FAIR, a novel approach which uses fuzzy logic in order to provide a privacy risks assessment of selected apps. The FAIR approach benefits from the beacon alarming system introduced in Section 3, as shown in Fig. 3 and detailed in Algorithm 1. We exploit fuzzy logic as an appropriate method that is widely adopted in a mixed variety of IT systems such as wireless networks, intrusion detection systems, etc. [18–20], especially, as it has been extensively employed as the core of decision-making systems [21]. The output of a fuzzy controller is obtained from the fuzzification of inputs using the associated membership functions (MFs). A crisp input is then converted into different members of the associated MFs (based on its value). MF maps the elements of a crisp input into numerical values in the interval $[0, 1]$ and it represents the membership degrees of a variable to a given set. In the following, the expected inputs and output are described.

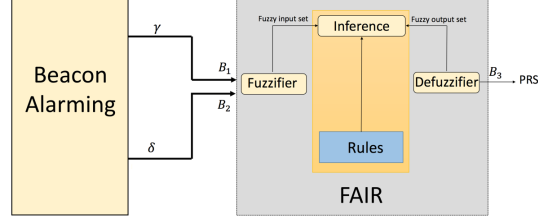


Fig. 3. A high level architecture of FAIR.

Algorithm 1 Pseudo-code of all steps in FAIR.

1. **Procedure FAIR**
 2. **START**
 3. **Define** linguistic variables and terms
 4. **Construct** membership functions **AND** rule base
 6. **START Fuzzification**
 7. **Calculate** $\gamma = T_{a_i}^t |_{\mathcal{DP}} \setminus N_{a_i}^t$ **AND** $\delta = D_{a_i}^t |_{\mathcal{NP}} \setminus N_{a_i}^t$
 7. **Convert** crisp input data to fuzzy values using MFs
 8. **START Inference**
 9. **Evaluate** rules in the rule base
 10. **Combine** results of each rule
 11. **START Defuzzification**
 7. **Calculate** $B^* = \sum_{i=1}^5 \text{COA}_i \cdot \text{Area}_i \setminus \sum_{i=1}^5 \text{Area}_i$
 10. **Convert** output data to non-fuzzy values
 12. **END Defuzzification**
 13. **END Inference**
 13. **END Fuzzification**
 14. **END Procedure**
-

4.1 Fuzzification

The fuzzification process converts crisp data (inputs) into MFs (fuzzy data).

Definition 1. Let γ indicate the first input of FAIR. γ is then defined as follows:

$$\gamma = \frac{T_{a_i}^t |_{\mathcal{DP}}}{N_{a_i}^t}, \quad (2)$$

where $T_{a_i}^t |_{\mathcal{DP}}$ represents the total number of accesses that app a_i has (in time t) to the dangerous privacy sensitive permissions (\mathcal{DP} s), e.g. `READ_CONTACTS`, etc., and $N_{a_i}^t$ shows the total number of accesses that app a_i has to the permissions (both \mathcal{DP} s and \mathcal{NP} s). It is evident that, the value of γ is always in the range $[0, 1]$. γ at its worst case is 1, which means that the app only accesses \mathcal{DP} s. The rationale behind this formulation is to investigate the impact of the access

frequency to privacy sensitive \mathcal{DP} s.

Definition 2. Let δ indicate the second input of FAIR. δ is then defined as follows:

$$\delta = \frac{D_{a_i}^t |_{\mathcal{NP}}}{N_{a_i}^t}, \quad (3)$$

where $D_{a_i}^t |_{\mathcal{NP}}$ represents the total number of accesses that a given app a_i has to privacy sensitive \mathcal{NP} s (discussed in Section 3.1, such as `ACCESS_WIFI_STATE`). The main idea behind this mathematical model is to evaluate the importance of accesses to privacy sensitive \mathcal{NP} s. Because of the capability of fuzzy logic in supporting and implementing decision making systems, it enables us to figure out the impact of both γ and δ on the overall evaluation of privacy risk scores (PRSs) simultaneously.

Definition 3. Let A_1 , A_2 , and A_3 represent the crisp sets. Then, $\mu_{B_1} : A_1 \rightarrow [0, 1]$, $\mu_{B_2} : A_2 \rightarrow [0, 1]$ and $\mu_{B_3} : A_3 \rightarrow [0, 1]$ are called the MFs of B_1 , B_2 and B_3 , which define the fuzzy sets B_1 , B_2 and B_3 of A_1 , A_2 and A_3 . To perform fuzzification process, we should map crisp sets into fuzzy sets as follows (B_1 and B_2 indicate the inputs, and B_3 shows the output):

$$B_1 = \gamma \in \{\text{low, medium, high}\}, B_2 = \delta \in \{\text{low, medium, high}\}. \quad (4)$$

$$B_3 = \text{Privacy Risk Score} \in \{\text{VL, L, M, H, VH}\}. \quad (5)$$

Remark 1. The fuzzy variable γ has three fuzzy states including: low, medium and high, and its MFs are shown by Fig. 4(a).

Remark 2. The fuzzy variable δ has three fuzzy states including: low, medium and high, and its MFs are shown by Fig. 4(b).

Remark 3. The output represents the privacy risk score (PRS) and has five fuzzy states including: VL (Very Low), L (Low), M (Medium), H (High), and VH (Very High). Also, its MFs are shown by Fig. 4(c).

As it was previously mentioned, the fuzzy rules are directly obtained based on the number of states defined for the inputs, i.e. the more states we define for the inputs, the more fuzzy rules we need to initiate. Thus, to keep the implementation overhead of our fuzzy inference system (FIS) low [21], three states for each input and five states for the output have been defined with respect to the expert knowledge.

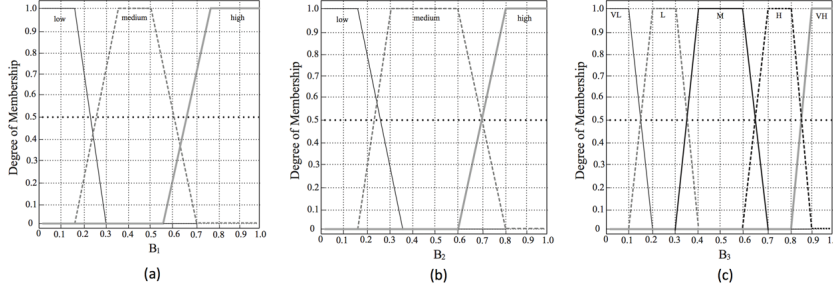


Fig. 4. Fuzzy membership functions (a) B_1 (γ), (b) B_2 (δ), and (c) B_3 (PRS).

4.2 Fuzzy controller rules

The control rule is the core of every FIS. To obtain the fuzzy output, a combination of MFs with the control rules is required. In this paper, we use a standard Mamdani type fuzzy system [21] using a bipartite fuzzifier (since we have two inputs). We show the collection of R fuzzy IF-THEN rules as follows:

$$\forall R^i : \exists b_1 \text{ AND } b_2 \mid \text{IF } b_1 \text{ is } B_1^i, b_2 \text{ is } B_2^i, \text{ THEN } b_3 \text{ is } B_3^i, \quad (6)$$

where B_1^i , B_2^i , and B_3^i are fuzzy states which were defined in the previous section, respectively. To obtain the fuzzy system output, we should define the fuzzy rules. In a Mamdani fuzzy system, the number of possible rules is defined as $N_{\text{inputs}} \times N_{\text{mf}}$, where N_{inputs} is the number of inputs, and N_{mf} is the number of MFs. As a result, we have nine fuzzy rules ($3^2 = 9$) which are shown in Table 1.

Table 1: FIS Rules

No.	B_1	B_2	B_3
1	low	low	VL
2	low	medium	L
3	low	high	H
4	medium	low	M
5	medium	medium	M
6	medium	high	H
7	high	low	VH
8	high	medium	VH
9	high	high	VH

For the sake of simplicity, we used a trapezoidal function to model the fuzzy sets. So that, the calculations of MFs are simpler, as opposite to other functions

such as Sigmoidal, Guassian, etc. The trapezoidal function T is defined as below:

$$T(b_1; \varepsilon, m_1, m_2, \zeta) = \begin{cases} \frac{b_1 - \varepsilon}{m_1 - \varepsilon}, & \text{if } b_1 \in [\varepsilon, m_1]. \\ 1, & \text{if } b_1 \in [m_1, m_2]. \\ \frac{\zeta - b_1}{\zeta - m_2}, & \text{if } b_1 \in [m_2, \zeta]. \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

In (7), ε and ζ , and m_1 and m_2 are the valleys and climaxes, respectively. This trapezoidal function maps b_1 to a value between $[0, 1]$ and the degree of membership called $\mu(b_1)$ is then generated. In fact, we should use (7) to obtain the membership degrees of B_1 , B_2 and B_3 ($\mu_{B_1}(b_1)$, $\mu_{B_2}(b_2)$, and $\mu_{B_3}(b_3)$). In other words, $\mu_{B_1}(b_1), \mu_{B_2}(b_2), \mu_{B_3}(b_3) \in [0, 1]$, and $B_1 = (b_1, \mu_{B_1}(b_1) | b_1 \in A_1)$, $B_2 = (b_2, \mu_{B_2}(b_2) | b_2 \in A_2)$, and $B_3 = (b_3, \mu_{B_3}(b_3) | b_3 \in A_3)$.

4.3 Defuzzification

Defuzzification is aimed to discover the numerical result of our fuzzy system and calculate the crisp output B^* . We use the center of areas (COA) method for defuzzification [21]. In this method, the fuzzy logic controller first calculates the area under the scaled MFs and within the range of the output variable. Then, it uses the following equation to calculate the geometric center of this area:

$$\text{COA} = \frac{\int \mu_z(z)zdz}{\int \mu_z(z).dz}. \quad (8)$$

5 Analysis and Results

In this section, we evaluate the functionality of our proposed approach. In our experimental setup, we have chosen three sets (categories) of apps in which, each set comprises five apps (in total 15 apps). The rationale behind the selection of those apps is as follows: we selected those apps resulting from the first search result page when the user type a certain keyword. The keywords were selected from the most popular app categories. The selected apps were chosen from the top charts in Google Play store, i.e. apps with more than one million downloads. Fig. 5 depicts a graphical representation of the proposed FIS that lets us examine the output surface of the FIS for any one or two inputs. In other words, this graphical interface simply shows us how γ (B_1) and δ (B_2) are mapped to the output (B_3). The colors change according to the output values.

5.1 Theoretical analysis

In this subsection, we mathematically analyse the functionality of FAIR in estimating privacy risk scores (B^*). Fig. 6 re-illustrates the MFs for the output which were previously shown by Fig. 4(c). Let N and D be the numerator and

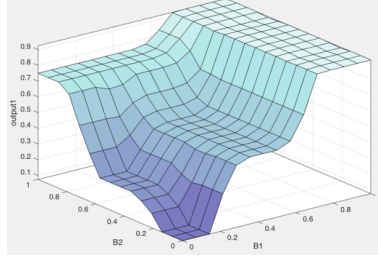


Fig. 5. The graphical structure of the proposed FIS for privacy risk assessment.

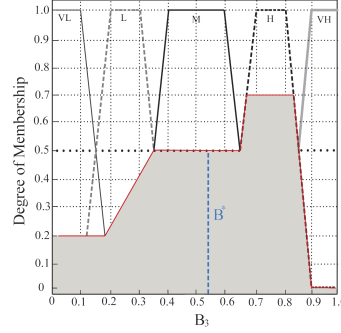


Fig. 6. Calculation of B^* .

denominator of (8), respectively. Then, we have:

$$N = \left[\int_0^{0.18} (0.2)zdz + \int_{0.18}^{0.35} \left(\frac{z - 0.18}{3} \right) z dz + \int_{0.35}^{0.65} (0.5)z dz \right. \\ \left. + \int_{0.65}^{0.68} \left(\frac{z - 0.65}{2} \right) z dz + \int_{0.68}^{0.83} (0.7)z dz + \int_{0.83}^{0.9} (0.9 - z)z dz \right]. \quad (9)$$

$$D = \left[\int_0^{0.18} (0.2)dz + \int_{0.18}^{0.35} \left(\frac{z - 0.18}{3} \right) dz + \int_{0.35}^{0.65} (0.5)dz \right. \\ \left. + \int_{0.65}^{0.68} \left(\frac{z - 0.65}{2} \right) dz + \int_{0.68}^{0.83} (0.7)dz + \int_{0.83}^{0.9} (0.9 - z)dz \right]. \quad (10)$$

If we show the crisp output by B^* , then:

$$B^* = \frac{N}{D} = \frac{0.16153}{0.2984917} \simeq 0.54. \quad (11)$$

The value of B^* (0.54) belongs to membership function M (Medium), where $\gamma(B_1)$ is whether low (when the number of accesses to normal permissions

is negligible), medium (when the number of accesses to normal permissions is moderate, which is between $[0.3, 0.6]$), or high (when the number of accesses to normal permissions is remarkable, which is between $[0.7, 1.0]$), and $\delta(B_2)$ is whether high (when the number of accesses to dangerous permissions is significantly high), medium (when the number of access to dangerous permissions is average, which is between $[0.3, 0.6]$), or low (when the number of access to dangerous permissions is too less). This value means that the combination of dangerous accesses and normal accesses w.r.t the associated fuzzy rules, leads to the situation in which FAIR decides that the overall PRS is M (Medium).

5.2 Experimental analysis

The experiment consists of two separate phases that each took three days. In the first phase of the experiment, we only granted the permissions which are necessary for these 15 apps to work properly. Accordingly, we opened them once and allowed to run in the background without user interaction. Afterwards, we collected and analysed the resources that all these 15 apps were accessing (i.e. permission requests). Fig. 7 shows the results of our analysis (the numbers in each cell show the times that each app had accessed a given permission).

Permissions	Health & Fitness					Social Networks					Dating & Friends				
	S Health	Google Fit	Lifesum	Pedometer	Calorie Counter	Facebook	Twitter	Instagram	LinkedIn	Pinterest	LOVOO	OkCupid	Tinder	Badoo	SayHi
READ_EXTERNAL_STORAGE	594	10	2	5	6	35	16	427	3	14	21	8	14	50	5
WRITE_EXTERNAL_STORAGE	594	10	2	5	6	35	16	427	3	14	21	8	14	50	5
READ_PHONE_STATE	-	-	-	-	-	5	-	-	-	-	-	-	4	35	-
ACCESS_WIFI_STATE	-	-	-	-	-	-	-	-	-	-	-	-	-	57	-
ACCESS_FINE_LOCATION	-	130	-	-	7	-	-	-	-	-	-	-	-	395	-
ACCESS_COARSE_LOCATION	-	-	-	-	-	-	-	-	-	-	-	-	-	5	2
READ_CONTACTS	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-
WRITE_CONTACTS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RECORD_AUDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CAMERA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BODY_SENSORS	425	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 7. Result of app monitoring for the first phase of the experiment.

In the second phase of the experiment, we granted as many permissions as possible to the apps (both normal and dangerous). Afterwards, we monitored them to investigate which one is extensively accessing permissions, even when there is no apparent reason for accessing that permission. It is worth to mention that, during the second phase the only interaction was creating an account in the corresponding apps. The results of this phase is shown in Fig. 8.

Permissions	Health & Fitness					Social Networks					Dating & Friends				
	S Health	Google Fit	Lifesum	Pedometer	Calorie Counter	Facebook	Twitter	Instagram	LinkedIn	Pinterest	LOVOO	OkCupid	Tinder	Badoo	SayHi
READ_EXTERNAL_STORAGE	1067	18	44	1	43	1531	63	580	28	54	143	41	212	349	27
WRITE_EXTERNAL_STORAGE	1067	18	42	1	43	1375	49	583	27	51	118	41	196	343	29
READ_PHONE_STATE	-	-	-	-	-	4	-	-	-	-	-	-	4	176	-
ACCESS_WIFI_STATE	-	-	-	-	-	39	-	-	-	-	-	-	-	170	-
ACCESS_FINE_LOCATION	3	452	-	-	123	346	43	31	-	-	37	35	37	599	-
ACCESS_COARSE_LOCATION	-	-	-	-	16	381	-	-	-	-	4	3	-	47	29
READ_CONTACTS	-	-	-	-	2	5	14	-	6	6	-	-	-	1	-
WRITE_CONTACTS	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-
RECORD_AUDIO	-	-	-	-	-	1	8	2	-	-	-	-	-	1	-
CAMERA	4	-	4	-	4	15	16	27	-	5	-	-	-	10	8
BODY_SENSORS	465	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 8. Result of app monitoring for the second phase of the experiment.

5.3 Initial results

Fig. 9 shows the main results of the two phase experiment. As we can observe, all the PRSs associated with each app has been measured by FAIR based on the resources that they have accessed. It is important to note that we only focused on the resource accesses that are directly related to the users' privacy, e.g. an app which had accessed `WAKE_LOCK` permission was not considered while calculating PRS since this resource access is hardware-oriented.

Fig. 10 shows all the changes that we observed during performing both phases of the experiment regarding each app. This is a sensible way to figure out the variations of PRSs for each installed app to imagine a general overview regarding the privacy invasive behaviours of the apps. It is evident that the invasiveness degree of Social Networks and Dating & Friends apps has tangibly increased since they have unreasonably accessed dangerous permissions even if the user does nothing with the smartphone, e.g. our findings showed that some apps of these two categories had accessed to certain kinds of dangerous permissions (e.g. Camera) even if the smartphone was not being used.

5.4 Discussion

Our findings confirm a considerable difference between the PRSs in both phases of the experiment. This backs up the point that users must pay careful attention while they grant a dangerous permission to an app. This is why we implemented beacon alarming in such a way as to support users for granting/limiting permissions that they feel they might be offended. Moreover, in calculating PRSs, we neglected accesses to the external storage. The logic behind this is twofold. First, in Android there is no way to discriminate different accesses to storage, e.g. we cannot find whether the app accessed photo, video, etc. Second, all apps can read and write files placed on the external storage. It means, this is the basic permissions for every app. Moreover, we have categorised all the resource

App	First Phase		Second Phase	
	B ₃	PRS	B ₃	PRS
S Health	1	VH	1	VH
Google Fit	0.51	M	0.53	M
Lifesum	0	VL	0.93	VH
Pedometer	0	VL	0	VL
Calorie Counter	0.24	L	0.51	M
Facebook	0.07	VL	0.92	VH
Twitter	0	VL	0.81	H
Instagram	0	VL	0.94	VH
LinkedIn	0	VL	0.29	L
Pinterest	0	VL	0.71	H
LOVOO	0	VL	0.50	M
OkCupid	0	VL	0.62	M
Tinder	0.11	VL	0.91	VH
Badoo	0.92	VH	0.89	VH
SayHi	0.14	VL	0.85	VH

Fig. 9. Associated PRSs with each app calculated by FAIR.

accesses, and we only focus on the resources that are directly related to the users' privacy.

The scope of this paper comprises Android OS. Regardless of the choice of the research area, the proposed approach for monitoring resources cannot be applied to other smartphone platforms (e.g. iOS). Furthermore, integrating additional methods of data collection (e.g. user perceptions) could have increased the scope and depth of analyses.

6 Conclusions and Future Work

In this paper, we proposed a new paradigm towards protecting privacy in smartphone ecosystems. FAIR provides a high flexible architecture by applying fuzzy logic to measure the privacy risk score of apps; thanks to the monitoring tool, FAIR is able to inform users about privacy invasive behaviour of apps installed on their devices. To realise the promising properties of FAIR, the essential mathematical formulation, including analysis of normal and dangerous accesses was introduced. Moreover, the GUI has been designed in such a way that we tried to encourage users to review their permissions more efficiently and report apps that showed privacy aggressive practices. We believe that the findings and insights discussed in this paper can encourage privacy researchers to devise more and better privacy functions to address current privacy challenges in smartphone ecosystems. In our future work, we intend to consider not only the behavior of each installed app, but also the expected functionality and declared permissions requirements when measuring the privacy risk score. Furthermore, we aim to carry out an extensive user study in order to better understand the importance

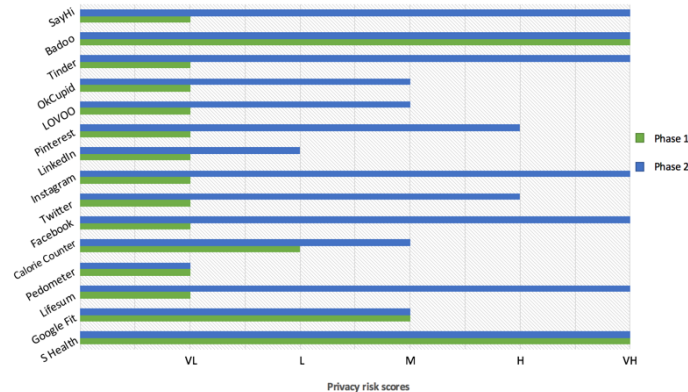


Fig. 10. A comparison between results obtained from both phases of the experiment.

of different privacy related resources, as well as the benefits and potential needs of FAIR from the user perspective.

References

1. A. Naghizadeh, B. Razeghi, E. Meamari, M. Hatamian, and R. E. Atani, “C-trust: A trust management system to improve fairness on circular P2P networks,” *Peer-to-Peer Networking and Applications*, vol. 9, no. 6, 1128–1144 (2016)
2. “Smartphone OS Market Share, 2016 Q2,” accessed December 6th, 2016, <https://www.idc.com/prodserv/smartphone-os-market-share.jsp>
3. “97% of malicious mobile malware targets Android,” accessed December 6th, 2016, <http://www.scmagazineuk.com/updated-97-of-malicious-mobile-malware-targets-android/article/422783/>
4. G. Bal, and K. Rannenberg, “User control mechanisms for privacy protection should go hand in hand with privacy-consequence information: The case of smartphone apps,” in *Proceedings of W3C Workshop on Privacy and User-Centric Controls*, Germany, 1–5 (2014)
5. “Android Developers,” accessed April 6th, 2017, <https://developer.android.com/index.html>
6. A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, “Android permissions: User attention, comprehension, and behavior,” in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, USA, 1–14 (2012)
7. P. G. Kelley, L. F. Cranor, and N. Sadeh, “Privacy as part of the app decision-making process,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, France, 3393–3402 (2013)
8. P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, “A conundrum of permissions: installing applications on an android smartphone,” in *Proceedings of the 26th International Conference on Financial Cryptography and Data Security*, Bonaire, 68–79 (2012)

9. M. Nauman, S. Khan, and X. Zhang, "Apex: Extending android permission model and enforcement with user-defined runtime constraints," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, China, 328–332 (2010)
10. P. Gilbert, B. G. Chun, L. Cox, and J. Jung, "Automating privacy testing of smartphone applications," Technical Report CS-2011-02, Duke University (2011)
11. A. Beresford, A. Rice, and N. Sohan, "MockDroid: trading privacy for application functionality on smartphones," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, USA, 49–54 (2011)
12. Y. Zhou, X. Zhang, X. Jiang, and V. W. Freech, "Taming information-stealing smartphone applications (on Android)," in *Proceedings of the 4th International Conference on Trust and Trustworthy Computing*, USA, 93–107 (2011)
13. P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, "AdDroid: privilege separation for applications and advertisers in Android," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, South Korea, 71–72 (2012)
14. V. F. Taylor, and I. Martinovic, "SecuRank: Starving permission-hungry apps using contextual permission analysis," in *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, Austria, 43–52 (2016)
15. A. P. Felt, E. Chin, S. Hanna, and D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, USA, 627–638 (2011)
16. M. Hatamian, and J. Serna, "Beacon alarming: Informed decision-making supporter and privacy risk analyser in smartphone applications," in *Proceedings of the 35th IEEE International Conference on Consumer Electronics (ICCE)*, USA, 468–471 (2017)
17. "Google removes vital privacy feature from Android, claiming its release was accidental," accessed July 17, 2016, <https://www.eff.org/deeplinks/2013/12/google-removes-vital-privacy-features-android-shortly-after-adding-them/>
18. B. Razeghi, M. Hatamian, A. Naghizadeh, S. Sabeti, and G. A. Hodtani, "A novel relay selection scheme for multi-user cooperation communications using fuzzy logic," in *Proceedings of the 12th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Taiwan, 241–246 (2015)
19. S. Berenjian, M. Shajari, N. Farshid, and M. Hatamian, "Intelligent automated intrusion response system based on fuzzy decision making and risk assessment," in *Proceedings of the 8th IEEE International Conference on Intelligent Systems (IS)*, Bulgaria, 709–714 (2016)
20. P. Tavakkoli, D. M. Souran, S. Tavakkoli, M. Hatamian, A. Mehrabian, and V. E. Balas, "Classification of the liver disorders data using multi-layer adaptive neuro-fuzzy inference system," in *Proceedings of the 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, USA, 1–4 (2015)
21. G. Chen, and T. T. Pham, Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems. CRC press (2001).