Multi-View Permission Risk Notification for Smartphone System

Carol Fung[1]*, Bahman Rashidi[1], and Vivian Genaro Motti[2]
[1]Department of of Computer Sciences, Virginia Commonwealth University, USA
cfung@vcu.edu
[2]Department of Information Sciences and Technology, George Mason University, USA
vmotti@gmu.edu

Received: December 23, 2018; Accepted: March 7, 2019; Published: March 31, 2019

Abstract

The current mobile architecture design allows mobiles apps to have unprecedented access to sensitive user information. While users are concerned about privacy breaching, they may not be able to evaluate the privacy risk when downloading apps from smartphone application marketplaces. Currently, Android users only receive Android permission requests, which appear when an app attempts to access phone resources and the user can choose to grant or deny the requests. The current permission requests interface provides little information to help users understand the risk of granting those requests. In this work, we study how privacy notification interface can play an important role in assisting users in making informed decision regarding permission control. To address this issue, we propose a novel multi-view privacy notification mechanism that provides customized notification interfaces that help users obtain necessary information about the risk behind granting a permission. The implementation of our model includes a new design of User Interface (UI), interpreting apps' activities risks, and users' preferences adaption. We also propose a set of metrics to evaluate the usability of the notification system. To evaluate the usability of our mechanism, we conducted a user survey and analyzed users' feedback.

Keywords: Smartphone Application, User-computer interaction, Privacy, Usability, Android, Multi-view, Customized Interface Design

1 Introduction

Current Android smartphone operating systems allow applications to access to sensitive resources such as SMS, camera and microphone, through various permission control strategies. For example, some previous versions of Android operating system (e.g. Android 5 and earlier versions) require users to accept all permission requests from an app at installation time [1] as a prerequisite to run the app. IOS and more recent Android OSs (Marshmallow and after) request user permissions only at runtime when a resource is used by the app for the first time [2]. However, more and more malicious apps utilize this loophole and request irrelevant resource access from users [3].

Studies have shown that only a small percentage (3%) of smartphone users read Android installation-time permission requests and make decisions carefully [4]. There are several reasons that few smartphone users are making careful decisions on app permission requests. First, users tend to rush through the installation process and like to install apps immediately. Second, users are not aware of the privacy and security risks that granting certain permissions may cause private information leakage or other types of

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 10:1 (March 2019), pp. 42-57
DOI: 10.22667/JOWUA.2019.03.31.042

*Corresponding author: Computer Science Department, School Of Engineering, Virginia Commonwealth University, 401 West Main Street, Richmond, Virginia, USA, Tel: +1-(804)828-9731, Fax: +1-(804)828-2771

security concerns. Third, users lack knowledge in information technology or mobile security. Inexperienced users may not understand the permission requests or the risk of their actions (grant or deny). As a result, users are put into the positions to decide whether "grant" or "deny" permissions or not without assistance. Therefore, users may end up granting excessive permissions to apps [5]. Figure 1 shows permission requests (for GPS) on iOS, Android and Windows phones. We can see that all of them have similar simplistic notification interfaces that provide little information to assist users to make decisions. For example, an expert user may not be willing to share exact location due to privacy concern, but would like to share their coarse GPS location so that the app can function normally. Also expert users may also want to know whether the GPS information leaves their phones to a remote location or not and/or what precision of their information is leaked. However, current simplistic interface designs on smartphones would not satisfy the need from expert users in a convenient way.

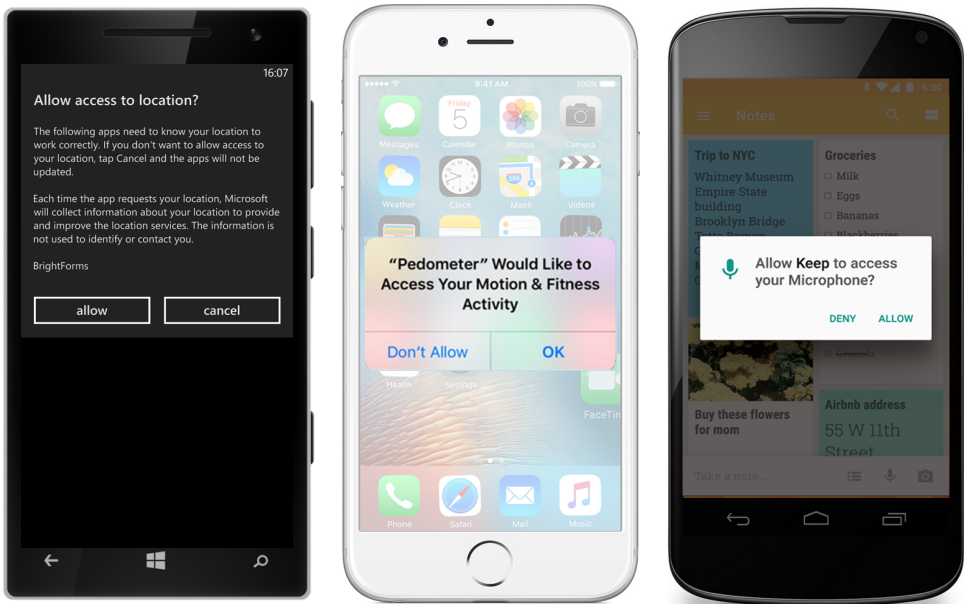


Figure 1: Current iPhone, Android and Windows phone's privacy notification

To address the above issues and improve the usability of the permission notifications system, we design customized notification interfaces for smartphone users based on their preferences and expertise levels. For example, we provide multiple notification interfaces which contains different levels of information details for users to choose from. Therefore, inexperienced users and expert users can have the information they want from difference notification interfaces. A customized interface can help users understand the information better and make informed decisions on permission control. To the best of our knowledge, we are the first to propose customized notification for smartphone users.

Our proposed model aims at improving the existing permission notification models through a multiview security notification model. In this model, we consider multiple important factors in the design of an effective permission notification model, which include users' knowledge, their experiences with malicious apps and their ability to understand certain information. Our model will provide multiple views of privacy risk notification interface for installed apps on devices. Each view provides a calibrated information presentation of permission requests. For instance, some of the views may target technical savvy users who have better understanding of IT, security, and mobile device privacy risks. Some other views may target inexperienced users with little understanding about IT (users who are not familiar with privacy risks and technical terms). Users then can choose the view they feel more comfortable with that help them understand better about the risk of granting a resource through the information it provides.

Users can also switch to different views at any point of time after their initial choice. This provides users with customized information as well as flexibility to help them make informed decisions. To evaluate the effectiveness of our design, we surveyed a group of 200 users to collect their feedback on using this interface and the analysis results is included in this paper.

The contributions of this paper can be summarized as follows: (1) We proposed a novel customized permission notification model based on users' preferences and background. (2) We developed a multi-view model that provides users with a set of views of privacy risks notification to choose from. (3) We conducted a survey involving 200 people to collect their feedback on our customizable multi-view notification system and provided an analysis report. To the best of our knowledge, our model is the first of its kind that provides multi-view features and strategies to assist smartphone users preserve their privacy.

1.1 Related Work

Some work has been done to assist users in permission control. For example, Fung et al. proposed a permission recommendation system [6, 7, 8, 9, 10] to provide access information to users regarding the likelihood that certain apps or permissions are malicious. Several other researchers have looked at the problem of Android permission management. In ProtectMyPrivacy (PMP) [11], the authors proposed to provide fine-grained control of permissions by adding recommendations (turn on or off) on each requested permission. On the aspect of human computer interaction, more effort has been done to provide improved interface to smartphone users. Studies [12] have shown the importance of making informed decisions by the users. For example, in [13], the authors propose to provide with runtime dialogs to inform users why a permission is considered necessary or not so that users can make informed decisions based on the additional information. The authors of [14] conducted a case study of the effectiveness of privacy notice to users with various lengths and wordings and found that short notices are more effective than long ones. However, none of the above work has taken the differences on the level of understanding among difference users and tailor interfaces for users. Therefore, their solutions are not able to provide customized decisions for everyone. In our work we will address the differences among users and propose to provide customized notification interfaces to users.

1.2 Paper Structure

The rest of the paper is organized as follows: Section 2 elaborates the details of our model and primary factors considered in the model. Section 3 presents our user study and our usability experiments. Section 4 discusses some of the challenges facing our model. Finally, Section 5 concludes the model.

2 Multi-view Model

The design of a permission notification should consider two factors. First, what content to put in the notification so that users can understand? This is a complex task because users come from various backgrounds and have different level of capability of understanding technical content. Second, the interface design of a permission notification is another important aspect of the design process. The user interface (UI) has a high impact on the success of delivering the actual privacy risk of permissions. Therefore, it is vital to consider these factors in designing any security and privacy model.

2.1 Multi-view Design

The knowledge level of a user has a direct impact on the user's understanding of a notification. For example, an inexperienced users may not realize that a Flash Light app should not request the location information of a user, while it may be apparent to expert users. Therefore, showing detailed GPS access information of an app is only helpful to experienced users. To assist users, we design multiple interfaces containing information about the requested permissions from apps for users of different knowledge levels. Each interface is called a *view* of information. Each view has three properties defined as follows:

Granularity: We define the granularity to be the format, scale and quantity of information we put into a notification. For example, the actual activity logs of an app are considered as fine grained and showing the overall risk of an app is considered as coarse grained.

Intricacy: Intricacy refers to the complexity of the information in terms of understanding, technical level and also interpretation by users. For example, the actual activity logs is considered as high intricacy (because it needs knowledge of app API and system calls) and overall risk level (the risk is presented in common spoken language) is considered as low intricacy.

Co-equality: Co-equality refers to the consistency of the information we put into the notification. For example, a skilled user's interpretation from the actual logs of apps should be the same or close to a user with low expertise from the overall risk of apps.

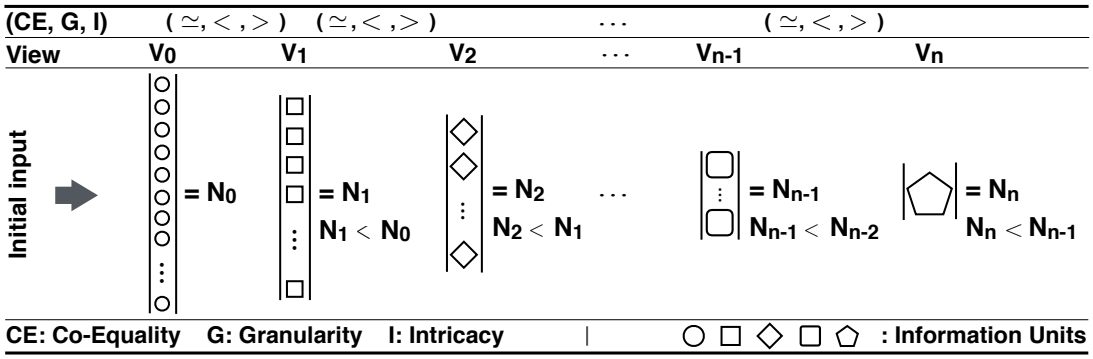


Figure 2: The process of generating multiple views of privacy risks to be presented to users

Figure 2 shows an abstract presentation of the set of views we create in our model, denoted by $\{N_0,...,N_n\}$. Multiple views have different characteristics in terms of granularity, co-equality and intricacy, denoted by (CE,G,I). The granularity and level of intercity of the views decreases from left to right. All views satisfy the co-equality property.

2.2 Permission Notification Interfaces

In this subsection we provide sample designs of the multi-view interfaces for users of different knowledge levels that provide information about the resource usage of an app. Users can select their preferred view that they feel most comfortable with. The designs are shown in Fig. 3. We can see that all views are different from each other in terms of granularity and intercity. The explanations of these views are as follows:

- *View 1*: This interface provides the most detailed information regarding an applications' behaviors and activities (Privacy related log data is shown). This view also has the most detailed statistic information.

- *View 2*: This interface does some risk assessment process on the information provided at level 0. The red numbers are assessed risks of the app. The red lines are suspicious activities that the app have had during runtime. This view has reduced amount of statistic information compared to view 1.

- *View 3*: This interface shows the assessed risk level of each requested permission (resource) by the application. The assessed risk is course-grained to 5 levels.

- *View 4*: This interface shows the overall risk level of the app and possible misused resources.

- *View 5*: This view suggests that an app is malicious or not.

2.2.1 User Preference:

The interaction portal (interfaces and setting) is to facilitate the interaction between users and devices. Through this component, we enable users to manage the privacy risk of permission notification of their devices. They can change the configuration and select their preferred views to monitor the risk of installed apps.

In our model we designed the system in way that users can choose their preferred views manually or automatically in which users leave it up to the system. The system shows privacy risks of the system and requested permissions to users through the selected views. If a user does not select a view, system selects one as a default view to be used. In our design, users are able to select views per app. Fig. 4(b) shows the portal that users can see what views are being selected for what apps. Users are able to change it at any-time.

Users will be informed about the privacy risks in three different ways. First, they can find the information at the device's setting. Second, they can receive the information at runtime using pop-ups. Third, they can see the information at the installation time. In the third way, the information that users will see is calculated from other users in the community (those who have previously installed and used the app).

2.2.2 Action Recommendation:

In general, following the permission notifications, users are asked to take an action on apps. In the existing solutions, there are only two options "allow" or "deny". However depending on the activities and behaviours of apps, these set of actions may not work. For example, if an app is malicious and has access to a set of sensitive resources, denying a single permission may not block the privacy and security risks. The proper actions in such case would be uninstalling the app to avoid this scenario.

In addition to the multi-interface model, we also provide users with a set of actions they can take. The actions vary depending on the risk of apps. For example, if the risk of app is high, the actions we recommend users to take are different from an app which is low risk. For each notification that our model generates, we also recommend a set of actions. Some actions are more strict than others. Figure 5 shows a set of actions buttons we recommend to users. For example, Figure 5(a) is recommended when apps are low risk and it is not necessary to recommend more strict actions. Figures 5(c)(d) recommend more strict actions including "uninstall". In the case of uninstall, we also provide users with a detailed explanation to inform them about the reasons behind the action we recommend.

Through action recommendation, we not only show users the risk behind apps, but also a set of proper actions that users can take.

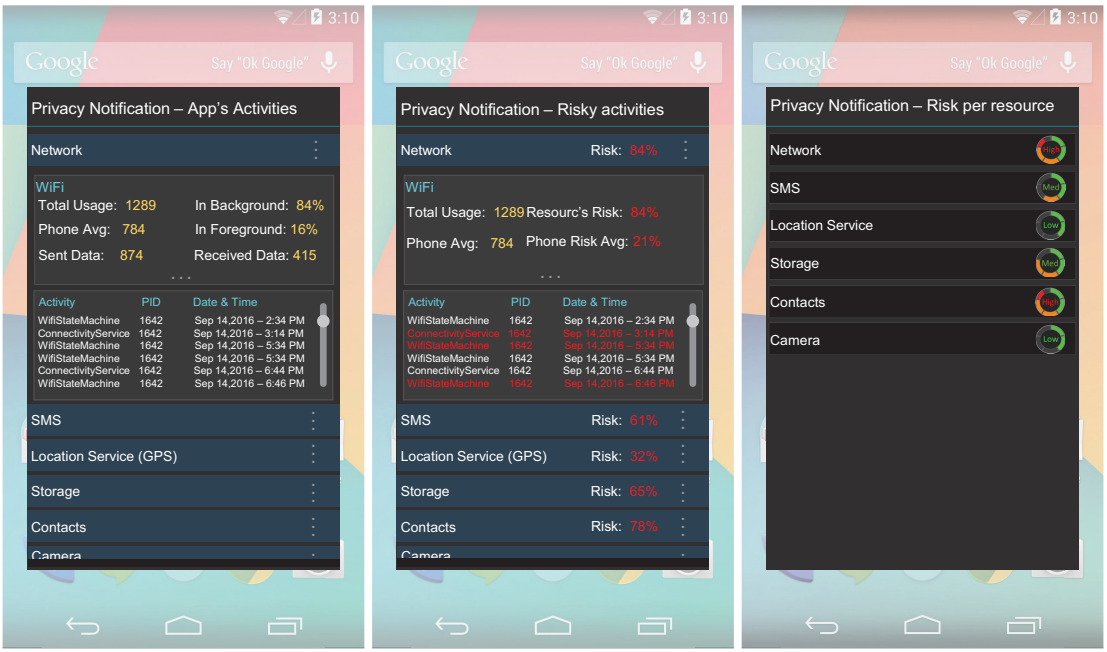
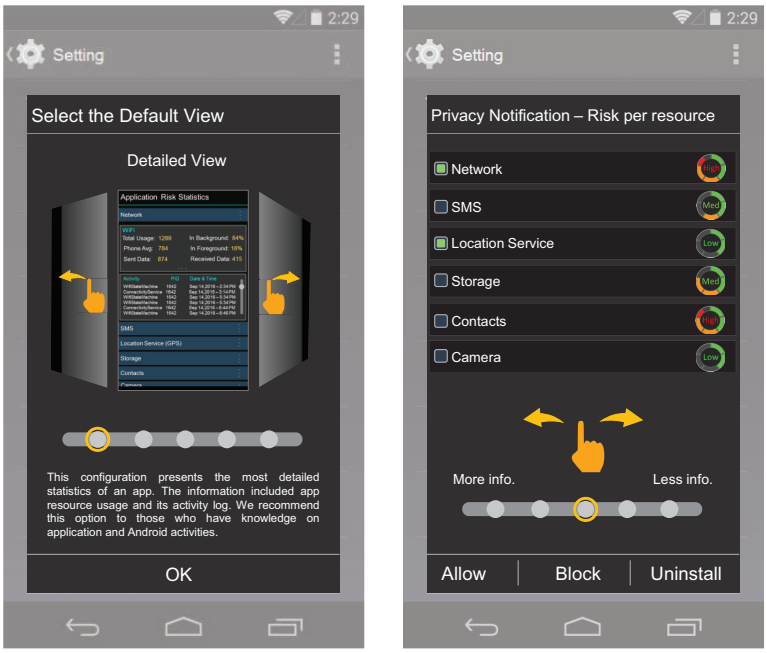


Figure 3: User Interfaces of 5 the views: (a) shows View 1 with highest of level intricacy; (b) shows View 2 which is similar to view 0 but it includes some assessed risks; (c) illustrates View 3 with the assessed risks for every requested resource; (d) shows View 4 that includes an overall risk of the app; (e) shows View 5 shows that an app is malicious or not



(a) (b)

Figure 4: Mode selection for privacy risk views (a) illustrates user interface of selecting a mode to see the views; (b) shows the interface of selecting views for both modes; (c) shows the interface of list of apps and their views.

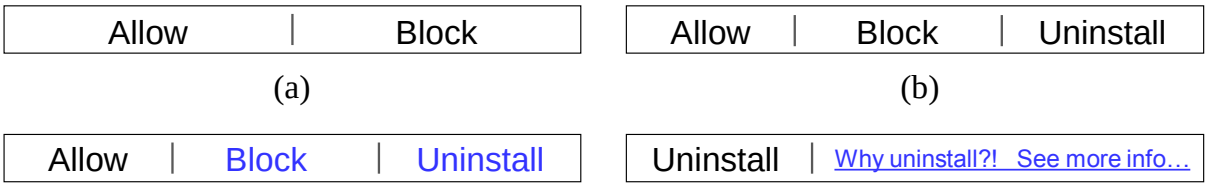


Figure 5: Designed buttons for the views.

3 User Feedback Study

To evaluate the effectiveness and usability of the multi-view risk notification design, we conducted a survey to collect feedback from users. In this section we discuss the results of the survey that is used to evaluate the usability of the notification model. We will answer the following questions in our user study; "*Is there a need/interest for a privacy/security notification system showing different views with different amount of information about the privacy/security risk of apps/permissions?"* and also if the answer to the first question is yes, then "*Is the interest distributed among all views or only one or two specific views?*".

3.1 Study Setup

In order to answer the questions mentioned in this section, we designed a survey to collect users feedback on the proposed model. Our user survey study process includes three phases: "Lab study", "Training" and "Feedback collection".

Lab study: before we publish the survey on Amazon MTurk we conducted pre-study in our lab to discover and resolve any ambiguity with the survey questions. That way the survey is easy to read and

understand and participants are less likely to have any issue to understand the questions of our survey. During this step, we made a set of corrections and redesigned the survey.

Training: before asking users' opinions, we decided to give necessary information to participants, so they understand the concept and technical terms in our survey. We explained the concept of permission notification to participants together with explanation of the single view (existing models) and also our model (multi-view). To avoid making any bias in participants, we designed the survey to be neutral and only focuse on the facts about the models.

Feedback collection: in order to collect feedback from users we used Amazon MTurk platform to publish our survey. We set the language and location of participants to be English speaking countries and English respectively. The reason behind this was that the content of the survey was in English. This helped us to have a successful training process. Figure 6 shows the targeted locations to launch our user study. We set the locations to be US, UK, Canada, Australia and New Zealand. In order to conduct the survey, we have obtained an IRB permission from the Virginia Commonwealth University under the license number IRB HM20003840_CR2. No identifiable information was collected in the survey.

In our survey we collected some basic unidentifiable demographic data. Table 1 shows the education level of participants. We set the education to "High-School and under", "Undergraduate" and "Graduate and above". Table 2 shows the demographic information related to the age of participants. Finally, Table 3 shows the distribution of gender among our participants.

In order to avoid making any bias in our survey, we placed the demographic information collection to the end of survey. This way, participants are not influenced by the answers they provide to the demographic section.

Table 1: Diversity of Participants (Education level)

Educational Level	High-School	Undergrad	Graduate and above
Number of participants	41	104	55

Table 2: Diversity of Participants (Age)

Age	20-30	30-40	40-50	50-60
Number of participants	53	82	41	24

Table 3: Diversity of Participants (Gender)

Gender	Female	Male	Others
Number of partiipants	90	107	3

3.2 Users' Model Preference

In this section, we present users' feedbacks on the single-view and multi-view models. Before we collect their feedbacks, we presented them a neutral explanation of the functionalities and features of each model. In this step, we only focused on the facts and features of the models. We designed this section in a way that users will navigate through the views. For each view, we also added an explanation, so that participants understand the features of each view.

Figure 7 shows the preferences of participants. As we can see, 170 of participants in our survey preferred the multi-view model and only 30 participants prefer the single-view model. This shows that vast majority of participants prefer to use the multi-view model.

Figure 6: Location of participant of our user study.

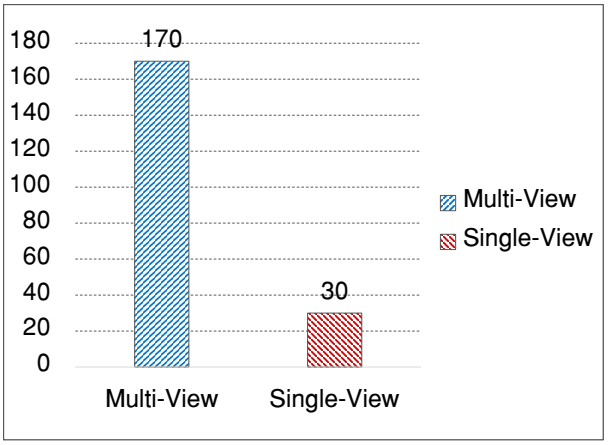


Figure 7: Participants model preferences in terms of single and multiple views.

In order to confirm the fact the participants have a clear understanding of both models, we decided to ask their reasons on choosing the models as a requirement. This way, we can make sure that they understand what they choose. We received 25 words per sentence in average from participants, which is sufficiently long to understand/interpret their reasons. We used "voyant-tools" word processor [15] to conduct an analysis on the inputs from participants. Table 4 shows the most common phrases among the participants feedbacks for both models. As highlighted in the table, we can see that participants have a clear understanding of both models. The core reason behind choosing single-view is "*simplicity*" and "*less information*". In contrast, those who chose multi-view mention "*more features*", "*interest in more information*" and also ""*ability to choose*". From these feedbacks, we can conclude that participant had a good understanding of models before they chose. They mainly pointed out the actual reasons we set to behind each model.

In order to see the correlation among some of the factors in our survey and the model preference, we measured the preference of participants with respect to "*malicious app experience*" and "*security concern*" factors. Figure 8(a) shows the model preference among users who have experienced mobile malware apps. In order to make sure that participants have a good understanding of what a malware is,

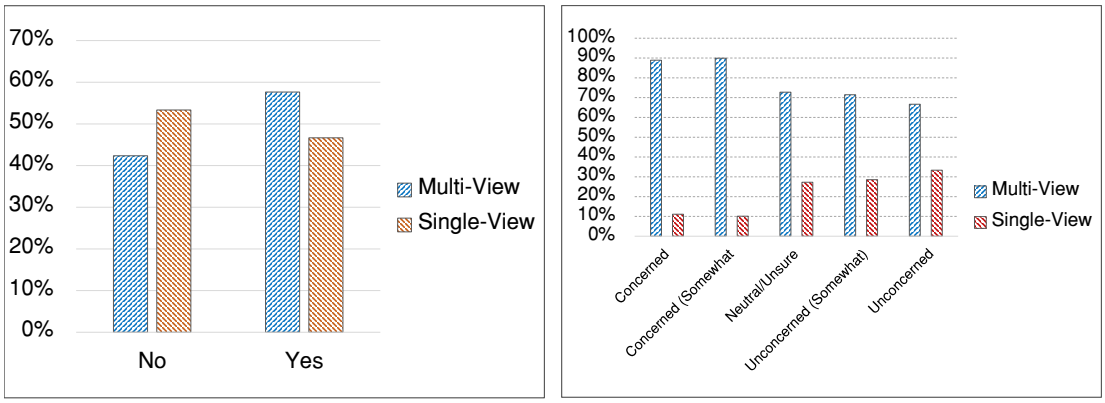
Table 4: Users feedback on their preferred model

Model	Single-View	Multi-View
Phrases	'easier to use'	'a lot more information'
	'less information '	'risks of applications'
	'simple'	**'has more features'**
	'less details'	'seems more useful'
	'simplistic'	**'i would like to know more'**
	'easy to understand'	'more options'
		'I can choose'

we explained this term during the training process. As we can see, those who have experienced malware, prefer the multi-view model over the single-view. The reason behind this may be the fact that they want to know more about the permission risks of apps. Additionally, those who have experienced malware on their devices, are already familiar with the concept of malware and they have a better understanding of malicious apps. We also noticed this fact from participants feedback in Table 4 (understanding of malicious activity and risks).

As security concern was one of the items we collected participants' answers, we decided to see the relation between the model preference with respect to the security concern of participants. Figure 8(b) shows the model preference of participants with considering the security concern. As we can see, participants with higher level of concern are more interested in using the multi-view model.

The feedback from participants answers the question "*Is there a need/interest for a privacy/security notification system showing different views with different amount of information about the privacy/security risk of apps/permissions?*". 85% of participants preferred the multi-view model which is designed to give users more insight about their apps. In addition, users with different backgrounds and security concerns, chose our proposed model over the single-view model. Therefore, the answer to this question is that users are in need of such system and also they have the understanding of what they need.



(a) (b)

Figure 8: Participants preferences in terms of single and multiple views: (a) Participants model preference by experience of malware; (b) Participants model preference by security concern.

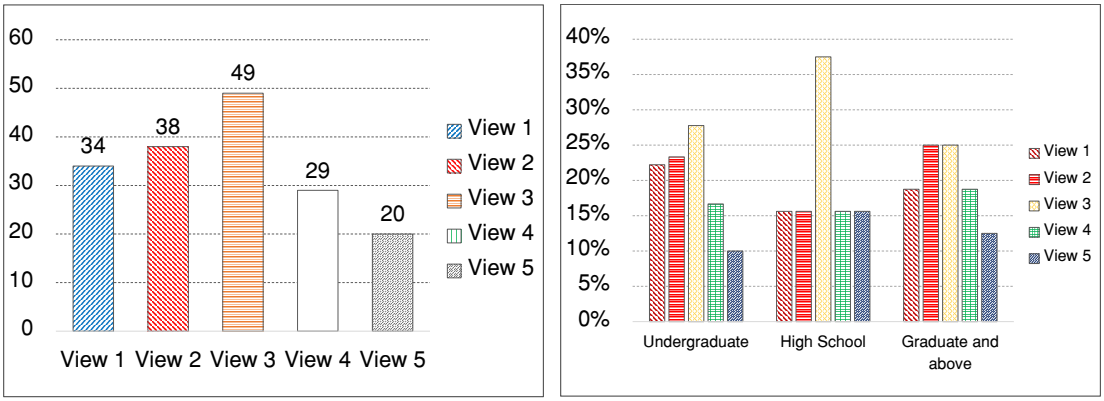
3.3 View Preference

In this section, we analyze the feedback from participants who prefer to use the multi-view model. We wanted to see if the view preferences are being distributed among different views. The distribution of preference among views shows the need for a model with different views containing different types of

Table 5: Users feedback on their preferred view of the multi-view model

View	View 1	View 2	View 3	View 4	View 5
Phrases	'most information' **'detailed information'** 'most detailed'	**'has risk assessment'** 'detailed information' **'malicious activity'**	**'multiple risks'** 'assessed risk' 'not much details' 'like the 5 level' **'visually appealing'** 'easy to understand'	'summarized' 'easy to understand' **'overall risk'** **'summary of risk'** 'informative' 'shows misused resources'	**'simple'** 'quick info.' 'less confusing'

information about the privacy risks of permission. Those participants who chose the multi-view model as their preferred model, were later ask to choose the view they prefer the most among all views. They were presented with details for each view including the features, purpose of the view, and the type of information included in the view. Figure 9(a) shows the distribution of participants' preferences among the views. As we can see, each view is being selected by a number of participants. This answers the questions of "*Is the interest distributed among all views or only one or two specific views?*". View 3 is the most preferred view among all views. We believe the reason behind this is the fact that this view provides a moderate amount of risk information and at the same time it is visually appealing. We can see that 34 and 38 participants are interested in view 1 and 2 respectively, which is surprisingly higher than our expectations.



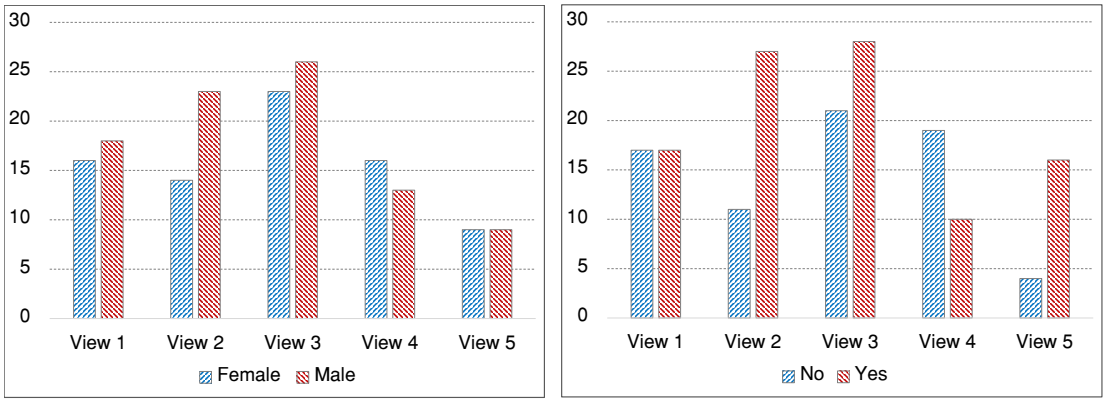
(a) (b)

Figure 9: Participants preferences in terms of single and multiple views: (a) Participants model preference by experience of malware; (b) Participans model preference by security concern.

We also calculated the correlation between view preference, gender and malware experience. Figure 10(a) shows the correlation between the gender factor and view preference. The correlation between female and male participants for all views is 0.71. As we can see, except for the View 2, both gender categories follow the same pattern. That is the reason behind the 0.71 correlation. It is worth mentioning the a correlation of 1 means that the two variables/factors always follow the same pattern. Figure 10(b) shows the correlation between malware experience and view preference. The calculated correlation value for this factor was 0.07, which is very small. This shows that there is almost no correlation between experiencing malware and views that participants have selected. View 3 is the only view that both categories follow the same pattern.

As we did for the model preference, we required participants to leave a reason for the view they prefer. This way, we can make sure that they understand views they chose. We received 27 words per sentence in average from participants, which is sufficiently long to understand/interpret their reasons. Table 5

shows the processed feedbacks from participants. The core phrases from participants are highlighted in the table. As we can see, participants have correct understanding of the core purposed behind each view.



(a) (b)

Figure 10: Correlation between gender, malware experience, and view preference: (a) Correlation between gender and view preference; (b) Correlation between malware experience and view preference.

4 Discussion

In this section we discuss some of the challenges related to the model. We also propose our solutions for the challenges. As we previously mentioned, consistency among the designed views is a key aspect. Users with different levels of knowledge should be able to understand/interpret the views the same. In addition, as we mentioned, recommending actions to users for each view is another challenge.

4.1 Consistency

Consistent views can help users to make correct decisions regardless of their background and knowledge. To be able to achieve such consistency, there should be a control system to evaluate the consistency of the views. Our proposed solution to address this challenges it to equip the model with *Control Theory* to be able to tune the views and enhance the consistency of views.

The main objective of control theory is to control a system. It helps a system to control the input-output flow and also make sure that the system's output follows a desired value. Control theory controls the input flow of a system and influences the behaviour of dynamic systems [16]. On the other words, it models a (non)physical system, using mathematical modeling, in terms of inputs, outputs and various components with different behaviours, use control systems design tools to develop controllers for those systems and implement controllers in physical systems employing available technology. This way, we can achieve a certain level of control over the behaviour of systems and manipulate them to the system behave in a desirable manner [17].

In order to be able to apply control theory to our model, we made modification in the architecture of a control theory unit. Fig. 12 illustrates the basic architecture of an adapted control theory system. We made changes in the processing unit of a control theory unit. The reason we made this change is that we needed to evaluate the quality of view before we generate it. That is why we made a short-cut in the model and until the qualification are not met our model does not generate the views. The components are explained as follows:

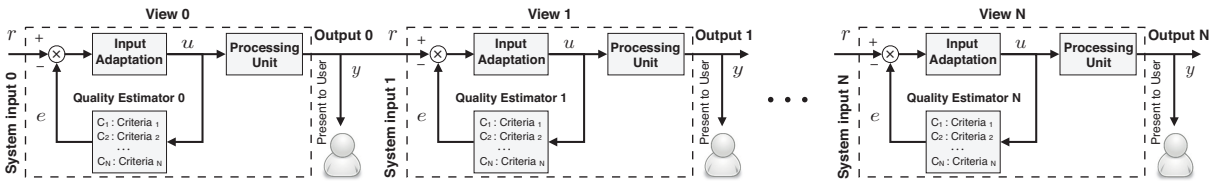


Figure 11: A chain of closed-loop control systems.

Processing Unit: A plant of a control theory system is the part of the system to be controlled. On the other words, a part of system that is responsible for generating the output is usually referred to as the plant. The usual objective of control theory is to control plant, so its output follows a desired value, called the reference (r), which may be a fixed or changing value.

Input Quality Estimator: The controller (compensator or simply filter) determines the setting of the control input needed to achieve the reference input. The controller computes values of the control input based on current and past values of control error. On the other words, The controller provides satisfactory characteristics for the total system.

Input Adaptation: The system for measurement of a variable (or signal) is called a sensor. The sensor transforms the measured output so that it can be compared with the reference input (e.g., smoothing stochastic of the output). It monitors the output and compares it with the reference. The difference between actual and desired output, called the error (e) signal, is applied as feedback to the input of the system, to bring the actual output closer to the reference.

Figure 12 shows an adapted version of a control theory unit. In this unit, the input is risk information we need to generate a view. The information can be the raw logs of apps or an evaluated risk. Depending on the design of a view, the units can change. The change can be the format of input required to generate the view and also the format of output for the view. The controller component of each unit, measures the qualification of the view. It measures the quality of input and makes decision whether the input and output requirements are being met or not.

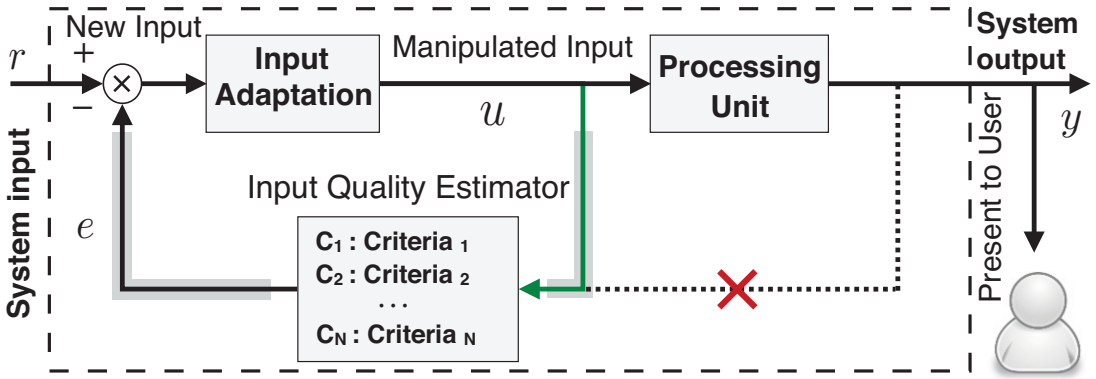


Figure 12: A adapted control theory unit.

Because there are multiple views in out proposed model, we need a chain of nodes to be able to generate all the views. Figure 11 shows a chain of units. Except the first unit, the output of the previous unit is the input for the next unit. Using such model, we can make sure that each view is being controlled in terms required qualifications. The controller can be customized depending on the policies and rules we set. As we can see, there should be a set of criteria to meet. It is worth mentioning that to be able to evaluate the consistency, we also need to have an evaluation process after applying such model. The evaluation process can be in the shape of user survey and collecting users feedback to evaluate the fact

that they have the same understanding of the views.

4.2 Action Recommendation

As recommending actions (buttons) in our views and interfaces is a key part of the model, there should be a mechanism on which actions should be selected and offered to users. For such mechanism, we mainly rely on the actual risk of apps. A threshold-based system can be a solution to such challenge. Defining a set of thresholds and mapping each risk threshold to a set of actions is the solution. For example, for an app with high risk activities, we can define a set of "Block" and "Uninstall" and for apps with low risk we offer milder actions such as "Block" and "Deny".

5 Conclusion

In this paper we proposed a model for generating multiple interface notifications. Our model considers users' background and knowledge levels and generates views with different levels of intricacy and granularity. Using this model, users have the option to choose their preferred interface and view. Users can choose the view that they can understand the most and are more comfortable with. Our user study shows that users are more interested in have a multi interface permission notification mechanism. Additionally, our study also shows that users' interests are distributed among different views. Users with different backgrounds have different preferences. Based on our study, users with higher concerns in terms of security and privacy of their smartphones and personal information, have higher interest in our multi view model. Finally, our user study showed that there is a need for such model and also users are interested in multiple views instead of one.

References

[1] F. Schaub, R. Balebako, A. L. Durity, and L. F. Cranor, "A design space for effective privacy notices," in *Proc. of the 11th Symposium On Usable Privacy and Security (SOUPS'15), Ottawa, Ontario, Canada*. USENIX, July 2015, pp. 1–17.

[2] D. Barrera, J. Clark, D. McCarney, and P. C. van Oorschot, "Understanding and improving app installation security mechanisms through empirical analysis of android," in *Proc. of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'12), Raleigh, North Carolina, USA*. ACM, October 2012, pp. 81–92.

[3] "Mcafee q3 2011 threats report shows 2011 on target to be the busiest in mobile malware history," November 2011, https://techcrunch.com/2011/11/20/mcafee-nearly-all-new-mobile-malware-in-q3-targeted-at-android-phones-up-37-percent [Online; accessed on March 25, 2019].

[4] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proc. of the 8th Symposium on Usable Privacy and Security (SOUPS'12), Pittsburgh, PA, USA*. ACM, July 2012, pp. 3:1–3:14.

[5] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang, "Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing," in *Proc. of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12), Pittsburgh, Pennsylvania, USA*. ACM, September 2012, pp. 501–510.

[6] B. Rashidi, C. Fung, and T. Vu, "Recdroid: A resource access permission control portal and recommendation service for smartphone users," in *Proc. of the 2014 ACM MobiCom Workshop on Security and Privacy in Mobile Environments (SPME'14), Maui, Hawaii, USA*. ACM, September 2014, pp. 13–18.

[7] B. Rashidi, C. Fung, and T. Vu, "Dude, ask the experts!: Android resource access permission recommendation with recdroid," in *Proc. of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (INM'15), Ottawa, Ontario, Canada*. IEEE, May 2015, pp. 296–304.

[8] B. Rashidi, C. Fung, and E. Bertino, "Android resource usage risk assessment using hidden markov model and online learning," *Computers & Security*, vol. 65, pp. 90–107, March 2017.

[9] B. Rashidi, C. Fung, A. Nguyen, T. Vu, and E. Bertino, "Android user privacy preserving through crowdsourcing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 773–787, 2018.

[10] B. Rashidi and C. Fung, "Disincentivizing malicious users in recdroid using bayesian game model," *Journal of Internet Services and Information Security (JISIS)*, vol. 5, no. 2, pp. 33–46, May 2015.

[11] Y. Agarwal and M. Hall, "Protectmyprivacy: Detecting and mitigating privacy leaks on ios devices using crowdsourcing," in *Proc. of the 11th Annual International Conference on Mobile Systems, Applications, and Services(MobiSys'13), Taipei, Taiwan*. ACM, June 2013, pp. 97–110.

[12] T. Westermann and I. Wechsung, "Empowering users to make informed permission request choices," in *Proc. of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI'15), Copenhagen, Denmark*. ACM, August 2015, pp. 1123–1125.

[13] B. Bonné, S. T. Peddinti, I. Bilogrevic, and N. Taft, "Exploring decision making with android's runtime permission dialogs using in-context surveys," in *Proc. of the 13th Symposium on Usable Privacy and Security (SOUPS'17), Santa Clara, California, USA*. USENIX, July 2017, pp. 195–210.

[14] F. Schaub, R. Balebako, and L. F. Cranor, "Designing effective privacy notices and controls," *IEEE Internet Computing*, vol. 21, no. 3, pp. 70–77, May 2017.

[15] "Voyant tools," https://voyant-tools.org/ [Online; accessed on March 25, 2019].

[16] J. Zabczyk, *Mathematical control theory: an introduction*. Springer Science & Business Media, 2009.

[17] S. P. Bhattacharyya, L. H. Keel, and A. Datta, *Linear control theory: structure, robustness, and optimization*. CRC press, January 2009.

Author Biography

Carol Fung received her PhD degree in computer science from the University of Waterloo (Canada). She is currently an assistant professor in Virginia Commonwealth University (USA). Her research area is network management and cyber security, including collaborative network and security, trust management, resource allocation, game theory, Bayesian inference theory and crowdsourcing. Her research has applications in SDN/NFV networks, 5G networks, cyber security and smartphone networks. She is the recipient of the IEEE/IFIP IM Young Professional Award in 2015, University of Waterloo Alumni Gold Medal in 2013, best paper awards three times in IM/NOMS.

Bahman Rashidi received his Ph.D. degree in computer science from Virginia Commonwealth University in 2018. His research interests include distributed systems, mobile systems and privacy issues in smartphone devices. He received the master's degree in computer engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2014. Being the top student in the program, he graduated with the Distinguished Master's Student of the year in research award for two consecutive years in 2012 and 2013. He is the recipient of the Outstanding Early-Career Student Researcher Award.

Vivian Genaro Motti is an Assistant Professor of Human-Computer Interaction (HCI) and the Director of the Human-Centric Design Lab (HCD Lab). She has extensive experience implementing and evaluating wearable technologies and conducting user studies to elicit users? needs. The ultimate goal of her research is to bridge the gap between what users need and what technology offers them. Her research interests are wearable computing, usable privacy and healthcare informatics. Her work has been funded by NSF, IARPA, TeachAccess and AARP.