Understanding Sensor Notifications on Mobile Devices

Zongheng Ma, Saeed Mirzamohammadi, Ardalan Amiri Sani

University of California, Irvine
zonghenm@uci.edu, saeed@uci.edu, ardalan@uci.edu

Abstract

Mobile devices, such as smartphones and tablets, use notifications to inform their users of events. Some security- and privacy-related events are time-sensitive: the user must be notified immediately. In this paper, we perform a user study with 40 participants to understand the properties of such time-sensitive notifications. We specifically focus on sensor notifications that notify the users when one of the sensitive sensors, such as camera, microphone, or location is being accessed. We show that none of the notification channels available on mobile devices, i.e., LED, vibration, sound, and display, can grab the user's attention in more than 24% of the time. Among them, vibration achieves the best success rates on average. Moreover, our results show that less intrusive channels, e.g., display, can achieve significantly better results if the device's physical context, i.e., ambient light intensity, is considered. Based on our findings, we suggest that display notification is the best option for camera while different vibration patterns are best options for microphone and location sensor.

CCS Concepts

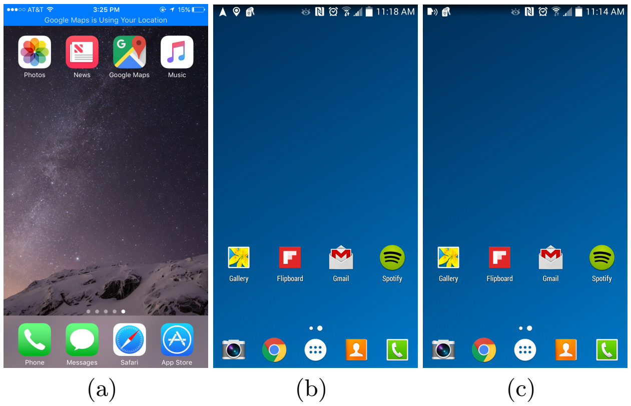
•Human-centered computing → User studies; Smartphones; •Security and privacy → Privacy protections; Social aspects of security and privacy; Usability in security and privacy;

Keywords

Sensor notifications; Mobile devices; User study

1. INTRODUCTION

Mobile devices use notifications to inform users of events, e.g., phone calls. Some security- and privacy-related notifications are *time-sensitive*: the user must be notified immediately. Examples are sensor notifications that inform the user of access to sensitive sensors (e.g., camera, microphone, and location) and AMBER alerts. In this paper, we set out to understand the properties of such notifications, i.e., *time-sensitive notifications*, through a user study. We perform the study in the context of sensor notifications but most of the results are applicable to other forms of time-sensitive notifications as well. Computers communicating with humans is one of the most fundamental aspects of computing. In this paper, we hope to understand how effectively a mobile device can grab its owner's attention using the means available to it. In an analogy to human-to-human communication, this is similar to one person seeking another person's attention in case of potential danger.



(a) (b) (c)

Figure 1: Sensor notifications in iOS and Android: (a) Location notification in iOS (the notification strip below the status bar on top of the screen, (b) location notification in Android (the icon in the top left corner), (c) microphone notification by the Samsung Voice Recorder application in Android (the icon in the top left corner).

We focus on sensor notifications due to their increasing importance. Mobile devices incorporate a set of privacy-sensitive sensors, most importantly, camera, microphone, and location (i.e., GPS or cellular network-based sensing). These sensors can capture sensitive information of the user including photos, videos, conversations, and locations. A prior study has shown that most users consider this information to be private and sensitive [9]. Unfortunately, there have been several incidents where attackers have attempted to access these sensors without user's knowledge, e.g., to capture unauthorized video or audio of the user or to track her whereabouts [2–4]. And tools are available for attackers to remotely control infected device's camera and microphone [1]. To address this problem, *sensor notifications* are used, which use some notification channel, such as LED and

HotMobile '17, February 21 - 22, 2017, Sonoma, CA, USA

ISBN 978-1-4503-4907-9/17/02. . . $15.00

DOI: http://dx.doi.org/10.1145/3032970.3032978

display, to notify the user whenever one of these sensors is being used. For example, mobile operating systems, such as Android and iOS, show some form of notification on the screen when a location sensor is accessed (Figure 1 (a) and (b)). Moreover, some applications inform the user when they record audio in the background (Figure 1 (c)).

The rationale for sensor notifications is that users can distinguish between legitimate and illegitimate accesses to these sensors. For example, if the user is using a photography application, s/he expects the camera to be on. But when the user is watching a video, s/he expects the camera to be off. Therefore, a notification about the use of camera in the latter scenario is likely to indicate malicious activity.

We identify three important requirements for a sensor notification. First, it must be able to attract the user's attention successfully at all times and as fast as possible. This is important as it will minimize the window of vulnerability, i.e., the period of time that malware can access the sensor without user's knowledge. Second, it must be unambiguous. The user must be able to quickly understand the meaning of the notification and differentiate it from other sensor notifications and other notifications in the system, e.g., phone calls or application updates. Third, it should not cause annoyance to the user. This last property is indeed important because notifications are triggered whenever the sensor is accessed, even if the access is legitimate, e.g., user taking a photo with the camera.

We ask: *which one of existing notification channels, i.e., LED, vibration, sound, and display, best satisfy these requirements?* We answer this question with a user study performed on 40 participants[1]. To perform the user study, we first designed and implemented various types of sensor notifications for camera, microphone, and location, in the Android operating system (§2). We then deployed our modified operating system on Google Nexus 5X smartphones and distributed them to the participants to use as their primary device for one week. In addition, we installed an application on the smartphone that emulates malware by using these sensors in the background a couple of times a day, which would result in a notification to the user by the operating system. We then asked the participants to log the sensor that was illegitimately used as soon as they spotted the corresponding notification. This allows us to assess the first two requirements mentioned earlier. To assess the third requirement, we asked the participants to fill out a questionnaire right after the study. We have made the data collected in the study publicly available[2].

To better understand the notifications, we decided to evaluate the impact of the device's physical context, i.e., ambient light intensity and ambient noise, on the effectiveness of the notifications. To do this, we collected these physical context measurements whenever a notification was triggered.

We present several important findings about sensor notifications. First, on average, none of the existing channels achieve a success rate (in capturing user's attention) higher than 24%. Second, vibration achieves the best success rates at 24%, and LED achieves the worst at 4%. Third, we find that sound is almost always a bad choice as it incurs significant annoyance to the user, while being outperformed by vibration and even display notification. Fourth, we find that, quite counterintuitively, existing android notifications that only rely on textual content to convey their meaning are the hardest for users to recognize in a timely manner. Finally, we realize that physical context has important impact on the effectiveness of notifications. For example, we find that our new display notification (which turns on the display and shows a strip on top of the screen with textual content in it) can be very effective in light environments. Based on our findings, we suggest to use this display notification for camera since it is effective in light environments, which is where malicious access to the camera can be effective. For microphone and location sensor, we suggest to use different vibration patterns since vibration achieves high success rate and its different patterns are easy for users to recognize.

2. SENSOR NOTIFICATION DESIGN

We designed and implemented the notifications using Android services (Android version 6.0.1, CyanogenMod version 13.0). The implementation of a sensor notification has two components: (*i*) detecting when a sensor is being accessed, and (*ii*) triggering a notification using one of the channels.

Detecting the Sensor Access. We implement the notifications for three sensors: microphone, camera, and location. We detect the access to the first two in the `MediaService`, which is part of the stack that implements the API used by applications to use these two sensors. For location, we use the `LocationManagerService`, which implements the API for applications to acquire location information.

With this implementation, we can reliably detect most application's use of these three sensors, except for two cases: First, we cannot detect the use of microphone for cellular phone calls since the modem does not use the microphone through the aforementioned API. Second, we cannot detect the use of Google Play location API, which some apps use (instead of `LocationManagerService` API) to acquire location information. In the study, since malicious accesses to sensors is emulated by an application designed by us (§3.1), this limitation does not have major effect on most of the results. It might only limit the extent to which we can assess user's annoyance with notifications when apps legitimately use a sensor. To mitigate this, we asked them to mainly use Skype for voice calls (rather than phone calls) and use Waze for navigation (rather than Google maps), both of which use standard audio and location API and hence are detected by our implementation.

Triggering Notifications. Once we detect that a sensor is being accessed, we trigger the notification for it using one of the five different notification channels supported in our prototype. The channels are as follows: blinking white LED (0.67 Hz blinking frequency with 67% duty cycle), vibration (1 second constant vibration), sound (3 second audio segment), and two notifications based on the display: one designed by us, which turns on the display (if off) and shows a strip on top of the screen with a text that mentions the name of the sensor being used (similar to iOS location notification in Figure 1 (a)), and one using existing notification supported by Android (which shows up in the pull-down notification bar, and which does not turn on the display if it is off). Thereafter, we refer to these last two notification channels as display and Android notifications, respectively.

It is important to note that the LED, display, and Android notifications stay on as long as the sensor is being used. On the other hand, vibration and sound notifications only run

[1] User study approved by UC Irvine's Office of Research, Human Research Protection, under IRB HS# 2016-3100.

[2] http://www.ics.uci.edu/~ardalan/notifdata.html

for a fixed period of time (i.e., 1 and 3 seconds, respectively), in order not to significantly disturb the user. One might wonder why we do not leave the first group of notifications (LED, display, and Android) on indefinitely until it is noticed and cleared by the user, very similar to what many applications' notifications do. This is because our goal is to assess the effectiveness of time-sensitive notifications, where it is critical for the user to be notified when the event is happening. Leaving the notification on even after the event will increase the odds of user's noticing the notification, but not necessarily in a timely manner. Therefore, to control this factor in the study, we clear the notification once the sensor is not accessed anymore.

3. USER STUDY

In this section, we describe the user study that we conducted to evaluate sensor notifications.

3.1 Recruitment and Study Logistics

We recruited 40 participants. Each participant was given a Nexus 5X smartphone with our instrumented Android operating system and was asked to use the smartphone for one whole week. 33 of participants were undergraduate students and 7 of them were graduate students, mostly majoring in computer science. Moreover, 14 of them were female. Our participants hence represent a more technically-advanced portion of the population. We compensated each participant with a $20 gift card.

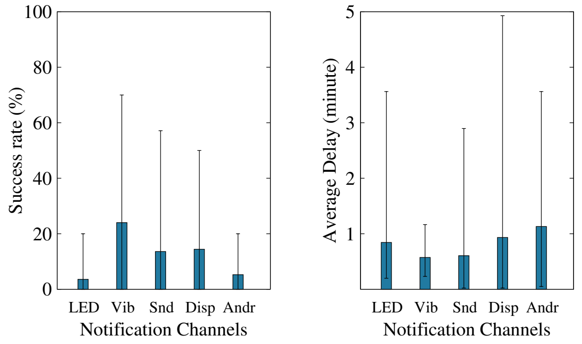
We asked the participants to use our smartphone as their main smartphone during the study. To facilitate this, we helped them set up their Google accounts on the smartphone and transfer their SIM cards from their own phones to the study smartphone, if possible. Due to SIM incompatibility, we managed to successfully perform the SIM card transfer only for about 10 of the participants. The other 30 still used our phone as their primary one but carried their own phone just for phone calls. We also asked the participants to regularly use apps that use camera, microphone, and location, in order to be able to provide us with feedback on the annoyance of the notifications. We installed Skype, Waze, and a camera application on the smartphones to facilitate the use of these applications for the participants.

In addition, we developed and installed an application, called `NotifTest`, on the phone, which plays two roles in the study. First, it emulates malware by accessing the sensors illegitimately in the background. To do this, the app uses these sensors in the background roughly 5 times a day at random times between 10 A.M. and 9 P.M. We limited the use of sensors to this time interval in order not to disturb the participants at night time. Each time, the app uses the sensor for 5 seconds. We chose this number since it is long enough for malware to gain useful information from the sensors and also it is short enough to challenge the notification mechanism in quickly grabbing the user's attention. At recruit time, we instructed the participants about the meaning of each notifications, e.g., blinking white LED notifications means that the camera is being used. Moreover, the `NotifTest` app allows the participants to check the notification for each sensor, if they forget, which helps them more easily remember the meaning of each notification. We also asked the participants to avoid lowering the system volume and disabling the vibration for extended periods of time, which would otherwise render these notifications useless. Our analysis of user study data shows that participants' phone was on high volume (higher half of the volume range) in 52% of the time and the vibration was on in 94% of the time.

Second, the app provides a mechanism for the participants to report the sensor accesses that they spot by logging it. More specifically, we asked the participants to open the `NotifTest` app and record the name of the sensor that is being illegitimately used as soon as they noticed the corresponding notification. Illegitimate use refers to the use of sensors in the background, one that the participant does not expect. Note that we assume our `NotifTest` app to be the only source of illegitimate access to sensors. We believe that this is a reasonable assumption as the operating system image on the smartphone is freshly installed; one that is not infected with malware. There are, however, some applications that legitimately use the location sensor in the background as well (e.g., navigation apps), which will result in the participants noticing some unexpected legitimate notifications, resulting in additional reports. In the analysis of the data, we can filter out these cases as there will be no records of the triggered notifications in our logs before the reports. Note that this legitimate background access to the location sensor makes its notification less effective in case of illegitimate accesses (compared to camera and microphone) since it makes it difficult for the user to distinguish. However, addressing this issue is out of the scope of this paper.

As discussed, we attempt to understand whether users can correctly associate a notification with the event that it represents. We use two mechanisms to answer this question. First, we measure the correctness of user reports of notifications. This shows whether they can associate the notification channel with its corresponding sensor. Second, for about 2 to 3 times a day, we trigger *fake notifications* that use a variation of the notification mechanism in order to assess the user's ability to differentiate it with an authentic notification. We have designed fake notifications for all of the channels. For LED, we turn on the LED with a different color, i.e., blue, but with the same blinking pattern. For vibrator, we use a different vibrating pattern, i.e., on/off vibration for 1.9 seconds at a frequency 1.4 Hz and 71% duty cycle. For sound, we play a 2 second audio segment, which is different from the authentic sound notification (i.e., different tune). For display and Android notifications, we use a different text in the notification. For display notification, we also use a different color in the strip. For Android notification, we use the same icon, relying on the user to read the text. At recruit time, we explicitly instructed the participants that they should look for the exact form of notification associated with the sensors, e.g., by paying attention to the color of LED. However, we did not tell them that our app would trigger these fake notifications in order not to affect the result of the study.

Also, as mentioned before, we are interested in understanding the effect of physical context on the effectiveness of the notifications. Therefore, right before and after each sensor access, the `NotifTest` app records some context measurements. More specifically, for 5 seconds before and after each access, it records the ambient noise level using the Android `MediaRecorder` API (i.e., `getMaxAmplitude()`) and the ambient light intensity using the light sensor. We disable the microphone's notification when microphone is used by our own system to measure the ambient noise.



(a) (b)

Figure 2: (a) Notification channels' success rate in capturing the participants' attention. (b) Participants' delay in reporting the notifications.

At the end of the study, we asked the participants fill out a questionnaire about their experience with the notifications. Specifically, we asked them about the annoyance caused to them by each notification channel.

3.2 Notification Assignment

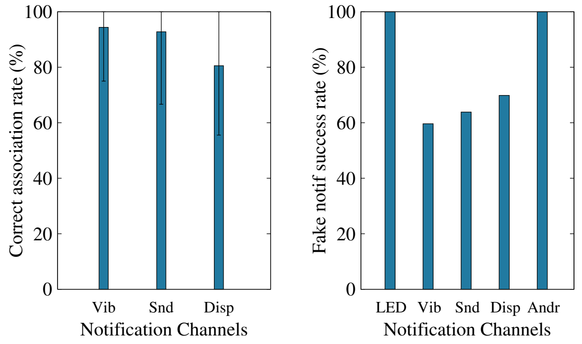
As mentioned previously, we support notifications for three sensors, i.e., camera, microphone, and location, and five notification channels, i.e., LED, vibration, sound, display, and Android notifications. Therefore, there are 15 combinations of (sensor, notification) pairs. We deploy all 15 pairs in our user study. Each user can test three of these combinations (since there are three sensors). Therefore, all combinations can be tested with five participants. As a result, since we recruited 40 participants, every combination is tested by 8 different participants.

4. RESULTS

We use the data collected in the user study to answer several questions about sensor notifications. We present every measured metric as an average (along with maximum and minimum shown using error-bars in figures) over all the participants who qualified for that metric (e.g., vibration success rate is shown as average among all users who were assigned the vibration channel for one of the sensors). Moreover, we only considered the participants who had at least 3 data points for the metric (e.g., three vibration notifications) and made sure that at least 3 participants qualify for the metric in order to have statistically reliable results.

Q1. How effective is each notification channel in attracting the user's attention?

We define effectiveness in terms of success rate and report delay. More specifically, we measure the percentage of times that the user reported a sensor usage after a notification was triggered using a specific channel. We also measure the delay of the report. Note that we put an upper bound of 5 minutes on the delay. That is, if the report comes after this upper bound, we do not consider it as a successful report. This is because, as mentioned in §2, notifications might be triggered by existing applications, and therefore user's report might be as a response to those triggers. In the recruitment, we tell the participants to report a notification as soon as possible. Therefore, we believe that this simple filtering method provides good accuracy.



(a) (b)

Figure 3: (a) The accuracy of associating notification channels with sensors. (b) The success of fake notifications in fooling the participants.

Figures 2 (a) and (b) show the success rate and report delay for different channels, respectively. We have the following findings.

Finding 1. All of the existing notification channels in mobile devices today achieve relatively low success rate on average (less than 24%). This is a significant finding as it shows that system and application designers cannot expect to be able to grab the user's attention quickly and successfully at all times.

Finding 2. Vibrator achieve the highest success rate at 24%, whereas LED achieves the lowest at 4%. This result is also significant since LED is commonly used for notifications in mobile devices these days. Our study shows that LED notifications are largely ineffective for time-sensitive notifications.

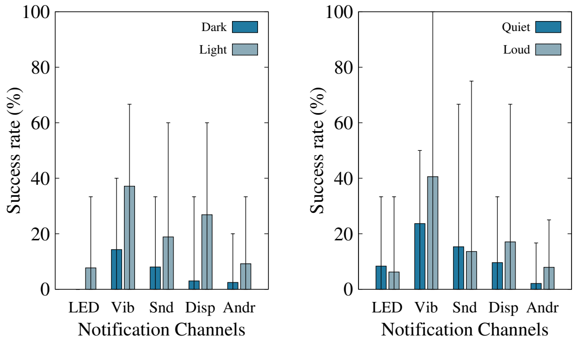
Finding 3. There is a reverse relationship between success rate and report delay. We believe there are two reasons behind this. First, notifications that are more successful in grabbing the user's attention, do so faster on average. Second, vibration and sound achieve lower report delay since they are more intrusive and cause the user to (maybe unknowingly) take action faster.

Q2. How unambiguous is each notification channel?

As mentioned before, we attempt to understand whether users can correctly associate the notification with the event that it represents. We answer this question using two metrics. First, we measure the correctness of user reports of notifications. That is, for notifications that successfully grab the user's attention, we measure how often the user reports the accessed sensors correctly. Second, we measure the percentage of times that the user falls for fake (i.e., variations of) notifications (§3.1).

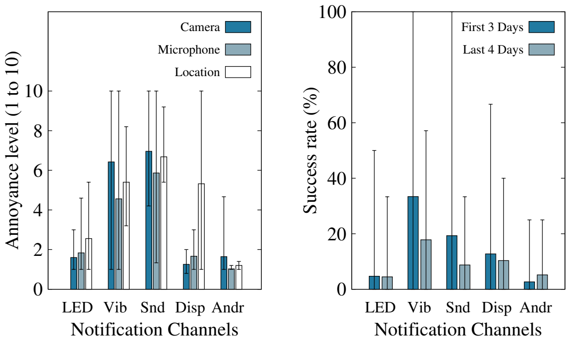
Figures 3 (a) and (b) show these metrics, respectively. Note that, in both figures, we only show the channels for which we have enough data to make a statistically reliable conclusion (§4). The missing results for LED and Android notifications in Figure 3 (a) is because we do not have at least 3 users, each with at least three successful reports for these channels.

It is important to understand how we calculate the success rate of fake notifications. One can simply report the percentage of time that the user reports a sensor access after a fake notification is triggered. However, this number would underestimate the success rate of fake notifications because it is not clear whether the user did not report the



(a) (b)

Figure 4: Effects of (a) ambient light and (b) ambient noise on sensor notifications.



(a) (b)

Figure 5: (a) User's annoyance. (b) Notification's effectiveness over time.

fake notification due to missing the notification or due to understanding that it is not the right notification. There is no way for us to directly measure this; therefore we use an approximate approach. We use the success rate of the corresponding notification channel from Figure 2 (a) to adjust the results for fake notifications. In this approximation, we assume that the success rate of a notification (Figure 2) is almost the same as the success rate of its variation (i.e., fake notification) in capturing the user's attention. With this assumption, the percentage of fake notifications that caused a report (out of all the triggered ones) divided by the success rate of the corresponding channel gives us the percentage of fake notifications that succeed in fooling the participant out of those fake notifications that s/he notices. Due to approximation, this approach can result in success rate of more than 100%, in which case, we have capped the number to 100% in the Figure (the actual value of both capped bars is about 200%). We have two important findings.

Finding 4. Users can correctly determine the notification channel used for each sensor. This is either because they remember the meaning of the notification or they check it if they forget (§3.1). Indeed, we found that on average participants did check the meaning of each notification channel between 3.5 to 6.5 times in the span of one week study.

Finding 5. Fake notifications can fool the users in many instances. More surprising, fake LED, display, Android notifications have the highest success rate. For LED, this can be due to difficulty of distinguishing the color (given the small LED size). For display and Android notifications, this can be due to the fact that these two notification channels mainly rely on textual information to convey their meaning (in contrast to sound and vibration that leverage different patterns and audios). Given that notifications only last for 5 seconds (while sensor is being accessed), textual content end up being harder to recognize in a timely manner. Indeed, in the final questionnaire, one of the participants noted that:

"notification bar/pop-up window can be hard to remember because sometime I'll forget the content of notification".

Another interesting observation is that display notification is less ambiguous than the Android notification. We believe that this is because fake display notification uses a different color strip on the screen that can be easily detected by the user (without the user having to read the text on it), whereas the Android notification uses the same icon, relying on the user to pull-down the menu and read the text. This implies that if Android notifications are to be used, it is best to use different icons and not merely rely on textual content.

Q3. Do notifications' effectiveness depend on system's physical context?

In the study, we record two contextual data whenever the `NotifTest` triggers a notification, namely, the ambient light and ambient noise. We categorize each context into two groups: light and dark, and loud and quiet. We then investigate whether the notification's effectiveness depend on these factors. We choose 5 lux as the boundary between dark and light environments since it represents the illuminance at a moderately dark environment. Unfortunately, lack of proper documentation on Android's API did not allow us to convert the ambient noise readings to decibel. Therefore, for the boundary between loud and quiet, we performed experiments and chose one that represents a modestly quiet environment in an office environment. Figures 4 (a) and (b) show the results. We have two important findings.

Finding 6. Ambient light is an important indicator for the success of notifications. Most importantly, in light environments, LED, display, and Android notifications have much better success rate. Indeed, the display notification surpasses the sound in these environments. Given that speaker has high annoyance level (see Q4 below), the system must use the display notification in light environments. Vibration remains the best options in dark environments, which might represent the device being in the user's pocket. Note that high ambient light might mean that the device is either being used by the user or sitting somewhere, e.g., on a desk. We do not differentiate between these cases.

Finding 7. Against our intuition, even in quiet environments, vibrator achieves a better success rate than sound. This finding shows that sound channel can be safely replaced with vibration at all times without lowering the success rate of notification.

Q4. How much annoyance does each notification channel cause to the user?

We answer this question by studying the questionnaire filled out by the users. Figure 5 (a) shows the results. We have the following findings.

Finding 8. Sound results in the highest level of annoyance to the participants followed by vibration. As mentioned, given that vibration has higher success rate than sound (even in quiet environments), this result means vibration must be preferred over sound.

Finding 9. Vibration and sound are especially intolerable for camera. We believe this is because these notifications heavily interfere with user's legitimate use of the camera.

Q5. Do sensor notifications' effectiveness degrade over time?

We measure the success rate of notifications for the first and second half of the study (i.e., first 3 days and last 4 days) and show the results in in Figure 5 (b).

Finding 10. We find that sensor notifications' success rate degrades over time. This poses a challenge for time-sensitive notifications deployed to be used for extended periods of time. Note that our user study spans one week only and degradation can be worse after a longer period of time.

5. DISCUSSIONS

Based on our findings, we make suggestions for the choice of sensor notifications. First, we suggest to use our new display notification (§2) for the camera. This notification channel achieves high success rate in light environments where malicious access to camera can invade user's privacy. Moreover, it causes low annoyance and does well in unambiguity.

Second, we suggest to use different vibration patterns (e.g., constant vs. chopped) for microphone and location sensor. Vibration achieves the highest success rate in all environments, including dark environments, and causes less annoyance compared to sound. Moreover, user can recognize different vibration patterns. For these two sensors, textual Android notification on the display can be accompanied to help the user recognize the notification even better.

As part of our future work, we plan to investigate the impact of other context information, e.g., orientation, on the effectiveness of notifications. Moreover, we plan to design new types of notifications, e.g., by combining existing ones, using new channels such as a smartwatch, or by using context information at runtime, and evaluate them in user studies. We also plan to evaluate using the best channel for all notifications followed by disambiguation on the display, as suggested earlier.

6. RELATED WORK

Desktops' and laptops' webcams use sensor notifications as well: they use an LED to notify the user when the webcam is recording. In a user study, Portnoff et al. [14] found that when the user is performing computer-based or paper-based tasks, the LED notification succeeds in only 45% and 5% of times, respectively. They also found that if a full-screen glyph on the screen is used for the notification, the success rates increase to 93% and 59%, respectively. Our study tries to answer similar questions, but for mobile devices, which have a different usage model and different notification channels from those of desktops and laptops.

Several existing works present solutions for better notification systems for mobile applications [5–7,10,13,15]. These solutions identify the best context (e.g., user's activity, time, and location) to send a notification to users in order to more successfully reach them and to reduce their cognitive load. These solutions are mainly for notifications by applications, and hence can be safely delayed. In contrast, our study is for time-sensitive notifications by the system, where it is critical that the user is notified as soon as possible.

Patil et al. study the trade-offs between immediate and delayed feedbacks to the users about privacy-sensitive events, e.g., location sharing [11,12]. Among others, they find that immediate feedback increases the sense of privacy violation in users. In our study, unlike theirs, we study the effectiveness of channels for time-sensitive notifications.

Viola presented a system design for enforcing sensor notifications in a trustworthy manner [8]. It uses the hypervisor or the operating system kernel to insert formally verified checks in order to guarantee that a sensor notification cannot be disabled by malware. It, however, does not address the choice of the notification channel. It does not present a design for implementing the notifications either.

7. CONCLUSIONS

We presented the results of a 40-participant user study conducted to understand sensor notifications on mobile devices. Our study showed that none of the existing channels, i.e., LED, vibration, sound, display, and Android notifications achieve more than 24% success rate in grabbing the user's attention. Moreover, it showed that device's physical context has an important impact on the effectiveness of the notification. Based on our findings, we suggested using the display notification for camera and different vibration patterns for microphone and location sensor.

8. REFERENCES

[1] Dendroid: Android Trojan Being Commercialized. http://blog.trustlook.com/2014/03/20/dendroid-android-trojan-commercialized/.

[2] How the NSA can 'turn on' your phone remotely. http://money.cnn.com/2014/06/06/technology/security/nsa-turn-on-phone/.

[3] Man spies on Miss Teen USA. http://www.reuters.com/article/2013/10/31/us-usa-missteen-extortion-idUSBRE99U1G520131031.

[4] Men spy on women through their webcams. http://arstechnica.com/tech-policy/2013/03/rat-breeders-meet-the-men-who-spy-on-women-through-their-webcams/.

[5] J. E. Fischer, C. Greenhalgh, and S. Benford. Investigating Episodes of Mobile Phone Activity as Indicators of Opportune Moments to Deliver Notifications. In *Proc. ACM MobileHCI*, 2011.

[6] J. Ho and S. S. Intille. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proc. ACM CHI*, 2005.

[7] A. Mehrotra, M. Musolesi, R. Hendley, and V. Pejovic. Designing Content-Driven Intelligent Notification Mechanisms for Mobile Applications. In *Proc. ACM UbiComp*, 2015.

[8] S. Mirzamohammadi and A. Amiri Sani. Viola: Trustworthy Sensor Notifications for Enhanced Privacy on Mobile Systems. In *Proc. ACM MobiSys*, 2016.

[9] I. Muslukhov, Y. Boshmaf, C. Kuo, J. Lester, and K. Beznosov. Understanding Users' Requirements for Data Protection in Smartphones. In *Proc. IEEE Int. Conf. on Data Engineering Workshops (ICDEW)*, 2012.

[10] T. Okoshi, J. Ramos, H. Nozaki, J. Nakazawa, A. K. Dey, and H. Tokuda. Attelia: Reducing User's Cognitive Load due to Interruptive Notifications on Smart Phones. In *Proc. IEEE PerCom*, 2015.

[11] S. Patil, R. Hoyle, R. Schlegel, A. Kapadia, and A. J. Lee. Interrupt Now or Inform Later?: Comparing immediate and delayed privacy feedback. In *Proc. ACM CHI*, 2015.

[12] S. Patil, R. Schlegel, A. Kapadia, and A. J. Lee. Reflection or Action?: How Feedback and Control Affect Location Sharing decisions. In *Proc. ACM CHI*, 2014.

[13] V. Pejovic and M. Musolesi. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proc ACM UbiComp*, 2014.

[14] R. S. Portnoff, L. N. Lee, S. Egelman, P. Mishra, D. Leung, and D. Wagner. Somebody's Watching Me? Assessing the Effectiveness of Webcam Indicator Lights. In *Proc. ACM CHI*, 2015.

[15] L. D. Turner, S. M. Allen, and R. M. Whitaker. Interruptibility Prediction for Ubiquitous Systems: Conventions and New Directions from a Growing Field. In *Proc. ACM UbiComp*, 2015.