Neither Access nor Control: A Longitudinal Investigation of the Efficacy of User Access-Control Solutions on Smartphones

Masoud Mehrabi Koushki, Yue Huang, Julia Rubin,
and Konstantin Beznosov, *University of British Columbia*

https://www.usenix.org/conference/usenixsecurity22/presentation/koushki

This paper is included in the Proceedings of the
31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Neither Access nor Control: A Longitudinal Investigation of the Efficacy of User Access-Control Solutions on Smartphones

Masoud Mehrabi Koushki, Yue Huang, Julia Rubin, and Konstantin Beznosov
University of British Columbia
{mehrabi, huang13i, mjulia, beznosov}@ece.ubc.ca

Abstract

The incumbent all-or-nothing model of access control on smartphones has been known to dissatisfy users, due to high overhead (both cognitive and physical) and lack of device-sharing support. Several alternative models have been proposed. However, their efficacy has not been evaluated and compared empirically, due to a lack of detailed quantitative data on users' authorization needs. This paper bridges this gap with a 30-day diary study. We probed a near-representative sample ($N = 55$) of US smartphone users to gather a comprehensive list of tasks they perform on their phones and their authorization needs for each task. Using this data, we quantify, for the first time, the efficacy of the all-or-nothing model, demonstrating frequent unnecessary or missed interventions (false positive rate (FPR) = 90%, false negative rate (FNR) = 21%). In comparison, we show that app- or task-level models can improve the FPR up to 88% and the FNR up to 20%, albeit with a modest (up to 15%) increase in required upfront configuration. We also demonstrate that the context in which phone sharing happens is consistent up to 75% of the time, showing promise for context-based solutions.

1 Introduction

Providing strong physical security for smartphones is of utmost importance nowadays. Continued advancements in the capabilities of phones have enabled their use for a diverse range of applications, leading to users storing and accessing highly sensitive data and services (e.g., health data and tax filings) on the devices [18]. This situation has increased the damage the users would incur if there were any unauthorized physical access to their phone [42,43]. Thus, to mitigate such risk, practitioners and researchers have increased efforts to improve the physical security system of phones [3].

Generally, any physical security system consists of two components: the *authentication system*, which confirms the user's identity, and the *access-control* (a.k.a. *authorization*) *system*, which ensures the authenticated user can access only allowed functionalities [31].

The improvements to the authentication on smartphones, so far, have been significant. In the beginning, phones had the system designed around the something-you-know model, which translated to knowledge-based unlocking methods, such as a passcode or pattern. However, due to usability [25,47], memorability [32], and security issues [41,68,69] with these methods, there was a shift toward a something-you-are model. This is when manufacturers started offering biometric unlocking methods, such as fingerprint or face recognition [7,40,57]. More recently, implicit authentication solutions (which leverage behavioral biometrics) have been developed for smartphones as well [16,19,20,38,46].

For access control on smartphones, in contrast, there has not been much change. The system is still based on the all-or-nothing model; the user can utilize all or none of the phone's functionalities based on its unlocked or locked status [29,45]. This stagnancy is despite ever-increasing evidence that the system is inefficient in meeting users' needs in two key areas.

Firstly, the system disregards any differences in task (a.k.a. fine-grain actions) security requirements. Even though tasks performed on smartphones have different levels of sensitivity [14,18,29], the access-control system treats all of them the same. For example, whether the user wants to read a book or perform a financial transaction, the phone requires the same level of unlocking. Studies suggest that users perceive this model of protection as sometimes unnecessary (e.g., for reading the book) and sometimes insufficient (e.g., for financial transactions), and, are dissatisfied with it [35,37,59].

Secondly, the system lacks support for device sharing, despite the prevalence of the practice. Phone owners share their devices for a variety of reasons, such as financial necessity or technical help [24,35,44]. Yet the phones do not support secure sharing [29,43]. This omission results in the insecure practice of primary users sharing their passcodes or adding the biometrics of the secondary users to the phones [44]. This leads to primary users have limited ability to control what secondary users can access on the phone, which leads to unauthorized access and user dissatisfaction [35,43,44,50].

There have been several attempts at addressing these shortcomings. For example, to facilitate secure phone sharing, both Android and iOS now allow locking an app on the screen and not allow switching apps without the passcode [64, 65]. Also, to accommodate differences in task sensitivity, context-based [28] solutions have been developed.

However, the efficacy of the proposed solutions has not been evaluated empirically. As such, it is unclear if they succeed in addressing the shortcomings of the incumbent system. Worse, a lack of understanding still exists in the literature, regarding (1) the extent of the all-or-nothing model's failure in meeting users' needs, even in terms of basic measures of a false positive rate (FPR) and false negative rate (FNR); (2) the improved level of performance of the proposed alternatives under the same criteria and the trade-off in terms of increase in the required up-front configuration; and (3) the consistency of the contextual factors across sharing instances, to show if it is practical to use them for better access control.

In this paper, we addressed these gaps by conducting a longitudinal diary study. We asked a near-representative sample ($N = 55$) of the US smartphone-user population to install our custom Android app on their personal phones for 30 days. They used it to report their their access control needs, and the details (e.g., time, location) of their phone-sharing events.

Using this data, we estimated for how many tasks users perform each authorization solution either forces unnecessary authentication (i.e., a false negative (FN)) or fails to prevent unauthorized access (i.e., a false positive (FP)). We also examined the value distribution of the contextual factors across all of the reported sharing events.

The results found (unsurprisingly) the all-or-nothing model to be inefficient, with its average FPR and FNR being as high as 90.3% and 21.2%, respectively. In contrast, we found that app-level models could potentially decrease these rates to 4.8% and 11.3%; task-level ones could further improve the FNR to 1.7% (without changing the FPR), albeit with a 15% increase in configuration size. Hence, we found that overall an app-locking solution might strike the best balance between security and usability for most users. Lastly, we found phone sharing happens in the same context up to 75% of the time.

In summary, we make the following contributions to the field of access control on smartphones:

- A detailed quantitative view on users' authorization needs. We offer insight into how the needs differ across the categories of task functionality and among users.
- A quantitative evaluation of how the all-or-nothing model and its alternatives meet the access-control needs of users. These measurements provide stronger evidence as to how and why the status quo is suboptimal.
- A longitudinal investigation into the context of phone sharing. Given the subsequent consistency of contextual factors, we demonstrate how incorporating them into access-control decisions could help with a better balance between users' needs for both security and usability.

2 Related Work

2.1 Smartphone Users' Access-Control Needs

Several studies have qualitatively investigated users' needs. For example, Mazurek et al. [45] interviewed 33 smartphone users. They found that users' ideal access-control policies could not be easily defined in standard role-based terms and depended on factors such as location and presence of certain individuals. They also found that, as incumbent solutions cannot uphold such complex policies, users resort to constructing ad hoc solutions, such as hiding files. Based on such observations, the authors argued that users require reactive policy creation and finer-grain access control than an all-or-nothing model. Similarly, Hayashi et al. [29] interviewed 20 users and found that their needs go beyond all-or-nothing. Their participants wanted at least one of their apps to be protected by a lock, half to be without protection, and 20% to be split (only parts of them locked). Hence, the authors found that users' authorization needs are even finer-grain than app-level.

To understand users' authorization needs in phone-sharing settings, Karlson et al. [35] interviewed 12 users to explore why they shared phones, with whom, and the concerns they had when sharing. They found that participants expressed strong preferences about which data and functionality should be available to each guest user. The authors also found that sharing preferences might be location dependent. Later, Hang et al. [24] conducted focus groups with 25 participants and found sharing to be often impromptu. They also found sharing preferences to be strongly app and data dependent.

Jacobs et al. [34] reported a similar study on couples' practices with single-user device access. They conducted 20 interviews and an 8-day diary study. They found that sharing preferences often depended on content type, making the all-or-nothing model impractical.

Lastly, Matthews et al. [44] investigated sharing across multiple devices. They conducted a survey with 99 households and a 21-day diary study. They found device and account sharing to be common. They also identified six different types of sharing, ranging from borrowing devices to getting technical help. They suggested that access control could be designed differently for each sharing type. They also found that sharing often happened in the sharer's presence.

2.2 Alternatives to All-or-Nothing

Several solutions have been proposed to alleviate the deficiencies of the all-or-nothing model.

The first set of solutions provides task sensitivity (requiring authentication only when the app/task being used requires it). Commercially, both Android and iOS allow *lock screen access* to some apps [54]. This solution keeps the all-or-nothing model mostly intact but allows the user to launch certain nonsensitive apps (e.g., camera, calculator) without

Table 1: Summary of proposed and deployed user access-control solutions on smartphones.

	Category	Approach	Proposed and Deployed Solutions
Task sensitivity	All-or-nothing	Allows authorized users all functionality but none to unauthorized	Incumbent on current phones
	Lock screen access	Permits identical access to all-or-nothing, except for allowing access to a few apps without unlocking	Deployed on most phones [54]
	App-level access	Grants or denies access to each app individually	Progressive authentication [59] and several others [12,48,61,66]
Phone sharing	All-or-nothing	Allows secondary users full access to the phone	State of practice for most users [44]
	Profile switching	Provides a separate profile with isolated apps and data for each secondary user	Deployed on Android phones [30]
	Resource based	Denies secondary users access to sensitive phone resources (Wi-Fi, Bluetooth)	DiffUser [51]
	Session based	Specifies individually what apps the secondary user can access in each shared session	xShare [39], app pinning [64]

unlocking the phone. Alternatively, Riva et al. propose a more elaborate solution called *progressive authentication* [59]. It determines a level of confidence in the user's authenticity and only prompts for authentication when a launched app requires high confidence. There have also been several context-based solutions that allow access to certain apps in specific contexts (e.g., location) without authentication [12,48,61].

Another set of solutions aims to provide better phone-sharing support. Commercially, *profile switching* [30] is the incumbent solution on Android. It allows the device owner to create separate profiles for secondary users. However, research has shown that users often do not utilize profiles even when set up, due to high physical and cognitive overhead [9, 17]. Android and iOS also now support *app pinning* [64, 65]. It allows the owner to limit sharing to a single app by fixing the app on the screen and preventing the switching of apps.

Researchers have proposed more elaborate solutions. Ni et al. [51] propose *DiffUser*, which implements a role-based model of access control. Secondary users are assigned to a role: administrator, normal user, or guest. Each role has restricted access to certain resources (e.g., normal users can access SMS and contacts but cannot install or uninstall apps). Liu et al. [39] propose *xShare*, which allows the sharer to quickly put the phone in a restricted mode before handing it to a sharee. The sharer must respecify what apps are available in this mode every time they enable it.

To summarize, Table 1 provides an overview of the existing solutions, categorizing them based on their approach to access control. We should note that in this paper we were focused on solutions that control a human operator's access to a phone. Therefore, OS-level solutions that control apps' or processes' access to system resources, such as FlaskDroid [10] and others [4,5,67,71], were out of our scope.

2.3 Gaps in the Literature

Overall, we identified several gaps in the literature, which this paper aims to address. Firstly, while prior studies demonstrated the existence of issues with all-or-nothing, they did not offer insights into the prevalence of the issues. For example, while all-or-nothing's obliviousness to task sensitivity is suboptimal [24,29], it remained to be investigated what proportion of users' tasks are actually sensitive or how sensitivity varies by functionality and across users. Such investigations would help researchers and developers to understand how dire these issues really are and whether there is an actual need for the other solutions. Only when the variance of task sensitivity among users is high can one argue for solutions that increase authorization granularity. Otherwise, the high configuration effort required for such solutions makes them unattractive for most users. We address this gap in Section 4.1.

Furthermore, even if a need for new solutions is determined, the literature [24,34,35,44] did not offer quantitative measurements of users' needs. As such, there could not have been (and was not) any attempt to evaluate and compare the efficacy of all-or-nothing and its alternatives. This left it unclear whether the proposed alternatives would succeed in addressing the issues with all-or-nothing and if/where there would be a need for further research. We report our findings on the efficacy of different access-control models in Section 4.2. Last but not least, while the existing literature showed that users' access-control preferences for phone sharing indeed depended on contextual factors (e.g., location [45] and content [29, 34, 44]), the consistency of such factors had not been demonstrated. To determine the prospects of improving access-control decisions by incorporating these factors, it is important to understand how consistent they are. We shed light on this in Section 4.3.

3 Methodology

Formalizing what we discussed in Sections 1 and 2, our study was designed to answer the following research questions (RQs):

- RQ1: What tasks do smartphone users perform on their phones? What are their sharing preferences for the tasks?
- RQ2: To what extent, in terms of the FPR and FNR, do all-or-nothing and the alternative solutions meet the users' needs? How do they compare in configuration size?
- RQ3: How consistent are contextual factors across phone-sharing events?

As mentioned, we conducted a diary study to answer these questions. The study involved participants installing our Android app on their phones and using it to report the following data every day: (1) the tasks they performed with each of their apps, (2) the people (if any) they would allow to perform each task, and (3) the details (e.g., time, location) of any instances of sharing their phone with others.

Inspired by Hayashi et al. [29], we used tasks as the means to collect users' authorization needs. Conceptually, we defined a task as a distinct series of actions that could be performed with a mobile app and that was distinguishable in terms of purpose and sharing preferences. Technically, they comprise the functionality affordances of a mobile app, similar to affordances of real-world objects or user interfaces [52]. For example, the app Gmail affords the tasks "Sending email" and "Reading email," according to our participants.

To solicit tasks, we asked the participants to declare what they used an app for, while separating actions that had different purposes. We also asked them to further break down a task if there were parts of it they were unwilling to share with others. For example, if they had used the SMS app to both send and receive texts, they were asked to declare sending and receiving as separate tasks if they were willing to let others do one (e.g., the sending) but not the other (the reading). Otherwise, they could declare "sending and receiving SMS" as the task.

We used a longitudinal design to accommodate known limitations of human memory [8], only requiring participants to recall details of their phone usage (e.g., their tasks) over one day. While this still put some memory burden on the participants (and thus could have led to recall bias), an alternative cross-sectional design [70] would have been worse, unrealistically requiring participants to recall at one sitting the details of their phone usage from over a month.[1]

3.1 Data Collection

Our app kept track of the participants' daily app usage and invited them every night to fill out a diary. Using an Android app, instead of paper- or web-based diaries, allowed us to avoid asking participants repetitive questions (e.g., asking them to list all the apps they had used) and to also ask detailed questions about app usage context, as explained below.

The diary questions were fully structured. First, the participant was asked if they had shared their phone during the day. If so, they were shown a list of apps launched on the phone that day and were asked to indicate which apps were shared (see Figure A in the paper's online supplementary material [2] for an example).

Next, our app would select five apps (either shared or used by the participant themselves) to probe further about. We chose five because pilot studies (described later in this section) showed that it took ten minutes on average to complete a diary with this number of apps; taking any more time daily would result in lower data quality and higher chances of skipping diary questions or even dropping out of the study.

The five apps were selected using an algorithm (see supplementary material [2] for a flowchart). First priority were apps the participant reported sharing for the first time. We anticipated such apps to be rarer. If there were more than five such apps, we prioritized selecting those with higher usage time (i.e., those that had more time in the foreground, based on Android's UsageStats service [56]). If there were fewer than five newly shared apps, we selected from apps the participant used themselves for the first time (while again prioritizing apps with higher usage time). If these two priorities did not fill the quota, we filled it by randomly selecting apps.

Next, the participants were asked to provide the following data about each of the five selected apps:

For both shared and nonshared apps: A list of tasks performed with the app (either the participant had performed or they believed a sharee had). They could create new tasks or select from the ones declared previously (see Figure F in supplementary material [2]). They were also asked to specify their sharing preferences for each task. The question read, "For each task, select whom you'd generally allow to perform it on your phone." The answer options were "No one," "Anyone," or "Specific people." If they chose "Specific people," they were asked to provide a list of individuals. To reduce the cognitive load of this option, a list of individuals previously defined by the participant was provided (see Figure G in supplementary material [2]).

Specific to shared apps: Participants were shown a list of times the app in question was launched on the phone. They were asked to specify whom (either themselves or a sharee) used the app each time (see Figure C in supplementary material [2]). To specify a user/sharee, the participants needed to give them a nickname to use for any subsequent reporting of sharing with the same person. They also had to declare their

[1] As evidence in support of our design choice, we had only two instances of a participant stating in a daily diary that they did not remember sharing their phone.

Table 2: Demographics of study participants.

Variable	Category	% (#) of Participants
Gender	Female	47.3 (26)
	Male	52.7 (29)
Age	18-29	9.1 (5)
	30-49	67.3 (37)
	50-64	20 (11)
	65 and higher	3.6 (2)
Education	High school	27.3 (15)
	Associate	18.2 (10)
	Bachelor	45.5 (25)
	Graduate degree	9.1 (5)
Annual income (in US$1,000s)	Less than 10	5.5 (3)
	10-29	10.9 (6)
	30-60	30.9 (17)
	60-100	40.0 (22)
	100-150	5.5 (3)
	More than 150	7.3 (4)
Ethnicity	Asian	1.8 (1)
	Black	7.3 (4)
	Hispanic	9.1 (5)
	White	81.8 (45)

relationship with the sharee (e.g., spouse). Afterward, each time the app was reported to have been shared the participant was asked to specify the location where the sharing happened and whether they were present at the time of sharing (see Figure E in supplementary material [2]).

3.2 Pilot Studies

We conducted three pilot studies to test our data collection and analysis design. Detailed descriptions of them are provided in Appendix A.1. In summary, the results allowed us to refine the wording of some questions, thereby improving correct interpretation of the diary questions by participants. We also used the results to estimate the preferred daily diary time limit (10 minutes) and length of the study (1 month), and improve communications in the consent form. These improvements addressed security and privacy concerns that could have potentially swayed future participants from participating.

3.3 Participant Recruitment

We recruited participants through MTurk. We published an ad, inviting those interested in a "study of phone usage and sharing habits" to complete a screening survey (the survey questions are provided in the online supplementary material [2]), for a US$1 compensation. The ad was only visible to Turkers who were located in North America (according to MTurk) and had an approval rate of at least 85%. We received 408 responses to the survey and invited 226 of the respondents to install our app. The sample size was determined by power analysis, assuming 85% confidence level, 5% margin of error, and a target population size of 330 million for the US. We excluded participants who did not have an Android phone (67 entries) or showed clear signs of duplicate or low-quality data (20 entries; e.g., claiming to use Face ID on an Android phone). We also selected randomly from those who had a surplus of demographic quota in our sample (e.g., younger age ranges). Out of those invited, 65 installed the app, and 55 eventually completed the study.

Data collection took place between May and August 2021. At the end of the study, each participant was compensated with a gift card worth US$20 plus US$2 for every daily diary they completed. Compensation was paid per diary as a whole, not per report of sharing, in order to avoid participants providing bogus data for extra credit. This compensation model was inspired by other longitudinal studies [13, 36, 62] and was designed to reduce the probability of early drop out.

Our final sample was fairly representative of the US smartphone-user population, which is where all of the participants were located. As Table 2 shows, the sample was diverse in terms of age, gender, education, and income groups. We also performed chi-square tests. They showed no statistically significant differences (p-values > 0.05), in terms of the distribution of the above demographics, between our sample and the general US smartphone users as of 2019, as reported by the Pew Research Center [11]. However, our sample was not very diverse in terms of ethnicity. Age distribution was also skewed, even if the difference was not statistically significant. We discuss sample limitations further in Section 6.

3.4 Ethics

Data collection was conducted according to the policies and regulations of the University of British Columbia (UBC) and Canada. All study procedures were reviewed and approved by UBC's Behavioural Research Ethics Board (Certificate ID H20-03155). All types of data that the study app would collect and the purpose for that collection were disclosed, and participants consented to its collection. To preserve participants' privacy, four measures were employed: (1) the app was not programmed to collect any data outside the scope of the study; (2) the collected data was uploaded on a daily basis through an SSL-encrypted connection to a server hosted in Canada, deleted from the phone after upload, and stored on an encrypted disk; (3) personally identifiable information (e.g., contact email) was collected as part of the web-based screening survey but not through the study app; and (4) once the study ended, the app automatically disabled itself and prompted the users to uninstall it (see Figure Q in supplementary material [2]).

3.5 Data Analysis

RQ1 [Tasks]: To create a cohesive list of tasks, two researchers aggregated the tasks reported by the participants to merge functionally duplicate ones. They did so by performing *qualitative inductive coding* [6]. Every day, each researcher mapped each newly declared task to one of the existing ones in our codebook. A mapping meant that the researcher believed the new task was the same as the previously declared one. The criteria for mapping tasks was that (1) they would either be phrased identically or were similar in functionality, and (2) the researcher could not envision a scenario in which they would require different security, based on the participant's prior data. If a new task could not be mapped to an existing one, it was added to the codebook as is. The researchers met once a week to resolve any differences (disagreement rate was 4.5%). All disagreements were resolved; if the researchers could not agree on an aggregation, both of the declared tasks would enter the codebook.

Once the task codebook was finalized, the two researchers performed categorization of the tasks based on functionality. The aim was to use this categorization to investigate the correlation between functionality and task sensitivity.

To perform the categorization, each researcher labelled each task with one of the 32 functionality categories the Google Play store uses for apps [22]. The researcher used Google Play's guidelines [22] to decide which category would be used for a hypothetical app only affording that task. The guidelines provide broad examples of what apps should fit in each category (e.g., for the category "Entertainment," the examples read "Streaming video," "Movies," "TV," and "Interactive entertainment.") Next, the researchers met and resolved all differences in labeling, of which there were 69 instances. Resolutions were achieved through either agreeing on one category or merging categories that had all overlapping apps.

RQ2 [Comparing access-control solutions]: We used three metrics to compare the solutions:

- *False positive rate (FPR)* was calculated as the ratio of the number of tasks a solution mistakenly makes available to unauthorized users (e.g., by making all apps available to all users after an unlocking, even if only one app is intended to be shared with the user) by the total number of tasks. The FPR directly impacts the security of a solution, as a higher rate would indicate higher chances of unauthorized access to private data.
- *False negative rate (FNR)* was calculated as the ratio of the number of tasks that a solution mistakenly denies access to authorized users (e.g., by not letting unauthenticated users access a task that is intended to be shared with "Anyone") by the total number of tasks. This measure reflects on the usability of a solution, as it signifies the chances that a solution imposes unnecessary unlocking overhead on users.
- *Configuration size rate (CSR)* was calculated as the ratio of the number of access-control preferences each user would have to explicitly specify for a solution (e.g., one preference would be to indicate that an app should be available to "Anyone"), by the total number of a user's preferences (in this paper, we assume this to be their total number of tasks). We use the CSR as a basic estimate of the amount of effort necessary for a user to switch to a solution, without us having to assume any particular design for the user experience of the solution. For example, a CSR = 100% would mean that the solution expects the user to explicitly specify their sharing preference for all their tasks, where as a CSR = 1% would require only 1/100 of that effort.

The metrics were calculated separately for each participant and then averaged over all participants. It should be noted that we calculated both the FPR and FNR by counting the number of tasks, not the number of times a task was performed by a participant. In the scenarios (described below), we also did not differentiate between when an unlocked phone is shared and when the passcode is shared. While these choices could make the calculated FNR less reflective of the actual user experience (e.g., the user might face more interruptions from frequently done tasks), we made them to limit the impact on the FPR by frequently done nonsensitive tasks, which could make a solution look unrealistically secure. For example, an unauthorized execution of a highly sensitive task, such as banking, could be masked by the many authorized executions of a trivial task, such as checking the weather.

For task sensitivity, we calculated the metrics in seven scenarios, each corresponding to one type of solution:

- **ALL:** This scenario corresponds to one possible case of the all-or-nothing model. It represents a case where the user would not enable unlocking. Hence, the FNR would be zero. However, any task labelled as "No one" or "Specific people" would constitute an FP because it would mistakenly be available to anyone. This scenario is important to consider; it was estimated in 2020 to be the state of practice for 10% of smartphone users [46].
- **NOTHING:** This is another possible case for the all-or-nothing model. Here it is assumed that the user would enable authentication but not share their passcode. Hence, the FPR would be zero. However, any task labelled as "Anyone" or "Specific people" would constitute an FN as it would not be available to the corresponding users. This scenario is also important to consider; it represents the state of practice for nearly 90% of users [46].
- **LOCK_SCREEN_ACCESS:** This scenario represents the lock screen access solution. It is similar to NOTHING, but we assumed camera, calculator, and flashlight apps would be available before unlocking.

- **APP_CONSERVATIVE:** This represents a case where users would be able to lock apps individually but not individual tasks within them. It is representative of how progressive authentication [59] or other app-level solutions (see Table 1) perform. Calculating the FPR and FNR for it, however, requires resolving conflicts in the participant's labeling of an app's tasks. In this conservative scenario, we assume that the user would want to lock an app if they label any of its tasks as "No one."
- **APP_MAXIMAL:** This is similar to the APP_CONSERVATIVE scenario but uses a different strategy for conflict resolution. In this case, the model would assign access to an app based on the majority vote of the user's labeling of its tasks.
- **TASK_CONSERVATIVE:** In this hypothetical scenario, we assume that the system has perfect knowledge of the user's sharing preferences. However, inconsistent labeling of tasks would still constitute FPs/FNs, as they represent sharing preferences that a task-level system cannot uphold.[2] We devised this scenario to gauge an upper bound for the efficacy of task-based access control. To measure the FPR and FNR, a conflict resolution strategy is needed. In this conservative scenario, we assume that the user would prefer the most restrictive of their provided labels for each task.
- **TASK_MAXIMAL:** This is similar to TASK_CONSERVATIVE but uses a different conflict resolution strategy. Here we assume that the user would prefer to have tasks protected according to the majority vote of its labels. For example, if the user has labeled a task twice as "No one" and once as "Specific people," "No one" would be selected as its final label.

For phone-sharing solutions, we considered four scenarios:

- **ALL_OR_NOTHING:** We assume that the user either has not enabled unlocking or has enabled it but has shared their passcode with the sharee. Hence, we calculate the FPR and FNR by assuming the sharee has access to all tasks on the phone. This scenario represents the state of practice among most users [44].
- **PROFILE_SWITCHING:** This represents a case where the user would create a separate profile for each sharee and allow them to install their own apps. Hence, the FPR and FNR are calculated assuming that sharees cannot perform any task not explicitly shared with them.
- **DIFF_USER:** We assume that the user would designate all sharees as "Guest." We calculate the FPR and FNR by determining whether a sharee can perform a task, based on what resources (e.g., Wi-Fi) they have access to, according to DiffUser's rules [51] (e.g., "Guest" users do not have access to Bluetooth. Refer to Ni et al. [51] for the complete list of rules). We determine the resources needed for performing a task by examining the Android permissions requested by the app that affords the task.
- **X_SHARE:** We assume that the participant would put the phone in restricted mode before sharing it, while only granting access to the apps they reported sharing in that session. The FPR and FNR are computed by holding that the sharee can perform any tasks with shared apps but none with the other apps. In addition to xShare [39], this scenario represents how session-based solutions would perform (see Table 1) as repeatedly unpinning/pinning apps can achieve the same effect.

Table A.2 in the appendix provides the exact formula used to calculate each rate for each scenario.

RQ3 [Phone-sharing contextuality]: For each participant, we examined the variations of the following factors in all their reported cases of phone sharing: location, participant's presence, their relationship with the sharee, and functionality category of the shared tasks. We focus on these factors only, as they were reported by prior qualitative work to be correlated with sharing needs of users (see Section 2). Our goal was to corroborate such correlations quantitatively.

4 Results

4.1 RQ1: Tasks and Authorization Needs

Overall, the participants reported performing a large and functionally diverse set of tasks. Collectively, they declared 1,149 distinct tasks in total (after aggregation, as described in Section 3) for 571 distinct apps. On average, each participant reported performing 74 tasks (min = 24, max = 142) using 48 apps (min = 12, max = 103).

Examining the participants' access-control preferences for the tasks, we found them to be highly complex. The participants indicated they were willing to share a large portion (nearly 43%) of their tasks with others. Specifically, 23.7% of the tasks were labeled as being shareable with "Specific people," whereas 19.4% were shareable with "Anyone."[3] However, these "Specific people," were not limited to a narrow group of individuals, corroborating the qualitative findings of others [35, 44, 45]. Additionally, our results provide a more detailed distribution of the sharee-sharer relationships: spouses and boy/girlfriends comprised the largest groups of preferred sharees (64.6% of all "Specific people" tasks), followed by parents (13.7% of tasks), children (13.1%), and friends (3.6%). This allows us to argue for the feasibility of implicit user identification, as we discuss in Section 5.

[2] This entails that, to accurately take into account inconsistencies in the participants' labeling of tasks, we consider different labels of a task to be different (finer-grain) tasks that this system cannot distinguish between.

[3] Note that labeling a task as shareable does not indicate that the participant actually shared it. We discuss actual sharing in Section 4.3.

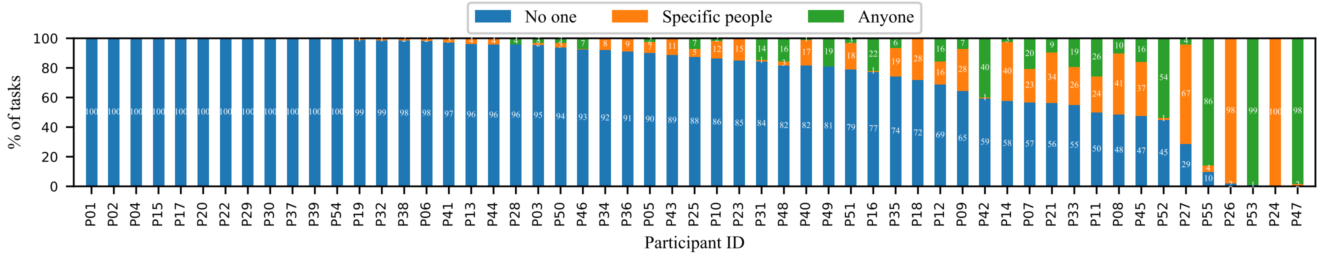

Figure 1: Distribution of sharing preferences of performed tasks for each study participant.

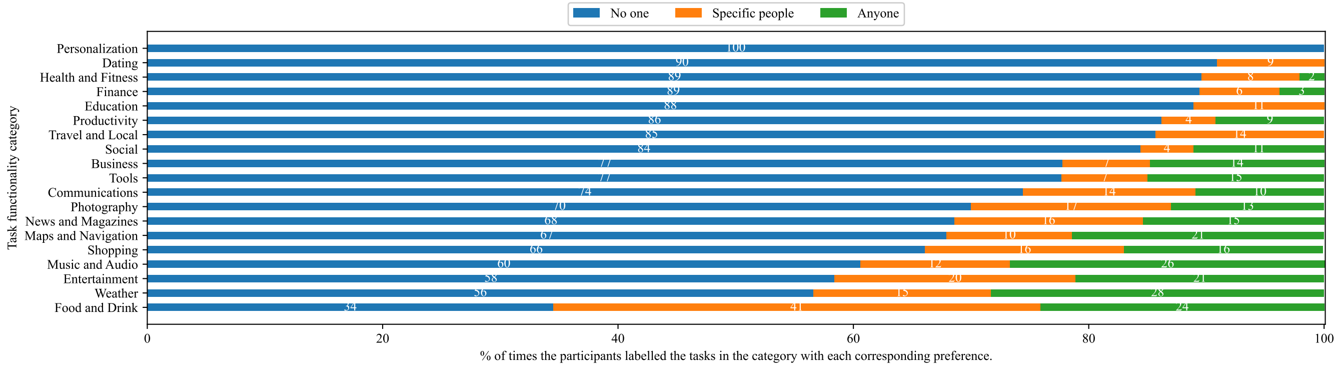

Figure 2: Distribution of sharing preferences across task functionality groups.

Adding more complexity to the matter, we found the access-control preferences to be highly individualized, as Figure 1 illustrates: About 22% of the participants were unwilling to share any tasks with anyone (e.g., participants P01, P02, and P04). We refer to them in the paper as *private users*. A small fraction, 5%, are referred to as *public users*, and they wanted to share everything with certain people (e.g., P24). Unsurprisingly, the majority (73%) were in between these extremes. We refer to them as *semiprivate users* (e.g., P7). Obviously, for public or private users, an all-or-nothing model of access control would suffice. However, a more complex system is needed for the majority: semiprivate users.

However, we found the abovementioned groups to be fairly heterogeneous in terms of demographic or phone-usage factors. This makes predicting one's access control needs by any potential future solution challenging. We tested the association between access-control categorization (i.e., whether a participant is a private, public, or semiprivate user) and several demographic and phone-usage factors. These included age (suggested by Qiu et al. [57] to be correlated with phone-locking habits), education level, hours of phone usage per day, depth of smartphone adoption (measured by the privacy app adoption questionnaire by Mehrabi et al. [38], which was based on the one by Marques et al. [43]), and whether the participant lived with someone else or shared phones with them (anticipating this would lead to different sharing habits). However, apart from age, none of the test results (see Table A.1 in the appendix) were statistically significant (p-values > 0.05) or strong (Cramér's V < 0.5). For age, while the link was statistically significant, it was not strong.

Next, we examined the correlation between functionality and task sensitivity, to gauge the feasibility of functionality-specific authorization solutions. From the categorization process we described in Section 3, we identified 19 separate functionality categories of tasks. We observed that the tasks were distributed relatively evenly among these categories, as demonstrated in Figure A.1. More importantly, however, we found a nonuniformity of access-control preferences among the task categories. As Figure 2 illustrates, while some tasks were mostly off-limits to others (e.g., the "Personalization" and "Dating" categories), other categories (e.g., "Food and Drink" and "Weather") were very often labeled as shareable, either with "Specific people" or with "Anyone." This findings suggest that (1) there is indeed a need for varying access-control treatment based on task functionality, and (2) such solutions might actually be feasible.

Next, we examined the data with higher granularity to compare how participants' task- and app-level preferences align. This was to gauge if there is a real need for higher granularity than app-level. Firstly, we found that apps often afford tasks with conflicting functionality. Our participants declared two different tasks per app on average, which were often from different categories. For example, business apps sometimes afforded tasks from "Communication," "Productivity," "Edu-

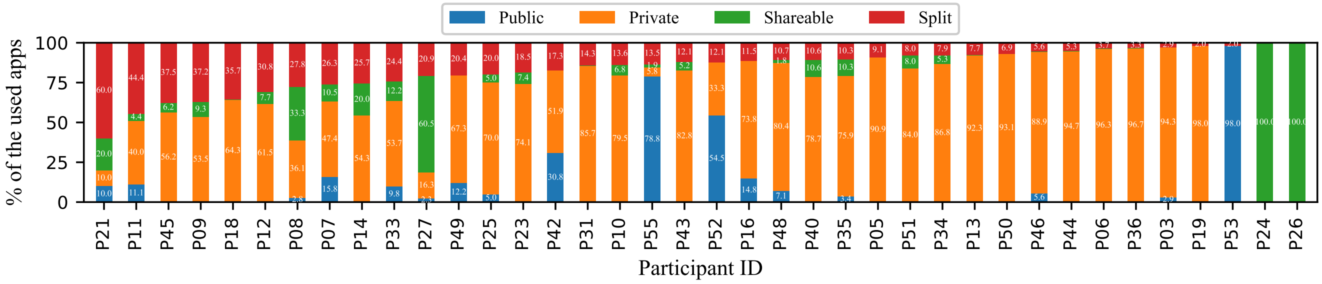

Figure 3: Distribution of app shareability categories among semiprivate study participants.

cation," or "Personalization" categories.

Consequently, we found apps to be incohesive in shareability. We observed that for each participant apps generally fell into one of the following categories when it came to the shareability of the tasks they afford:

- *Public apps* exclusively afford tasks shareable with "Anyone." Hence, conceptually, they would require no form of authentication before accessing them.
- *Private apps* are in direct contrast to public apps and afford exclusively "No one" tasks. Therefore they would almost always require user authentication in advance.
- *Shareable apps* exclusively afford tasks that are shareable with "Specific people." Hence they would require multi-user identification support.
- *Split apps* afford tasks with different levels of shareability and, as such, require not only user identification, but also granular authorization support.

Obviously, for public and private users all apps fell either into the public or the private app categories. However, for the majority of the participants (73%) the distribution of apps was not just between these two categories, as Figure 3 illustrates. The majority used split apps, which require a fine-grain model of access control. This result corroborates the findings of Hayashi et al. [29] that smartphone users consider a significant number of apps they use as split. However, our data provides a clearer picture of this issue by (1) showing the relatively high proportion of such tasks, and (2) demonstrating that the number of split apps varies significantly between users. This observation suggests the need for personalized access control, as opposed to one finer-grain solution for all.

Lastly, we also found that the perception of the shareability of an app was not consistent across study participants. Among the 206 apps that were used by more than one participant, 137 (66.5%) were categorized inconsistently by their users. For example, in the case of the Amazon shopping app, 65% of its users categorized it as private, 29% as split, 3% as shareable, and 3% as public.

In summary, we found users performing a large (1,149) and functionally diverse set of tasks (19 different categories) with their phones. We also found their access-control preferences for the tasks to be highly complex and varied by sharee (see Figure 1) and functionality (see Figure 2). Lastly, in agreement with previous work, we found apps to be incoherent units in terms of access-control requirements, with significant variance in the functionality and shareability of tasks they afford (see Figure 3). Overall, therefore, the results show quantitatively that there is indeed a need for more granular and customizable authorization solutions on smartphones.

4.2 RQ2: Comparison of Solutions

To start with, we unsurprisingly found the incumbent all-or-nothing solution to be inefficient. As Table 3 illustrates, the FPR of the ALL scenario was estimated at 90.3%, meaning ~90% of the users' tasks would be exposed to unauthorized users. This result clearly demonstrates the high risk associated with this scenario, even though 10% of users choose it anyway [46]. This acceptance of risk, however, becomes somewhat justifiable when we consider the NOTHING scenario. While NOTHING exposes no tasks to unauthorized users (FPR = 0%), a 21.2% FNR would be incurred, meaning up to 20% of the users' unlockings could be unnecessary.

We found the commercial solutions did not offer much improvement either. Obviously, the ultimate goal of a task-sensitive solution would be to reduce the FNR compared to the NOTHING scenario without increasing the CSR or FPR substantially. Yet, as Table 3 shows, the LOCK_SCREEN_ACCESS scenario fails to do so. It reduces the FPR by less than 1% only but with no change to the other metrics. This is unsurprising given the limited number of tasks afforded by the common lock screen apps (e.g., camera, calculator).

What seems to be effective, however, is increasing the granularity of access control. As the table shows, app-level models reduce the FPR by 7% with modest increases in the CSR and FNR (~0.5% and 5.5%, respectively), as in the case for both the CONSERVATIVE and MAXIMAL scenarios. This suggests increased granularity is beneficial, even when app-level decisions are made based on most-restrictive task labels, as in the CONSERVATIVE case. It is out of the scope

Table 3: The calculated FPR, FNR, and CSR of the evaluated scenarios.

	Scenario	FPR (%)	FNR (%)	CSR (%)
Task sensitivity	ALL	90.3	0.0	0.0
	NOTHING	0.0	21.2	0.0
	LOCK_SCREEN_ACCESS	0.0	20.5	0.0
	APP_CONSERVATIVE	0.0	14.0	4.8
	APP_MAXIMAL	1.2	11.3	5.9
	TASK_CONSERVATIVE	0.0	5.5	18.1
	TASK_MAXIMAL	2.3	1.7	18.1
Phone sharing	ALL_OR_NOTHING	18.1	0.0	1.8
	PROFILE_SWITCHING	0.0	0.0	16.0
	DIFF_USER	15.5	0.1	3.5
	X_SHARE	1.4	0.1	3.1

of this paper to gauge which scenario would be preferable to the end users (as it would largely depend on the final user experience). However, on a conceptual level it is clear that app-level access control can achieve the ultimate goal of a task-sensitive solution, as we described before.

Lastly, increasing the access-control granularity to task level was found to be even more beneficial, albeit with a more noticeable trade-off. As Table 3 shows, the TASK_CONSERVATIVE and TASK_MAXIMAL scenarios could further reduce the FPR as low as 1.7% but with an 18.1% increase in the CSR. This increase could be manageable, however. Considering our earlier finding that users perform 74 tasks on their phones on average, an 18% CSR would mean that users would need to specify as few as 13 sharing preferences to configure such a system. However, in contrast, configuring the NOTHING scenario would only require a one-time setup of unlocking, which could still be a noticeable gain based on a user's tolerance for an FP vs. their desire for a reduced FNR.

As for support for phone sharing, we again found the all-or-nothing model to be inefficient. As Table 3 shows, the ALL_OR_NOTHING scenario would lead to sharees not facing any unnecessary restrictions (FNR = 0% because we assume the passcode is shared with the sharee). However, more than 20% of the sharer's tasks would be exposed to unauthorized users as well, increasing the potential for security/privacy violations by social insiders, which other studies have reported to be prevalent [42, 43].

In comparison, profile switching might appear to be the ideal solution. Naturally (and in direct opposition to task sensitivity), the ultimate goal for any phone-sharing solution would be to reduce the FPR compared to the ALL_OR_NOTHING scenario without a substantial increase in the FNR or CSR. Profile switching, as shown in Table 3, seems to achieve this goal, as it makes the FPR and FNR zero. However, as the table also shows, the scheme comes with a substantial increase in the CSR, which seems to explain why users are reluctant to use it [9, 17].

The resource-based solution (the DIFF_USER scenario) does not seem to offer much improvement either. While it can reduce the FPR by ~5%, it also increases the FNR and CSR by 0.1% and 1.7% respectively, which somewhat diminishes its FPR gains. This is unsurprising given our previous finding on the multifaceted nature of current apps in terms of functionality and shareability (see Section 4.1), which could complicate mapping resources to tasks. For example, many tasks in an app might use Wi-Fi; these tasks can vary in sensitivity. Thus, it is not trivial to assert Wi-Fi as a sensitive resource.

Finally, the X_SHARE scenario appears to provide the best balance between usability and security. As seen in Table 3, it eliminates most of the false positives (FPR = 1.4%) while not increasing the FNR at all and the CSR by only 1.3%. Thus, among all the models it seems limiting access per session could be the most promising solution.

In summary, our results show the inefficacy of the all-or-nothing model and offer evidence that the current commercial solutions (e.g., the LOCK_SCREEN_ACCESS scenario) do not offer much improvement. Furthermore, our results show that app-level models could afford the best usability-security-configuration trade-off in order to support task sensitivity. For phone sharing, session-based control appears to offer the best such balance.

4.3 RQ3: Contextuality of Phone Sharing

To begin with, we found actual sharing of phones (as opposed to a willingness to share tasks, which was the basis of RQ1 and RQ2 results) to be less prevalent than reported in previous studies. Among our participants, only 16 (29%) reported sharing their phones with others. And 58 sharing events were reported in total over the course of the study. In contrast,

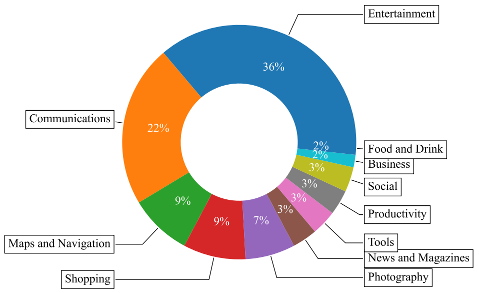

Figure 4: Distribution of functionality categories of reported shared tasks.

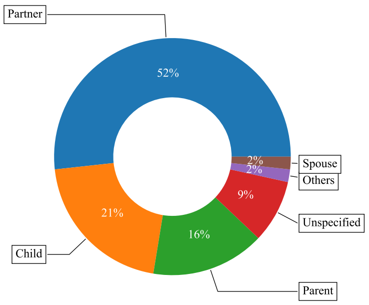

Figure 5: Distribution of reported sharee relationships.

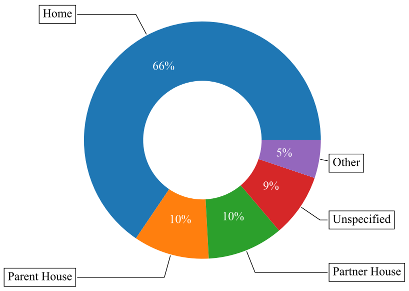

Figure 6: Distribution of reported phone-sharing locations.

Jacobs et al. [34] estimated that nearly 50% of users would share phones, and Matthews et al. [44] reported 14 sharing instances per participant (N = 25) over 21 days. Although it was out of our scope to investigate, we anticipate the decline in sharing is due to the increase in smartphone penetration of the consumer markets since the time of those studies (2016) [63], and the effects of the COVID-19 pandemic [53].

However, similar to a willingness to share, we found the tendency to actually share phones less easily distinguishable using demographic factors. Most of our anticipated antecedents showed no statistically significant correlation with a participant reporting at least one instance of sharing (p-value > 0.05, as Table A.1 shows).

Interestingly, the practice of sharing does not seem to be strictly aligned with access-control preferences either. Among the participants who reported sharing, one was a private user, one was public, and the rest (14) were semiprivate. Even though at least one participant (the private user) reported being unwilling to share any of their tasks with anyone, they reported sharing their phones on one occasion anyway. Hence, it seems sharing can indeed be impromptu sometimes, as found by prior work [34, 44].

As for the contextual factors of phone sharing, we found them to be fairly consistent. However, we should note in advance that our results might be affected by the COVID-19 pandemic and the stay-at-home orders. These circumstances might have forced participants to always share phones in the same locations and/or with the same people.

To study the contextual factors, we first examined the correlation between content and sharing preferences. Prior qualitative studies demonstrated that sharing is indeed content dependent [29, 34, 45]. Our results confirm this quantitatively. We observed that only specific task categories were reported as shared. As Figure 4 shows, most of the shared tasks were from a few specific categories (e.g., "Food and Drink," "Entertainment"), while tasks in some other categories (e.g., "Personalization") were never shared. Overall, 11 categories (58%) had at least one reported case of sharing, with 70% of all shared tasks being from 3 categories: "Entertainment," "Communication," and "Maps and Navigation." At the same time, tasks in 8 categories (42%) were never shared. This aligns well with the participants' reported willingness to share tasks (see Figure 1).

Next, we found the sharer's presence to be a determinant as well. Matthews et al. [44] found sharing to happen often when the sharer is present. Our data confirmed this finding while also showing high consistency. We found the majority (56 of 58 instances (96.6%)) of sharing events were reported to have happened in the presence of the sharer. (These are the sharing events known to the sharer, as our study relied on self-reported data.)

We also found the relationship between the phone owner and the sharee to be a significant factor. Prior studies demonstrated that users' sharing preferences depend significantly on who the phone is being shared with [9, 35, 45]. Our results confirmed this finding, too, while again showing high consistency. We observed a great majority of sharing events were with people the sharer was familiar with. As Figure 5 depicts, more than half (52%) of all sharings happened with romantic partners. Sharing with children and parents was the next most frequent, at 21% and 16% respectively. Overall, 89% of sharing instances happened with the immediate family of the sharer. This finding also aligns well with the preferences reported by the participants, discussed in Section 4.1.

Lastly, we found location to also be a fairly consistent determinant across sharing events. As Figure 6 shows, 66% of the events happened at the sharer's home. Sharing within the homes of parents and romantic partners was the next most frequent, at 10% each. And only 5% of the events happened at other locations, such as schools or other public places.

In summary, we found sharing to be universal (not limited to a specific group of users). We also found the contextual factors of phone sharing to be highly consistent (~75% of the time). Even though we found sharing to be less prevalent than previously reported, we observed phones being shared often with family members and partners, at familiar locations, in the presence of sharers, and for limited categories of tasks.

5 Discussion

Putting all of our results together, it is clear that modern smartphone users have diverse and complex access-control needs. They perform a functionally diverse set of tasks with a large number of apps. And they prefer to share several (sometimes even partially overlapping) subsets of tasks with different individuals (see Section 4.1) and in varying contexts (discussed in Section 4.3).

Making matters more complex, authorization needs vary significantly by task functionality (see Figure 1) and per user (see Figure 3). And it is difficult to predict users' preferences based on their demographic or phone-usage factors (see Section 4.1). Also, no functionality categories dominate the users' tasks to therefore limit the scope of access control to certain activities on the phone (see Section 4.1).

All these circumstances reduce the chance that any ad hoc solution (e.g., lock screen access) catering only to specific tasks or groups of users could achieve adequate efficacy. Hence, the need for robust and general-purpose access control on smartphones is now clearer than ever.

Yet the incumbent all-or-nothing solution falls short of this ideal. Firstly, it fails to provide timely and secure access to tasks, as its assumption that all tasks require the same level of protection is invalid. Our findings (Section 4.2) suggest that it could mistakenly expose more than 90% of the users' private tasks to unauthorized individuals or, conversely, unnecessarily hinder the users' access to 21% of their tasks.

Secondly, the solution fails to assert adequate control when users share their phones. We found that it could expose up to 20% of the users' private tasks to unauthorized sharees, as it forces the users to share passcodes (Section 4.2). This is especially important considering the clear distinction the users make between “Anyone” and “Specific people” tasks (Section 4.1). To them, a shared task is not necessarily a public one. So, even if they are willing to share a task, it is important to control with whom that task is shared.

As such, the dilemma users face when using the incumbent system is clearer now. They have to either (1) enable unlocking and face the overhead of 20% unnecessary authentications or (2) disable unlocking altogether for more convenience and risk 90% of their tasks being available to unauthorized users (as 10% of users actually do [46]).

Making the situation direr, most of the proposed/implemented alternative solutions do not seem to provide much improvement. The lock screen access solution only provides an insignificant improvement to the FNR (0.7%). And phone-sharing solutions, such as DiffUser, rely on restricting access to system resources to assert control, which nearly doubles the CSR in exchange for reducing the FPR marginally (see Section 4.2).

Profile switching presents more of a conflicted situation. Even though our results show it could theoretically eliminate FPs all together, its high CSR explains why most users decide not to adopt it [9, 17]. Moreover, we found evidence for the need to support impromptu phone sharing (see Section 4.3). However, profile switching does not easily support impromptu sharing, since it would require creating profiles on the spot.[4]

It seems, therefore, that users are left with neither timely access to their tasks nor proper control over whom performs them. But hope exists.

We found designs that could lead to better access control. To support task sensitivity, we found increased granularity a good starting point. The app-level models we examined were found to have nearly half the FNR of the widely used all-or-nothing system (11% vs. 21%) but with only modest increases in the CSR and FPR (see Section 4.2). However, the app-level model seems to be the sweet spot of granularity; creating even finer-grain task-level solutions would halve the FNR once more, while imposing a substantial increase in the CSR (18%). Thus, overall, our data supports wider adoption of app-locking solutions, which are currently only deployed on some Android phones [66].

For deliberate phone sharing, our results endorse session-based solutions. Approaches such as xShare [39] and app pinning, which allow users to quickly select which apps to share in each session, showed substantial reduction in the FNR (~20%), only a modest increase in the CSR (~1%), and no increase at all in the FPR (see Table 3 and Section 4.3).

Finally, our data also shows promise for context-based phone-sharing solutions. We found several contextual factors to be highly consistent (see Section 4.3), which could be incorporated into future solutions:

- Content. Only certain categories of tasks were shared by our participants. Thus, future access-control schemes could further reduce the FPR with a default denial of access to task categories universally perceived as private.
- Presence of the sharer. Deliberate sharing often happens in this situation (see Section 4.3). Thus, detecting the owner's presence (e.g., through smartwatches or other wearables) seems to be a promising approach to detect unauthorized access.
- The sharee. Phone sharing often happened with the same group of people (see Figure 5). Thus, an a priori policy definition and user isolation solutions (e.g., profile switching) could indeed work but, as discussed before, need to be made more convenient. The high proportion of close family members in the sharee distributions suggests automatic user identification (e.g., through behavioral biometrics [49]) could be one way of achieving this goal and poses a promising avenue of future research.
- Location. Most of the reported sharing events happened at familiar places, such as at home (see Figure 6). As such, location detection (e.g., using Wi-Fi or GPS as

[4] This creation would only be needed if the shared tasks were for “Specific people.” Otherwise a “Guest” profile would suffice.

done by Google Smart Lock for Android [23]) could provide better access control by limiting unauthorized access in unfamiliar locations.

In the end, we should note that our findings also demonstrate the need for flexibility in access control, as any one-size-fits-all solution would have unjustifiable trade-offs for some users. For example, if all users were forced to a highly granular task level, some users could benefit. But this level of access control would also impose a high CSR on semiprivate users, with no improvements in the FPR or FNR for them (e.g., P03 in Figure 3 who has few public apps and no split apps). For such users, an app-level system would be a better fit. Hence, it is important to consider solutions that increase granularity only when needed or only for those users who need it. This could certainly be achieved manually (e.g., by asking users directly to use a task-level system). However, as observed with profile switching (see Section 4.2), the increased cognitive load makes adoption an issue. An avenue for future research, therefore, can be to investigate the feasibility of automating this process (possibly like our implicit identification suggestion for profile switching).

6 Limitations

Any generalization of our results needs to be performed carefully due to the study's limitations.

Firstly, similar to other studies on smartphone usage [15, 27, 33], our sample was not representative of the global smartphone user population. For example, it has been shown that cultural factors affect users' unlocking behavior [26] and privacy attitudes [60]. However, due to limited resources we only included US participants in our study; as a result, one cannot generalize the results to non-US users. Cross-cultural studies are required to explore the link between culture and access control.

Secondly, our sample is not fully representative of the US smartphone user population either. While MTurk is shown to provide quality data for research in usable security [58], its known limitations (e.g., lack of diversity, tech savviness) [55] apply to our study too (for example, our participants are more than 80% white). Lack of diversity is a common limitation of smartphone studies [15, 29, 44] However, we believe that our findings are still valuable, as they provide the very first quantitative insight into users' access-control needs.

Thirdly, due to technical limitations we only included Android users in our study. Recent evidence suggests there are no significant differences in security/privacy attitudes between iOS and Android users [1]. However, cross-platform studies are required to investigate this matter further.

Fourthly, our results are based on self-reported data from the participants. Thus, as a general limitation of such studies, our data might sometimes be of lower ecological validity and not perfectly reflective of the users' true behavior [21, 46]. Also, the reported sharing events might be affected by the users' prolonged use of the all-or-nothing system. For example, the type of tasks they shared might have been influenced by what they felt comfortable sharing given the limitations of the incumbent system. Also, all-or-nothing might have caused participants to misinterpret the diary questions and believe that "Whom you'd generally allow to perform the task" meant only when they were physically presiding over the sharing. To mitigate this risk, we conducted several pilot studies (see 3.2) and did not find evidence of such varying misinterpretations. But they are still a possibility and could have caused us to underestimate the number of "Anyone" tasks. Having more such tasks, however, could only strengthen our argument that the all-or-nothing system is suboptimal, as these tasks increase its FPR (already over 20%) even further.

Lastly, as discussed in Section 4.3, our results might have been affected by the COVID-19 pandemic. For example, work-from-home orders might have influenced users' security/privacy preferences for their personal devices, which might contain work-related information [53]. Also, the consistency of location in phone-sharing events might have been due to stay-at-home orders.

7 Conclusion

Smartphones, nowadays, require strong physical security as they host an ever-increasing array of data and services. Yet most of the research and development so far has been toward their authentication systems. Their access-control systems have remained largely unchanged, still using the all-or-nothing model. In this paper, we solicited detailed task data from users and quantified just how inefficiently this model serves the users. We examined the status quo and found that most proposed/implemented alternatives do not provide much improvement either. Instead, we found that increasing the granularity of access up to a point and providing session- and context-based control might be two promising avenues for designing future systems that better support users' needs.

8 Acknowledgments

This research has been supported by a research grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) and a gift from Scotiabank to the University of British Columbia. The authors would like to thank members of the Laboratory for Education and Research in Secure Systems Engineering (LERSSE) for providing feedback on the reported research, our anonymous reviewers, and our shepherd, Mary Ellen Zurko from MIT Lincoln Laboratory, for all the feedback and suggestions they provided to improve the paper. Stylistic and copy editing by Lynn Slobogian helped to improve readability of this paper.

References

[1] Desiree Abrokwa, Shruti Das, Omer Akgul, and Michelle L Mazurek. Comparing security and privacy attitudes among US users of different smartphone and smart-speaker platforms. In *Proceedings of the Seventeenth Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, 2021.

[2] Neither access nor control. Online supplementary material. https://github.com/LERSSE/neither_access_nor_control.

[3] Nasser O Alshammari, Alexios Mylonas, Mohamed Sedky, Justin Champion, and Carolin Bauer. Exploring the adoption of physical security controls in smartphones. In *Proceedings of International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 2015.

[4] Enrico Bacis, Simone Mutti, and Stefano Paraboschi. Apppolicymodules: mandatory access control for third-party apps. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, 2015.

[5] Michael Backes, Sven Bugiel, Sebastian Gerling, and Philipp von Styp-Rekowsky. Android security framework: Extensible multi-layered access control on android. In *Proceedings of the 30th annual computer security applications conference*. Applied Computer Security Associates, 2014.

[6] Bruce L Berg, Howard Lune, and Howard Lune. *Qualitative research methods for the social sciences*, volume 8. Pearson Boston, MA, 2012.

[7] Rasekhar Bhagavatula, Blase Ur, Kevin Iacovino, Su Mon Kywe, Lorrie Faith Cranor, and Marios Savvides. Biometric authentication on iphone and android: Usability, perceptions, and influences on adoption. In *Proceedings of the Workshop on Usable Security*. Internet Society, 2015.

[8] Scott C Brown and Fergus IM Craik. Encoding and retrieval of information. *The Oxford handbook of memory*, pages 93–107, 2000.

[9] AJ Bernheim Brush and Kori M Inkpen. Yours, mine and ours? sharing and use of technology in domestic environments. In *Proceedings of the International Conference on Ubiquitous Computing*. Springer, 2007.

[10] Sven Bugiel, Stephen Heuser, and Ahmad-Reza Sadeghi. Flexible and fine-grained mandatory access control on android for diverse security and privacy policies. In *Proceedings of the 22nd USENIX Security Symposium*. USENIX Association, 2013.

[11] Pew Research Center. Mobile technology and home broadband 2019. https://www.pewinternet.org/2019/06/13/mobile-technology-and-home-broadband-2019/, 2019. Accessed: 2019-07-26.

[12] Mauro Conti, Vu Thien Nga Nguyen, and Bruno Crispo. Crepe: Context-related policy enforcement for android. In *Proceedings of the International Conference on Information Security*. Springer, 2010.

[13] Daniel DL Coppersmith, Evan M Kleiman, Catherine R Glenn, Alexander J Millner, and Matthew K Nock. The dynamics of social support among suicide attempters: A smartphone-based daily diary study. *Behaviour research and therapy*, 120:103348, 2019.

[14] Heather Crawford and Karen Renaud. Understanding user perceptions of transparent authentication on a mobile device. *Journal of Trust Management*, 1(1):7, 2014.

[15] Alexander De Luca, Alina Hang, Emanuel Von Zezschwitz, and Heinrich Hussmann. I feel like i'm taking selfies all day!: towards understanding biometric authentication on smartphones. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2015.

[16] Mohammad Omar Derawi, Claudia Nickel, Patrick Bours, and Christoph Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE Computer Society, 2010.

[17] Serge Egelman, AJ Bernheim Brush, and Kori M Inkpen. Family accounts: A new paradigm for user accounts within the home environment. In *Proceedings of the ACM conference on Computer supported cooperative work (CSCW)*, 2008.

[18] Serge Egelman, Sakshi Jain, Rebecca S Portnoff, Kerwell Liao, Sunny Consolvo, and David Wagner. Are you ready to lock? In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2014.

[19] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, Bogdan Carbunar, Yifei Jiang, and Nhung Nguyen. Continuous mobile authentication using touchscreen gestures. In *Proceedings of the IEEE Conference on Technologies for Homeland Security (HST)*. IEEE, 2012.

[20] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148, 2013.

[21] Robert M Gonyea. Self-reported data in institutional research: Review and recommendations. *New directions for institutional research*, 2005(127):73–89, 2005.

[22] Google. Choose a category and tags for your app or game. https://support.google.com/googleplay/android-developer/answer/9859673?hl=en. Accessed: 2021-08-24.

[23] Google. Use Google smart lock. https://bit.ly/2XTnGeG, 2019. Accessed: 2020-06-12.

[24] Alina Hang, Emanuel Von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. Too much information! user attitudes towards smartphone sharing. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction*, 2012.

[25] Marian Harbach, Alexander De Luca, and Serge Egelman. The anatomy of smartphone unlocking: A field study of android lock screens. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2016.

[26] Marian Harbach, Alexander De Luca, Nathan Malkin, and Serge Egelman. Keep on lockin'in the free world: a multi-national comparison of smartphone locking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2016.

[27] Marian Harbach, Emanuel Von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, 2014.

[28] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. Casa: context-aware scalable authentication. In *Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS)*. ACM, 2013.

[29] Eiji Hayashi, Oriana Riva, Karin Strauss, AJ Brush, and Stuart Schechter. Goldilocks and the two mobile devices: going beyond all-or-nothing access to a device's applications. In *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS)*. ACM, 2012.

[30] Nexus help. Delete, switch or add users. https://support.google.com/nexus/answer/2865483?hl=en. Accessed: 2021-08-30.

[31] Vincent C Hu, David Ferraiolo, D Richard Kuhn, et al. *Assessment of access control systems*. Citeseer, 2006.

[32] Jun Ho Huh, Hyoungshick Kim, Rakesh B Bobba, Masooda N Bashir, and Konstantin Beznosov. On the memorability of system-generated pins: Can chunking help? In *Eleventh Symposium On Usable Privacy and Security (SOUPS)*. USENIX Association, 2015.

[33] Jun Ho Huh, Saurabh Verma, Swathi Sri V Rayala, Rakesh B Bobba, Konstantin Beznosov, and Hyoungshick Kim. I don't use apple pay because it's less secure...: perception of security and usability in mobile tap-and-pay. In *Proceedings of Workshop on Usable Security*. Internet Society, 2017.

[34] Maia Jacobs, Henriette Cramer, and Louise Barkhuus. Caring about sharing: Couples' practices in single user device access. In *Proceedings of the 19th International Conference on Supporting Group Work (CSCW)*, 2016.

[35] Amy K Karlson, AJ Bernheim Brush, and Stuart Schechter. Can i borrow your phone? understanding concerns when sharing mobile phones. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2009.

[36] Amy K Karlson, Shamsi T Iqbal, Brian Meyers, Gonzalo Ramos, Kathy Lee, and John C Tang. Mobile taskflow in context: a screenshot study of smartphone usage. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2010.

[37] Hassan Khan, Aaron Atwater, and Urs Hengartner. Itus: an implicit authentication framework for android. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014.

[38] Masoud Mehrabi Koushki, Borke Obada-Obieh, Jun Ho Huh, and Konstantin Beznosov. On smartphone users' difficulty with understanding implicit authentication. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2021.

[39] Yunxin Liu, Ahmad Rahmati, Yuanhe Huang, Hyukjae Jang, Lin Zhong, Yongguang Zhang, and Shensheng Zhang. xshare: supporting impromptu sharing of mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009.

[40] Ahmed Mahfouz, Ildar Muslukhov, and Konstantin Beznosov. Android users in the wild: Their authentication and usage behavior. *Pervasive and Mobile Computing*, 32:50–61, 2016.

[41] Philipp Markert, Daniel V Bailey, Maximilian Golla, Markus Dürmuth, and Adam J AviG. This pin can be easily guessed: Analyzing the security of smartphone unlock pins. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020.

[42] Diogo Marques, Tiago Guerreiro, Luís Carriço, Ivan Beschastnikh, and Konstantin Beznosov. Vulnerability & blame: Making sense of unauthorized access to smartphones. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[43] Diogo Marques, Ildar Muslukhov, Tiago Guerreiro, Luís Carriço, and Konstantin Beznosov. Snooping on mobile phones: Prevalence and trends. In *Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, 2016.

[44] Tara Matthews, Kerwell Liao, Anna Turner, Marianne Berkovich, Robert Reeder, and Sunny Consolvo. " she'll just grab any device that's closer" a study of everyday device & account sharing in households. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2016.

[45] Michelle L Mazurek, JP Arsenault, Joanna Bresee, Nitin Gupta, Iulia Ion, Christina Johns, Daniel Lee, Yuan Liang, Jenny Olsen, Brandon Salmon, et al. Access control for home data sharing: Attitudes, needs and practices. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2010.

[46] Masoud Mehrabi Koushki, Borke Obada-Obieh, Jun Ho Huh, and Konstantin Beznosov. Is implicit authentication on smartphones really popular? on android users' perception of "smart lock for android". In *Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*. ACM, 2020.

[47] William Melicher, Darya Kurilova, Sean M Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. Usability and security of text passwords on mobile devices. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2016.

[48] Markus Miettinen, Stephan Heuser, Wiebke Kronz, Ahmad-Reza Sadeghi, and N Asokan. Conxsense: automated context classification for context-aware access control. In *Proceedings of the 9th ACM symposium on Information, computer and Communications Security*. ACM, 2014.

[49] Arsalan Mosenia, Susmita Sur-Kolay, Anand Raghunathan, and Niraj K Jha. Caba: Continuous authentication based on bioaura. *IEEE Transactions on Computers*, 66(5):759–772, 2017.

[50] Ildar Muslukhov, Yazan Boshmaf, Cynthia Kuo, Jonathan Lester, and Konstantin Beznosov. Know your enemy: the risk of unauthorized access in smartphones by insiders. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI)*, 2013.

[51] Xudong Ni, Zhimin Yang, Xiaole Bai, Adam C Champion, and Dong Xuan. Diffuser: Differentiated user access control on smartphones. In *Proceedings of the 6th International Conference on Mobile Adhoc and Sensor Systems*. IEEE, 2009.

[52] Don Norman. *The design of everyday things: Revised and expanded edition*. Constellation, 2013.

[53] Borke Obada-Obieh, Yue Huang, and Konstantin Beznosov. Challenges and threats of mass telecommuting: A qualitative study of workers. In *Proceedings of the Seventeenth Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, 2021.

[54] Google Support Pages. Google camera help. https://support.google.com/googlecamera/answer/6164997?hl=en. Accessed: 2021-08-30.

[55] Eyal Peer, Laura Brandimarte, Sonam Samat, and Alessandro Acquisti. Beyond the turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70:153–163, 2017.

[56] Android Developers Portal. Usage stats manager. https://developer.android.com/reference/android/app/usage/UsageStatsManager. Accessed: 2021-08-06.

[57] Lina Qiu, Alexander De Luca, Ildar Muslukhov, and Konstantin Beznosov. Towards understanding the link between age and smartphone authentication. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[58] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. How well do my results generalize? comparing security and privacy survey results from mturk, web, and telephone samples. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[59] Oriana Riva, Chuan Qin, Karin Strauss, and Dimitrios Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In *Proceedings of the USENIX Security Symposium*. USENIX Association, 2012.

[60] Nithya Sambasivan, Garen Checkley, Amna Batool, Nova Ahmed, David Nemer, Laura Sanely Gaytán-Lugo, Tara Matthews, Sunny Consolvo, and Elizabeth Churchill. " privacy is not for me, it's for those rich women": Performative privacy practices on mobile phones by women in south asia. In *Proceedings of the Fourteenth Symposium on Usable Privacy and Security (SOUPS)*, 2018.

[61] Julian Seifert, Alexander De Luca, Bettina Conradi, and Heinrich Hussmann. Treasurephone: Context-sensitive user data protection on mobile phones. In *Proceedings of the International Conference on Pervasive Computing*. Springer, 2010.

[62] Timothy Sohn, Kevin A Li, William G Griswold, and James D Hollan. A diary study of mobile information needs. In *Proceedings of the CHI conference on human factors in computing systems*. ACM, 2008.

[63] Stata. Global smartphone penetration rate as share of population from 2016 to 2020. https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/. Accessed: 2021-09-01.

[64] Apple support. User guided access with iphone. https://support.apple.com/en-us/HT202612. Accessed: 2021-08-13.

[65] Google support. Pin & unpin screens on android. https://support.google.com/android/answer/9455138?hl=en. Accessed: 2021-08-13.

[66] Samsung Support. What is the secure folder and how do i use it? https://www.samsung.com/uk/support/mobile-devices/what-is-the-secure-folder-and-how-do-i-use-it/. Accessed: 2021-09-14.

[67] Nanda Kumar Thanigaivelan, Ethiopia Nigussie, Antti Hakkala, Seppo Virtanen, and Jouni Isoaho. Codra: Context-based dynamically reconfigurable access control system for android. *Journal of Network and Computer Applications*, 101:1–17, 2018.

[68] Ding Wang, Qianchen Gu, Xinyi Huang, and Ping Wang. Understanding human-chosen pins: characteristics, distribution and security. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security (AsiaCCS)*. ACM, 2017.

[69] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proceedings of the ACM SIGSAC conference on computer and communications security (CHI)*. ACM, 2016.

[70] Xiaofeng Wang and Zhenshun Cheng. Cross-sectional studies: strengths, weaknesses, and recommendations. *Chest*, 158(1):S65–S71, 2020.

[71] Xueqiang Wang, Kun Sun, Yuewu Wang, and Jiwu Jing. Deepdroid: Dynamically enforcing enterprise policy on android devices. In *Proceedings of the Network and Distributed System Security (NDSS) Symposium*, 2015.

A Appendix

A.1 Description of Pilot Studies

We performed three pilot studies to test our methodology and app design. The first study involved lab testing sessions and qualitative interviews to evaluate the app's usability. We used our university's study-participant mailing list to recruit six participants (three of them male, ages 23-65). They performed a series of tasks with our app (e.g., filling out a daily diary) while we observed if they encountered any difficulties. We also interviewed them about their impressions of the app. Based on the results, we improved the app setup process and the wordings of some diary questions, and fixed a few bugs. For the second pilot study, we used the same mailing list again to recruit 7 participants (5 of them females, ages 27–37). They installed the app on their personal phones and used it to fill out diaries for two weeks. Then we interviewed them about their experience. Based on the results, we found it would be optimal if the diaries did not take more than 10 minutes per day. We also further improved the wordings of some questions and fixed some more bugs. More importantly, we found a lot of duplication of task declarations, which took unnecessary time from participants. To reduce fatigue, we implemented a system where each participant could see, anonymously, what tasks others had declared for each app. This way they could either select one of those tasks or declare a new one.

Interviews also showed that some participants might have privacy concerns with installing our app on their personal devices. To alleviate such concerns, we made it clearer in our study advertisement and consent form that the app would not collect any personal data, such as location or log-in credentials. We also invited participants to ask any questions they might have about the app or the purpose of the study, and let them know they can withdraw from the study at any time without repercussions. As a result of these changes, we observed in our screening surveys that very few potential participants of the main study declared privacy concerns as a reason for not joining the study.

The third pilot study was aimed at evaluating the efficacy of the intended recruitment channel (MTurk) and reverifying the study procedures. Specifically, we wanted to see if showing tasks defined by one participant to others would result in reduced diversity of declared tasks and eventually bias in the results. We recruited 8 participants from MTurk (4 of them female, ages 21–44) and asked them to install and use the app for 2 to 3 weeks. We found MTurk to be able to provide us with sufficiently diverse samples. We also observed being near theoretical saturation for task declarations after three weeks (i.e., very few new tasks were being declared). Hence we decided on one month as the length of the main study. Lastly, study results showed no signs of reduced task diversity; instead, an even more diverse set of tasks was defined compared to the second study.

A.2 Additional Methodology Details and Data Analysis Results

Table A.1: Results of chi-squared tests between access-control categorizations and our anticipated antecedents.

	Public or (Semi)Private	Whether They Share Phone
Age	p = 0.002	p = 0.048
	V = 0.433	V = 0.379
Education	p = 0.293	p = 0.367
	V = 0.258	V = 0.240
Hours of phone usage per day	p = 0.350	p = 0.967
	V = 0.201	V = 0.035
Privacy app adoption	p = 0.654	p = 0.279
	V = 0.149	V = 0.215
Living with others	p = 0.577	p = 0.158
	V = 0.141	V = 0.190

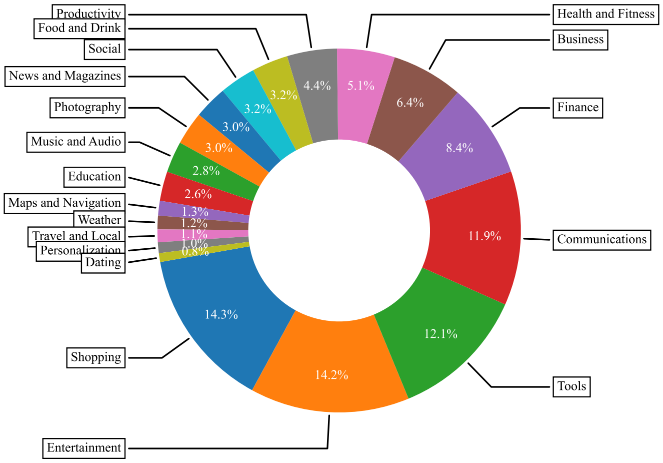

Figure A.1: Distribution of participants' declared tasks across functionality categories.

Table A.2: Formulas used to compute FPR, FNR, and CSR of the scenarios. CSR formulas assume a default "No one" policy. Set of "No one"/"Specific people"/"Anyone"/all tasks are denoted by NN/SP/AO/T. SPC indicates SP excluding current sharee. CA/MA indicate set of apps available to anyone, based on CONSERVATIVE/MAXIMAL strategy. FC/FM(t) indicate final CONSERVATIVE/MAXIMAL label of task t. X(Y) indicates sub-setting X based on Y (e.g., NN(MA) indicates NN tasks that are afforded only by apps in MA). As for strictness, NN is considered stricter than SP, which is stricter than AO.

Scenario		FPR	FNR	CSR
Task sensitivity	ALL	$\frac{\lvert NN\rvert+\lvert SP\rvert}{\lvert T\rvert}$	$\frac{0}{\lvert T\rvert}$	$\frac{0}{\lvert T\rvert}$
	NOTHING	$\frac{0}{\lvert T\rvert}$	$\frac{\lvert SP\rvert+\lvert AO\rvert}{\lvert T\rvert}$	$\frac{1}{\lvert T\rvert}$
	LOCK_SCREEN_ACCESS	$\frac{\lvert NN(Lock\_screen\_apps)\rvert}{\lvert T\rvert}$	$\frac{\lvert SP\rvert+\lvert AO\rvert-\lvert T(Lock\_screen\_apps)\rvert}{\lvert T\rvert}$	$\frac{0}{\lvert T\rvert}$
	APP_CONSERVATIVE	$\frac{\lvert NN(CA)\rvert}{\lvert T\rvert}$	$\frac{\lvert AO\rvert+\lvert SP(CA)\rvert}{\lvert T\rvert}$	$\frac{\lvert CA\rvert}{\lvert T\rvert}$
	APP_MAXIMAL	$\frac{\lvert NN(MA)\rvert}{\lvert T\rvert}$	$\frac{\lvert AO\rvert+\lvert SP(MA)\rvert}{\lvert T\rvert}$	$\frac{\lvert MA\rvert}{\lvert T\rvert}$
	TASK_CONSERVATIVE	$\frac{\sum_{t\in T}\lvert Stricter\_labels\_than\_FC(t)\rvert}{\lvert T\rvert}$	$\frac{\sum_{t\in T}\lvert Laxer\_labels\_than\_FC(t)\rvert}{\lvert T\rvert}$	$\frac{\lvert SP\rvert+\lvert AO\rvert}{\lvert T\rvert}$
	TASK_MAXIMAL	$\frac{\sum_{t\in T}\lvert Stricter\_labels\_than\_FM(t)\rvert}{\lvert T\rvert}$	$\frac{\sum_{t\in T}\lvert Laxer\_labels\_than\_FM(t)\rvert}{\lvert T\rvert}$	$\frac{\lvert SP\rvert+\lvert AO\rvert}{\lvert T\rvert}$
Phone sharing	ALL_OR_NOTHING	$\frac{\sum_{t\in T}\lvert NN(t)\rvert+\lvert SPC(t)\rvert}{\lvert T\rvert}$	$\frac{0}{\lvert T\rvert}$	$\frac{1}{\lvert T\rvert}$
	PROFILE_SWITCHING	$\frac{0}{\lvert T\rvert}$	$\frac{0}{\lvert T\rvert}$	$\frac{\lvert Sharees\rvert+\lvert Apps\_shared\_w\_each\_sharee\rvert}{\lvert T\rvert}$
	DIFF_USER	$\frac{\sum_{t\in T(Apps\_w/o\_sensitive\_permissions)}\lvert NN(t)\rvert}{\lvert T\rvert}$	$\frac{\sum_{t\in T(Apps\_w\_sensitive\_perms)}\lvert SP(t)\rvert+\lvert AO(t)\rvert}{\lvert T\rvert}$	$\frac{\lvert Sharees\rvert}{\lvert T\rvert}$
	X_SHARE	$\frac{\sum_{t\in T(Shared\_apps)}\lvert NN(t)\rvert+\lvert SPC(t)\rvert}{\lvert T\rvert}$	$\frac{0}{\lvert T\rvert}$	$\frac{\lvert Shared\_apps\rvert+1}{\lvert T\rvert}$