SnapApp: Reducing Authentication Overhead with a Time-Constrained Fast Unlock Option

Daniel Buschek, Fabian Hartmann, Emanuel von Zezschwitz, Alexander De Luca, Florian Alt
University of Munich (LMU), Germany
{daniel.buschek, emanuel.von.zezschwitz, alexander.de.luca, florian.alt}@ifi.lmu.de,
mail@fabian-hartmann.de

ABSTRACT

We present *SnapApp*, a novel unlock concept for mobile devices that reduces authentication overhead with a time-constrained quick-access option. *SnapApp* provides two unlock methods at once: While PIN entry enables full access to the device, users can also bypass authentication with a short sliding gesture ("*Snap*"). This grants access for a limited amount of time (e.g. 30 seconds). The device then automatically locks itself upon expiration. Our concept further explores limiting the possible number of *Snaps* in a row, and configuring blacklists for app use during short access (e.g. to exclude banking apps). We discuss opportunities and challenges of this concept based on a 30-day field study with 18 participants, including data logging and experience sampling methods. *Snaps* significantly reduced unlock times, and our app was perceived to offer a good tradeoff. Conceptual challenges include, for example, supporting users in configuring their blacklists.

Author Keywords

Smartphone authentication; Usable privacy and security; Time-constrained device access

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Recent research has shown that users spend considerable amounts of time on unlocking their smartphones, although their interactions after unlocking can be quite short [12]: Half of all phone interactions last less than 30 seconds, and 90% last less than four minutes [27]. For example, a user might just briefly check for new messages, view a public transport app or read some news. Many so-called "micro-usages" even last less than 15 seconds [9], which adds up to serious authentication overhead, when considering reported unlock times for PIN (≈4.7s) and Android unlock pattern (≈3s) [12].

CHI'16, May 07–12, 2016, San Jose, CA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3362-7/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858164

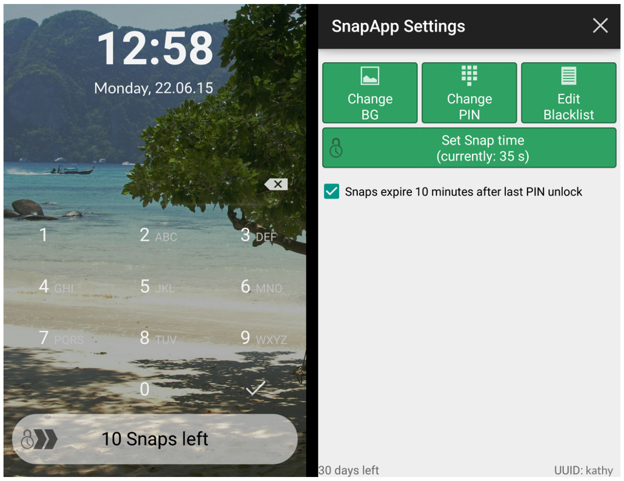
12:58
Monday, 22.06.15
10 Snaps left
SnapApp Settings
Set Snap time (currently: 35 s)
☑ Snaps expire 10 minutes after last PIN unlock
30 days left
UUID: kathy

Figure 1. *SnapApp's* lockscreen (left) and settings (right). Users can either enter their PIN for unconstrained access, or perform a left-to-right swipe at the bottom (i.e. "*Snap*"). In this case, they gain time-constrained access, for example limited to 30 seconds. After that time has passed, the phone automatically locks itself. During short access sessions, launching apps from a user-defined blacklist will lock the device immediately.

We argue that a fast, effortless unlock option may save time during short interactions, by granting access to the device for a limited amount of time without the need for a secure unlock. We assume that, during such short interactions of less than 30 seconds, many attacks are impossible to perform, and the device should lock immediately if critical actions occur (e.g. launching a banking app during short access).

This paper investigates such a concept for time-constrained access control on mobile devices, called *SnapApp* (Figure 1). We discuss findings from a 30-day field study. *Snaps* significantly reduced unlock times, but participants perceived it as slower than their usual methods. Short access was mostly used for insensitive content in uncritical contexts. It was often used for messaging, and was perceived as more secure than only using swipe, although few users configured blacklists. The blacklist is a user-defined list of apps, which cannot be opened during short access, only during full access (i.e. with PIN). Trying to open a blacklisted app during short access locks the device immediately.

We contribute: 1) a concept for a time-constrained fast unlock option on mobile devices, 2) evaluated in a 30-day field study with 18 users, 3) leading to novel insights into opportunities and challenges of such a concept, and implications for future investigation and design of time-constrained unlock methods.

RELATED WORK

Despite being considered convenient and easy to use [25], current smartphone authentication systems like PIN and the Android unlock pattern are susceptible to manifold threats: 1) Users often choose simple authentication secrets (e.g. PINs based on birthdates) that are easy to guess [1, 22]. 2) In the most common setups, for instance if line visibility for the Android unlock pattern is turned on, the patterns can easily be shoulder surfed [24]. 3) Interaction with the smartphone leaves traces like smudge stains that give away the secret [2].

Due to these vulnerabilities, many ongoing research efforts aim to provide more secure forms of smartphone authentication. Examples include additional biometric security layers on top of usual authentication systems [5, 7, 28], multitouch authentication [21], additional (invisible) channels [3] or enhanced graphical authentication [16, 19, 26]. These are all-or-nothing mechanisms: Users either have access to all data and services on their devices or to none at all.

However, research on mobile device access and sharing indicates that such all-or-nothing access control does not fit the users' needs [11, 14, 15]. Hang et al. [11] found that participants would allow unprotected access to complete apps or parts of specific apps. For instance, reading Facebook posts on a locked device could be ok, while writing posts is not. Hayashi et al. [14] even found that users would like to have around half or their applications available even if their device was locked. They also tested different finer-grained control mechanisms and received positive user feedback.

In addition, recent research on real world smartphone use and locking risks by Harbach et al. [12] showed that users only seldomly access sensitive data on their devices and that there were few instances of situations in which their mobile devices were at risk. Related to this, Egelman et al. [8] report that in many cases, users lack motivation to lock and protect their devices as they consider the risk rather low. Both papers advocate to minimize the need for authentication, e.g. by allowing access to specific functionality without locking, in order to lower the burden for the users.

One proposed approach is location or context-sensitive authentication or data access [10, 13, 17, 20]: Here, the device's security level is automatically adapted based on inferred context, like a certain location or being with a specific group of people. For instance, consider contexts like "home" or the information that the device was not handed to or taken by another person. In these cases, authentication may not be required at all, or an easier (more convenient) form of authentication could be provided to the users.

However, this could potentially open new security holes, since home or other private contexts are not necessarily safe places (e.g. due to insider threat [18]). Thus, more recent work has introduced authentication systems that are easy to use and can be switched to a secure mode by the user if improved security is needed [23]. On the downside, such methods still follow an all-or-nothing approach: user have to authenticate or they cannot access anything.

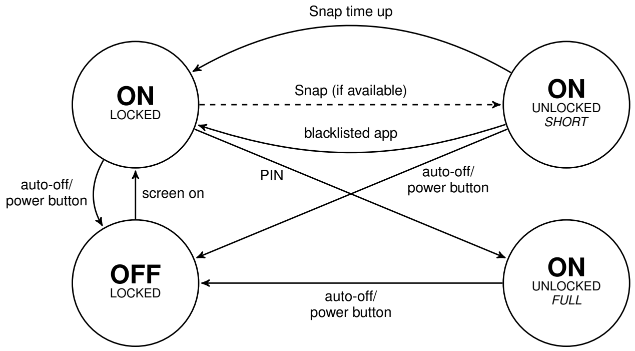

Figure 2. States and transitions of the *SnapApp* concept.

Approaches in industry include, for example, a number of Android apps (e.g. *AppLock*[1], *CM Security Antivirus Applock*[2]) which enable locks per app, meaning that a PIN, pattern or password has to be entered when launching that app. Furthermore, the Android app *ProtectedApps*[3] allows users to set a timer, that starts after closing an app, and then locks that app if it runs out so it cannot be started again without unlocking. In contrast, SnapApp's timer starts after unlocking the device, and locks the *whole device* upon expiration.

This paper investigates a novel alternative to reduce authentication overhead. While no previous approach showed two methods on the device unlock screen, the key aspect of our concept is to present two unlock methods at once: 1) PIN as a representative for secure access, and 2) a simple sliding gesture for fast access. Interaction after fast access is constrained by a maximum duration and by a blacklist that excludes user-defined apps from being used without PIN entry.

SNAPAPP CONCEPT

The core concept of *SnapApp* is to provide two unlock methods at once (Figure 1) to reduce authentication overhead for short usage sessions. In contrast to current unlock methods, our concept thus provides users with *two* trade-offs regarding authentication time versus actual use: either spending more time on authentication for full access, or quickly bypassing authentication for limited access. In particular, users decide how to unlock based on their expected interaction time:

- *PIN entry* gives full access to the device, as usual.
- A *fast sliding gesture* grants time-constrained access to a user-defined subset of applications.

Short access via slide (i.e. *Snap*) limits access to a user-specified amount of time (*Snap time*). The device locks itself after this time has passed. Expiration is announced via short vibration, issued at five seconds prior to the session's end.

The system allows ten *Snaps* in a row without entering the PIN. The current number of *Snaps* is displayed on the *Snap* bar – see Figure 1 ("10 Snaps left"). When all *Snaps* have been used, unlock by sliding cannot be used any more. Successful PIN entry sets the number of *Snaps* back to ten.

[1] https://play.google.com/store/apps/details?id=com.domobile.applock

[2] https://play.google.com/store/apps/details?id=com.cleanmaster.security

[3] http://protectedapps.apk.defim.de/ - *all last accessed 7th Jan. 2016*

We also use *Snap deactivation*: Available *Snaps* expire automatically after ten minutes since the last PIN unlock, which allows users to chain a reasonable amount of *Snaps* in many typical situations that require frequent short accesses, such as keeping up a conversation via messenger apps. Users can turn the *Snap deactivation* feature off in the settings (Figure 1).

Users can optionally exclude certain apps from being used without entering a PIN, using a configurable *blacklist*. If a blacklisted app is launched during a short access session, the device locks itself immediately. This also resets the available number of *Snaps* to zero. Thus, PIN entry is always required after trying to launch a blacklisted app.

Finally, if an ongoing phone call takes longer than the *Snap time*, the device locks itself only after the call or immediately if the user switched to a blacklisted app during the call.

To give an overview, Figure 2 visualises the states and transitions that result from these concepts.

SNAPAPP SECURITY

At first glance, unprotected access potentially opens the device to attacks. To counter these, the system contains several features (mentioned before) that reduce this risk: To keep potential attackers from simply using one *Snap* after the other to emulate full access, the maximum number of consequent *Snaps* is limited to ten. After this number of *Snaps,* the user has to enter the PIN to re-authenticate with the device. Security is further increased by *Snap deactivation*, meaning automatic expiration of *Snaps* after ten minutes since the last PIN unlock. This is useful, for example, when the phone gets lost. Therefore, an attacker has a very limited timeframe to both steal and then use the device.

Even then, a blacklist provides additional protection: While system apps and the *SnapApp* settings are automatically blacklisted, users can select additional apps that they consider sensitive. If any blacklisted app is used during short access, the device locks itself immediately and all *Snaps* expire.

Finally, compared to common authentication, *SnapApp* reduces the number of cases where shoulder surfing can occur (i.e. others learning PIN, password, or pattern by spying on the user's input [24]). Users seldom access apps or data that they consider sensitive [12]. In consequence, this means that every time they want to perform an insensitive task, they still potentially give away sensitive information, namely their PIN, password or pattern. *SnapApp* minimises this risk.

USER STUDY

Study Design

We conducted a field study, logging configured settings and usage behaviour: user-defined *Snap time*, *Snap deactivation* on/off, number of sessions, app usage, chosen unlock method per session (i.e. short or full), blacklist usage, and answers to the experience sampling questionnaires. The study addressed the following research questions:

1. How well accepted is a time-constrained unlock option?
2. Do users adapt the method's parameters (*Snap time*, *Snap deactivation*) to their needs?
3. How do users configure their blacklists?
4. Do users' settings mirror their privacy/security concerns?
5. Are phones used differently during short access sessions compared to full access sessions?

Apparatus

We implemented the *SnapApp* concept as an Android app that replaced the user's normal startup-screen/lockscreen. Besides the concept's functionality, the app logged its use as well as general app usage on the device. Anonymous data was stored locally on the device, and then sent automatically to a secure server every 24 hours, given available WiFi access.

Participants

We distributed a preliminary survey via a university mailing list. It was completed by 240 people, out of which 195 owned devices with suitable Android versions. We invited all users of PIN, swipe-to-unlock and no lock to the main study. We received 50 complete pre-study questionnaires; 29 of those participants completed the field study and the post-study questionnaire. Excluding eleven cases with technical issues post-hoc (see results) resulted in 18 final participants. Their mean age was 24 (range: 19–64). Seven were female. Nine used PIN, four pattern, and five swipe or no lock mechanism. We offered a 20 EUR gift card or study credits as compensation.

Procedure

The study procedure included a preliminary survey, an initial questionnaire, 30 days of field study with passive data logging and active experience sampling, and a final questionnaire.

A preliminary survey assessed device model, Android version, and usual and current unlock method(s) of respondents to our study announcement. We invited PIN, pattern and swipe-to-unlock users to participate in our study.

At the beginning of the study, participants completed an initial questionnaire, which assessed a variety of information, such as demographics, privacy and security concerns, perceived smartphone usage, and used unlock methods.

Participants were then sent our app with detailed instructions on setup and use. This information was also accessible at any time via an accompanying website. Participants were also encouraged to contact us in case of unforeseen problems. The field study lasted for 30 days.

We collected feedback during app use via experience sampling, meaning that our app asked users questions *in situ* (see e.g. [6, 12]). We used two kinds of mini-questionnaires: The first was shown directly after unlocking (with a chance of 20%), assessing environment (e.g. home, bar), its perceived criticality, and perceived sensitivity of the data going to be accessed. The second questionnaire was shown upon expiration of a short access session (i.e. when the chosen short *Snap time* had likely been not sufficient for the current session), also with a chance of 20%. Both questionnaires were restricted to never appear more than once within one hour.

At the end, participants were contacted and given a final questionnaire, which repeated some of the questions from the first one. Additionally, it assessed subjective ratings of our concept, regarding usability and privacy/security aspects.

user	PIN	Snap	% Snap	time up locks	blacklist locks	blacklist apps	usual method
#1	447	1,043	70.00%	7.57%	0.77%	0	pattern
#2	751	646	46.24%	31.26%	10.22%	0	PIN
#3	599	389	39.37%	2.06%	5.91%	0	swipe
#4	392	228	36.77%	21.05%	0.44%	0	PIN
#5	41	22	34.92%	27.27%	45.45%	0	pattern
#6	1,010	525	34.20%	14.10%	9.52%	14	swipe
#7	670	296	30.64%	13.51%	2.36%	0	swipe
#8	1,641	250	13.22%	46.40%	2.40%	1	pattern
#9	1,081	157	12.68%	8.28%	12.10%	0	PIN
#10	1,555	217	12.25%	27.65%	2.76%	0	pattern
#11	554	68	10.93%	33.82%	35.29%	2	PIN
#12	1,626	142	8.03%	56.33%	1.41%	0	PIN
#13	1,041	74	6.64%	54.05%	2.70%	0	PIN
#14	468	33	6.59%	6.06%	36.36%	10	PIN
#15	1,237	75	5.72%	33.33%	1.33%	0	PIN
#16	743	43	5.47%	30.23%	44.19%	64	swipe
#17	2,484	28	1.11%	67.86%	0.00%	0	PIN
#18	125	0	0.00%	-	-	0	swipe
Total	**16,465**	**4,236**	**20.82%**	**28.29%**	**12.54%**		

Table 1. Summary of participants' sessions and used unlock methods. From left to right: Number of PIN unlocks and *Snaps*, the resulting ratio of *Snaps*, ratio of locks caused by expired *Snap* time and by blacklist violations, number of blacklisted apps, and the participant's normally used unlock method. The last row shows aggregated results.

RESULTS

We logged 247,919 data points (locks/unlocks, blacklist changes, experience sampling, app use). We removed 12,187 (4.92%) entries in the clean-up process described below. The number of entries per user ranged from 653 to 33,915.

Data Preprocessing

Our app uses a background service to monitor blacklist violations. For the purpose of the study, the service was also used for data logging. Unfortunately, some devices kill background services when users open many apps in parallel and memory runs low. This causes our app to restart, locking the phone in the process. We displayed a permanent status message (showing the remaining number of study days) in the notification bar, to force Android to give higher priority to our service. However, the final data contained many such forced restarts. This was revealed post-hoc by examining the unlock-to-lock ratio in the logfiles (ideally 1:1, i.e. 100%), which showed that 11 participants had a ratio of less than 90%, indicating many restarts. To not distort results, we excluded these participants from the analyses, and removed incomplete unlock sessions for the remaining participants.

Unlock Times

We measured time from the *screen on* event until the *unlock* event. We removed unlock times longer than 45 seconds as outliers. This threshold was chosen as about ten times the average duration for PIN entry reported in related work [12].

Access by *Snap* took 3.47 seconds on average ($sd = 1.70s$, $min = 2.38s$, $max = 7.88s$). PIN required 4.68 seconds ($sd = 1.14s$, $min = 3.16s$, $max = 7.50s$). The difference between *Snap* and PIN was significant (paired t-test, $t(16) = 2.278$, $p < 0.05$; here, $df = 16$ since for the test we had to exclude the one user who never used *Snaps*). If these times may seem long, note that users in the field might be interrupted in between turning on the screen and unlocking. This does not bias the results, as external interruptions are independent of the unlock method. We removed extreme outliers (see above).

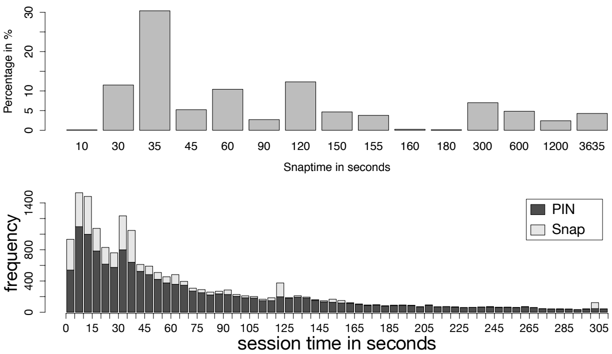

Figure 3. Distribution of *Snap* times as configured by participants (top), and distribution of unlock methods for occurring session durations (bottom). Together, these figures show that participants configured and used access by *Snap* primarily as intended, namely for short sessions.

Session Types and Durations

During the 30 days of the study, the 18 participants activated their phones 36,520 times in total. Unlocks happened in 20,701 cases – we call these cases "sessions". There are more activations than sessions, since participants could view information without unlocking, such as the time, or the Android status bar, for example to check for new notifications. Table 1 summarises these results, described in the following:

Per participant and day, we logged an average of 71.31 activations ($sd = 34.85$, $min = 4.17$, $max = 148.70$) and 40.35 sessions ($sd = 20.11$, $min = 2.10$, $max = 148.70$). Access by *Snap* was used in 4,236 sessions. This is an average of 20.82% of all sessions per participant ($sd = 19.23\%$, $min = 0\%$, $max = 70\%$).

On average, short access sessions (i.e. sessions unlocked with time-constrained access via *Snap*) lasted 50.49 seconds ($sd = 70.09s$, $min = 0.11s$, $max = 1174.00s$), while full access sessions (i.e. sessions unlocked without time limit via PIN) lasted 198.00 seconds ($sd = 429.70s$, $min = 0.62s$, $max = 9116.00s$). Figure 3 (bottom plot) shows observed session times and chosen access methods: Peaks occur for times correlating with commonly chosen *Snap times*, for example at 35s, 60s, 120s and 300s, since sessions are terminated (i.e. phone locked) when *Snap time* runs out.

Snap Times and Snap Resets

Our app allowed participants to change the maximum short access time (*Snap time*) via a settings screen as often as they liked during the study. Our 18 participants chose 15 different *Snap time* values, ranging from 10 to 3,635 seconds (1 hour 35 seconds). The default was 35 seconds. 13 participants did not initially change the default setting, four of them neither changed it later. Five participants changed the setting once, another five twice, and three users adjusted it four times.

Figure 3 (top) shows the distribution of these values for all 4,236 logged short access sessions of 17 users. One user did not use short access. This results in 30.37% for the most used *Snap time*, the default setting of 35 seconds. Other commonly chosen *Snap times* were: 120 seconds (12.30% occurrence), 30s (11.50%), and 60s (10.42%), followed by 300s (7.01%) and 45s (5.23%). All other times are below the 5% mark.

Besides *Snap time*, our app allowed users to turn the so-called *Snap deactivation* on or off: By default, available *Snaps* expired automatically after ten minutes since the last full PIN unlock. Four of 18 users disabled *Snap deactivation* (Figure 1 right, checkbox), mostly within the first half of the study.

Blacklist Usage

Our concept uses a blacklist of apps that can never be opened without a PIN. Any attempt to launch a blacklisted app during a short access session locks the device immediately. The blacklist contains at least the Android system settings and the *SnapApp* settings. This cannot be changed for security reasons (e.g. to prevent uninstalling *SnapApp* during short access), but people were free to add and remove any other installed apps to and from the blacklist via a settings screen as often as they liked during the study.

Table 1 shows that 16 participants faced blacklist locks due to the two settings apps excluded by default. User 5 had the highest ratio of blacklist locks. This user tried to open *SnapApp* during short access several times (which is forbidden to avoid that attackers increase *Snap time*). Half of these cases occurred in the first 30 minutes after installing the app.

Five users actively configured their blacklists:

- Participant #8 blacklisted WhatsApp only, and changed the *Snap time* twice, to ten seconds at last.
- Participant #11 blacklisted e-mail and contact apps, and changed the *Snap time* once to 45 seconds.
- Participant #6 blacklisted ten apps with accounts, including payment apps and games (e.g. PayPal, Netflix, Runtastic), without changing the default *Snap time* of 35 seconds.
- Participant #14 appropriated the *SnapApp* concept to effectively implement "app-based authentication", by increasing *Snap time* to over 2 hours, combined with blacklisting messengers, e-mail client, firewall, and settings and developer tools with extended system privileges.
- Participant #16 implemented a whitelist approach, adding all apps to the list first, before removing only four apps (camera, browser, PDF viewer, Google Hangouts), without changing the default *Snap time*.

The configurations of three of these participants (#11, #14, #16) resulted in more forced short access session endings caused by blacklist violations, namely 35.29%, 36.36% and 44.19% – compared to the total average of 12.54%.

Application Usage

We logged application usage: Table 2 summarises the apps used during short and full access sessions. Categories are based on those defined by each app's entry in the Google Play Store, but were manually adapted to be more meaningful (e.g. creating a *Browser* category), following related work [4].

Messaging was in the two most frequently used categories for both access methods. However, messengers had a clearer lead during short access, with 34.87% and a gap of 15.55% to the next category. *E-mail* apps were launched less often during short access than full access (4.51% vs 7.12%).

Short Access		Full Access	
Messaging	34.87%	Other	26.08%
Other	19.32%	Messaging	25.88%
Phone calls	12.76%	Phone calls	11.19%
Web browsing	7.66%	Web browsing	7.71%
E-mail	4.51%	E-mail	7.12%
Music	4.49%	Games	4.84%
Games	3.94%	News / magazines	4.36%
Photography	3.06%	Photography	3.59%
News / magazines	3.04%	Social networks	3.08%
Social networks	2.84%	Music	2.03%
Journey planner	1.27%	Videos	1.70%
Shopping	1.26%	Shopping	1.14%
Videos	0.51%	Maps / navigation	0.75%
Maps / navigation	0.47%	Journey planner	0.53%

Table 2. Categories of apps used in short and full access sessions.

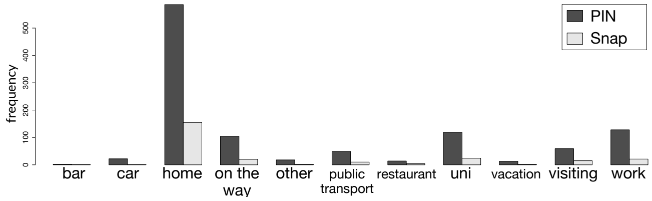

Figure 4. Distribution of short and full access unlocks used in different environments. They were reported by participants in the experience sampling questionnaire shown directly after unlocking.

Since *Other* contains the most diverse apps, we analysed it in more detail. The most used apps in this category during short access were: Google App (7.70%), clock (3.07%), calendar (1.86%) and Google Play Store (1.29%). Most used *Other* apps during full access were: Google App (9.57%), system settings (3.80%), Google Play Store (2.61%), clock (2.20%) and calendar (0.84%). *Games* were used more during full access (4.84%) than short access (3.94%).

Phone calls (short: 12.76%, full: 11.19%) and *Web browsing* (short: 7.66%, full: 7.71%) were used almost equally with both access methods, similar to *Photography* (short: 3.06%, full: 3.59%) and *Shopping* (short: 1.26%, full: 1.14%). Note that phone calls were not cancelled, if they took longer than the *Snap time* – the device locked itself after the call or if the user switched to a blacklisted app during the call.

We also took a closer look at *Photography* apps: During short access, the camera (2.05%) was used more often than the gallery (1.01%). In contrast, during full access, use of the gallery (2.01%) exceeded use of the camera (1.59%).

Experience Sampling

To collect feedback and subjective data during use, our app showed two types of mini questionnaires (see procedure section): One was shown directly after unlocking, the other upon expiration of a short access session.

In total, participants answered 1,455 questionnaires, while 1,509 were skipped via pressing a "Not now" button. The first questionnaire was randomly displayed 2,841 times after unlocks (1,369 not dismissed). On average, each user completed 76.06 such questionnaires (*sd* = 45.66, *min* = 3, *max* = 179). The second questionnaire was displayed 123 times after *Snap time* expired. With 86 completions, participants on average replied 4.78 times (*sd* = 6.15, *min* = 0, *max* = 27).

Unlock Environments

Figure 4 shows the distribution of unlocks in different reported environments for both access methods: *Home* was most common with 741 occurrences, followed by *work* (149), *university* (143), *on the way* (124), *visiting* (74) and *public transport* (59). Remaining locations were *car* (23), *other* (20), *restaurant* (18), *vacation* (15) and *bar* (3). The ratio of short access per location varied from 4.35% (*car*) to 33.33% (*bar*) with an average of 17.13%. Locations of the *other* category were not predefined but rather entered by the participants. This included, for example, the movies or a theater.

Criticality of Environments

We also asked participants how "critical" their current environment was. In the context of the study, "critical" refers to potential privacy/security issues. Table 3 summarises the responses, revealing participants' diverging perceptions of the criticality of their surroundings.

Ratings of the current environment for all sessions in descending order were: 50% uncritical (695), 15.56% rather uncritical (213), 13.73% rather critical (188), 13.15% neutral (180), and 6.79% critical (93).

For brevity, the following percentages combine the two non-neutral ratings on each side. Rather private environments were rated less critical: *home* (92% uncritical, 6% neutral, 2% critical), *visiting* (65% uncritical, 23% neutral, 12% critical) and *car* (39% uncritical, 52% neutral, 9% critical).

On the other hand, public places were rated more critical, as expected: *restaurant* (33% uncritical, 6% neutral, 61% critical), *bar* (0% uncritical, 0% neutral, 100% critical) and *public transport* (15% uncritical, 12% neutral, 73% critical).

Ratings diverged for other environments: *vacation* (20% uncritical, 67% neutral, 13% critical), *on the way* (39% uncritical, 23% neutral, 38% critical), *work* (32% uncritical, 21% neutral, 47% critical) and *university* (40% uncritical, 14% neutral, 46% critical). University/work were likely rated critical by some because there are more people present (e.g. concerns of theft or privacy, like someone sitting behind the user).

Sensitivity of Accessed Data

The experience sampling questionnaire, shown directly after unlocking, also asked participants to rate the sensitivity of the data that they are about to access. Participants rated: 35.14% neutral (481), 32.51% insensitive (445), 19.28% rather sensitive (264), 7.82% rather insensitive (107) and 5.25% sensitive (72). These results are interesting in combination with the perceived criticality of the environment. Table 4 summarises environment ratings and associated data ratings.

We found that in 63.99% of all unlock sessions, insensitive to neutral data was accessed within uncritical to neutral environments. In contrast, sensitive and rather sensitive data was accessed in rather critical to critical areas with a share of 9.06%.

Environment	Ratings by Participants				
	uncritical	**rather uncr.**	**neutral**	**rather cr.**	**critical**
bar	0 (0%)	0 (0%)	0 (0%)	3 (100%)	0 (0%)
car	7 (30%)	2 (9%)	12 (52%)	2 (9%)	0 (0%)
home	552 (74%)	127 (17%)	48 (6%)	12 (2%)	2 (0%)
on the way	40 (32%)	8 (6%)	29 (23%)	36 (29%)	11 (9%)
public transport	7 (12%)	2 (3%)	7 (12%)	28 (47%)	15 (25%)
restaurant	1 (6%)	5 (28%)	1 (6%)	9 (50%)	2 (11%)
university	45 (31%)	12 (8%)	20 (14%)	31 (22%)	35 (24%)
on vacation	3 (20%)	0 (0%)	10 (67%)	1 (7%)	1 (7%)
on a visit	20 (27%)	28 (38%)	17 (23%)	8 (11%)	1 (1%)
at work	19 (13%)	29 (19%)	31 (21%)	56 (38%)	14 (9%)

Table 3. Participants' perceived criticality of their environments reported via the experience sampling questionnaire shown after unlocking. These results show that participants largely agree for some environments like home and public transport, as expected. However, their perceptions diverge for other environments, such as work or university.

Data Rating	Environment Rating				
	uncritical	**rather uncr.**	**neutral**	**rather cr.**	**critical**
insensitive	389 (87%)	6 (1%)	16 (4%)	7 (2%)	27 (6%)
rather insens.	34 (32%)	40 (37%)	16 (15%)	11 (10%)	6 (6%)
neutral	155 (32%)	108 (22%)	112 (23%)	83 (17%)	23 (5%)
rather sens.	78 (30%)	53 (20%)	26 (10%)	79 (30%)	28 (11%)
sensitive	39 (54%)	6 (8%)	10 (14%)	8 (11%)	9 (12%)

Table 4. Perceived sensitivity of data intended to be accessed in the upcoming short access session, in combination with the currently perceived criticality of the environment.

Data Sensitivity Rating	Short Access	Full Access
insensitive	54.90%	27.38%
rather insensitive	12.55%	6.73%
neutral	15.69%	39.59%
rather sensitive	14.51%	20.38%
sensitive	2.35%	5.92%

Table 5. Comparison of the distribution of perceived data sensitivity between access methods. Short access was mainly used for insensitive data.

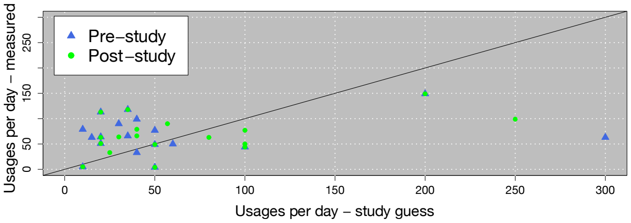

Figure 5. Daily usage frequency, compared between guessed usages (x), and measured ones (y). Participants mostly tended to underestimate their daily number of usages, similar to findings of related work [12].

To complete the picture, Table 5 shows the chosen access methods in relationship to perceived data sensitivity: Accessing insensitive data was the most common case for unlocking with *Snaps*, with 54.90% of these sessions. Neutral (15.69%), rather sensitive (14.51%) and rather insensitive (12.55%) data was reported in similar amounts, but accessing sensitive data rarely happened (2.35%). In contrast, during full access sessions, the most common data rating was neutral (39.59%). Similarly, rather sensitive (20.38%) and sensitive data (5.92%) was accessed more often after PIN unlock. This resulted in 27.38% insensitive and 6.73% rather insensitive cases, about half of the values observed for short access.

Estimated and Actual Daily Usage Frequency

Participants answered one questionnaire prior to the study, another one afterwards. Both asked for the number of phone usages per day. Figure 5 compares these guesses with the measured usages. Points above the diagonal are underestimations by the participants, points below indicate overestimation. In both guesses, most participants underestimated their phone usage. This matches the findings in related work [12].

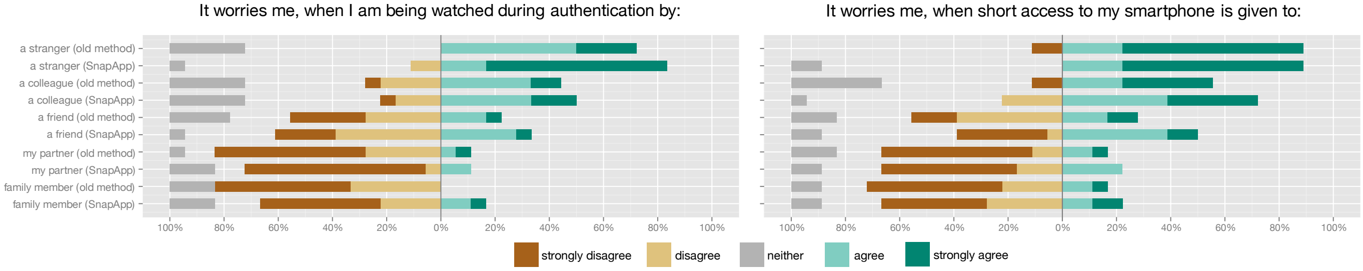

Figure 6. Ratings for two possibly worrying situations: Being watched while authenticating (left), and others gaining access to the device for a short amount of time (right). Overall, participants were slightly more worried for *SnapApp* compared to their usual unlock methods.

Rank	Pre-Study Rating	Post-study Rating	Logged Data
1	*messaging*	*messaging*	*messaging*
2	journey planner	e-mail	others
3	e-mail	journey planner	phone calls
4	*web browsing*	photography	*web browsing*
5	maps / navigation	web browsing	e-mail
6	photography	*music*	*music*
7	phone calls	social networks	games
8	news / magazines	maps / navigation	photography
9	music	phone calls	news / magazines
10	*social networks*	others	*social networks*
11	shopping	games	journey planner
12	games	news / magazines	shopping
13	*videos*	*videos*	*videos*
14	others	shopping	maps / navigation

Table 6. App category rankings based on perceived usage frequency in short sessions, assessed with questionnaires before the study (left) and afterwards (centre), compared to logged usage (right). Matching rankings are *highlighted*.

Estimated and Actual App Usage in Short Sessions

Both initial and final questionnaire also asked participants to rate their use of apps of certain categories during short phone sessions (specified as < 1 minute) on a 5-point Likert scale. We then compared these perceptions with the measured actual app use, as summarised in Table 6. Rankings in this table are based on the resulting Likert scores, and logged usage frequency during short access sessions, respectively.

Participants correctly judged their relative use of apps for messaging, browsing, social networks, and videos for sessions under a minute. They overestimated use of apps for e-mail and photography in such short sessions. They also initially underestimated music. Games were also used more often than estimated. In contrast, maps and navigation, as well as journey planners, were actually used less than estimated.

Specific Privacy and Security Concerns

Our questionnaires asked about being watched during authentication and others gaining access to the device for a short amount of time. Participants rated their agreement with worrying in these cases, with respect to 1) using *SnapApp* or their usual unlock method, and to 2) the kind of observer/other user. Figure 6 shows that participants worried marginally more for *SnapApp* compared to their usual methods (not significant, Wilcoxon signed rank tests, all $p > 0.05$).

Worries of Being Watched

Regarding being watched by others, participants mostly answered as expected: They worried the most to be watched by strangers, followed by colleagues, friends, partners, and family members. Table 7 shows the median ratings for their usual unlock method, Table 8 the median ratings for *SnapApp*.

Watched by	swipe users			PIN/pattern users		
	median	min	max	median	min	max
strangers	4	3	5	4	3	5
colleagues	4	1	5	3	2	5
friends	2	1	4	2	1	5
partners	1	1	2	1	1	5
family	1	1	3	2	1	3

Table 7. Participants' ratings for worries of being watched while authenticating with their usual unlock methods.

Watched by	swipe users			PIN/pattern users		
	median	min	max	median	min	max
strangers	5	3	5	5	2	5
colleagues	3	2	4	4	1	5
friends	2	1	4	2	1	5
partners	1	1	4	1	1	4
family	1	1	5	2	1	4

Table 8. Participants' ratings for worries of being watched while authenticating with *SnapApp*.

Short access by	swipe users			PIN/pattern users		
	median	min	max	median	min	max
strangers	5	1	5	5	1	5
colleagues	4	1	5	4	1	5
friends	3	1	5	2	1	5
partners	1	1	4	1	1	5
family	1	1	5	2	1	4

Table 9. Participants' ratings for worries of others gaining short access while authenticating with their usual unlock methods.

Short access by	swipe users			PIN/pattern users		
	median	min	max	median	min	max
strangers	5	4	5	5	3	5
colleagues	4	2	5	4	2	5
friends	3	1	4	4	1	5
partners	3	1	4	1	1	4
family	4	1	5	2	1	5

Table 10. Participants' ratings for worries of others gaining short access while authenticating with *SnapApp*.

Worries of Others Gaining Short Access

The second question addressed the risk of giving other people access to the unlocked device for a short amount of time, either via *SnappApp* or in usual ways, like handing over the unlocked phone.

Participants worried most about strangers gaining access. Concerns about colleagues or friends accessing the device were higher for *SnapApp* than for their usual methods. Table 9 shows the median ratings for their usual unlock method, Table 10 the median ratings for *SnapApp*.

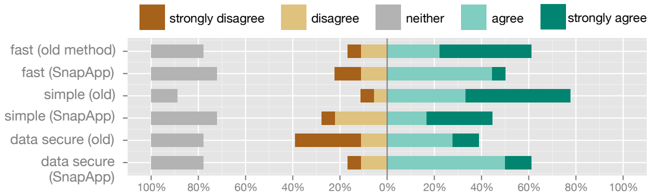

Figure 7. Participants' ratings of *SnapApp* and their usual unlock methods, regarding speed, ease-of-use, and security.

Short access by	swipe users			PIN/pattern users		
	median	min	max	median	min	max
speed (old method)	5	1	5	4	2	5
speed (*SnapApp*)	4	1	4	3	2	5
ease-of-use (old method)	5	1	5	4	2	5
ease-of-use (*SnapApp*)	2	2	4	3	1	5
data security (old method)	1	1	1	4	2	5
data security (*SnapApp*)	4	4	5	3	1	5

Table 11. Comparison of ratings for *SnapApp* and users' old methods.

Subjective Comparison of Unlock Methods

Participants rated both their usual unlock method (pre-study questionnaire) as well as our app (post-study questionnaire) – regarding speed, ease-of-use, and security. Figure 7 shows that most participants strongly agreed that the old unlock method was fast, with about 10% less agreeing on this for *SnapApp*. Simplicity of access with *SnapApp* received over 30% less agreement and about 15% more disagreement compared to usual unlock methods.

In contrast, *SnapApp* received higher ratings for data security than the usual methods, for which almost 40% of participants disagreed with being secure. Differences between *SnapApp* and the old methods were not significant (Wilcoxon signed rank tests, all $p > 0.05$). Table 11 shows a comparison of these ratings split by participants' usual unlock method.

SnapApp Experience

The post-study questionnaire also asked about aspects specific to *SnapApp* and the study, as shown in Figure 8. Over 80% of users agreed or strongly agreed on locking their phones before putting them away, showing that data logged between unlock and lock represents active use. About 60% regarded *SnapApp* as no time saver, while 25% were neutral. About 20% felt stressed during *Snaps*, and over 60% stated to have chained *Snaps*. Two thirds were annoyed when a short access session ended before they had completed their current interaction. Opinions diverged for vibration warnings announcing the end of short access sessions, found helpful by about 45% and unhelpful by 50%. Over half of the participants agreed on having sometimes realised that they could have used a *Snap* instead of full access after unlocking via PIN. Finally, while less than 20% of participants agreed that the blacklist was a useful feature for them, 50% stated that they would not use *SnapApp* without the blacklist feature.

Further Feedback

Our post-study questionnaire also asked participants if they want to keep using *SnapApp* (without data logging), or whether they will return to their usual unlock method. Participants also explained their decisions in a free comment field.

Three participants decided to keep *SnapApp* beyond the study. All 15 participants who decided against it left a comment to explain their decisions: Seven participants said they removed *SnapApp* due to the double PIN problem and three due to unexpected locks (explanation of these technical issues in limitations section). Besides these technical problems, three swipe users stated that they do not want to decide which method to take for each unlock. Another two participants mentioned delay before the *SnapApp* lockscreen showed up.

The following reasons were mentioned once each: one participant did not like the design, one wanted pattern instead of PIN as a full unlock method, and another one missed widgets on the lockscreen. For one PIN user, *SnapApp* was too insecure, while another PIN user was too accustomed to using PIN instead of a *Snap*. One user did not like *Snap deactivation*, whereas another one said that *Snap time* was too short. The last two statements suggest that these two participants did not read the instructions and never opened the app view itself, since the *SnapApp* settings allowed participants to configure both *Snap deactivation* and *Snap time*.

17 participants also provided a total of eight reasons why they liked *SnapApp*. The most commonly mentioned aspect (7 participants) was the "good" and "innovative" concept. One participant with both technical problems called it "theoretically convenient". Two participants found that *Snaps* were useful, while another two liked the customisable *Snap time*. Further two participants liked the implementation. Two more said they appreciated that their data was more secure, whereby one normally used swipe unlock and the other one PIN. Moreover, one participant valued the time saved with *SnapApp*, and another one the possibility to use multiple *Snaps* in a row. One swipe user liked that *SnapApp* was easier than using PIN only.

LIMITATIONS

Replacing the Android system lockscreen with our custom lockscreen app came with two technical issues:

First, device locks were caused when the OS seemingly unexpectedly killed our prototype's background service, likely due to low memory. However, an update that was installed by all participants during the field study fixed the problem.

Second, on many devices, the original Android system PIN prompt still shows up after unlocking via a custom lockscreen app, such as ours. Although it can be dismissed by a single touch without entering the PIN a second time, its appearance still has to be considered as annoying.

Twelve participants faced both problems. Two more only experienced the double PIN problem. This likely influenced users' views on some aspects, as further commented on in the discussion section.

Technical issues and prototype limitations (e.g. no lockscreen widgets) were the predominant reasons for most participants to refrain from using *SnapApp* after the study. We argue that even though only three participants decided to keep *SnapApp* beyond the study, the concept itself was received better than this number suggests. This is backed by participants' post-study feedback on what they liked about *SnapApp*. Seven ex-

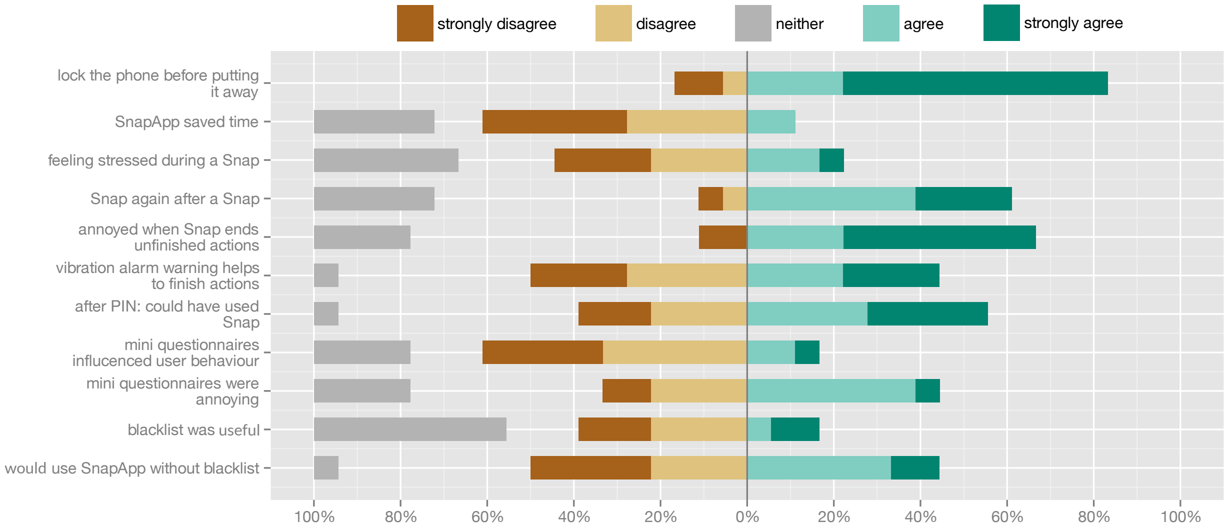

Figure 8. Ratings of aspects specific to *SnapApp* and the study.

plicitly stated that they liked the short-access concept, including participants who decided against keeping it. We thus expect that a smoother integration of our concept into the Android lock system would greatly increase adoption rates. In particular, this means modifying the OS to replace the original unlock screen with the *SnapApp* one, so that only this one (“unkillable”) unlock screen is shown.

It is possible that behaviour was influenced by the study itself, in particular the experience sampling questionnaires shown on the device itself. However, less than 20% of participants gave a positive answer when asked about feeling influenced.

DISCUSSION

We discuss the specific research questions in detail. In addition, we were able to gather further insights into the usability and perception of time-constrained access models.

The Time-constrained Access Model was Accepted

Every fifth access was based on a *Snap*. As *Snaps* usually expired after ten minutes and were thus only available for that duration after each secure unlock, this result is generally promising. The achieved reduction in PIN entries, especially in critical (public) environments, also improves security with respect to shoulder surfing attacks.

Moreover, the majority of participants agreed that they sometimes could have used *Snap* instead, after unlocking via PIN. This indicates that the established habit of PIN entry might have lowered the number of short acccess usages, even in appropriate contexts.

Analyses showed that *Snaps* were primarily adopted for the intended scenarios with short interactions. *Snaps* were mostly used to access data perceived as insensitive. In contrast, participants chose to authenticate with their PIN when they intended to access more sensitive data. Hence, we argue that the observed number of *Snap* uses and the general usage pattern demonstrates acceptance of the time-constrained fast unlock method and a general understanding of the concept.

The Implementation Decreased Speed and Ease-of-Use

SnapApp was perceived as not quite as fast as the usual methods. Former swipe users may have judged *SnapApp* as slower due to PIN entry being required for unconstrained access. Moreover, *SnapApp* was rated worse regarding ease-of-use.

However, logged data shows that *Snaps* were significantly faster. Ratings were likely negatively influenced by the technical issues with lockscreen replacements in Android.

Hence, it is important to differentiate between acceptance of the concept and the ratings for the prototype implementation. While the concept seems to provide various positive aspects and was accepted, the prototype implementation negatively influenced the user experience.

Perceived Security is Influenced by Usual Unlock Method

Ratings on data security were not homogeneous across both groups of usual unlock methods. Former swipe users rated security with *SnapApp* much higher than with their old method, likely since *SnapApp* introduced a secure method (PIN). In contrast, ratings by PIN/pattern were lower for *SnapApp* than for their usual method. Presumably, the fast unlock option was seen as a risk by former PIN/pattern users.

Overall, participants were slightly more worried about being watched during unlock with *SnapApp*. PIN and pattern users were probably concerned about others discovering *Snaps* as a way to bypass authentication, while swipe users might have worried to be observed when entering PINs. Others accessing the device for short amounts of time caused only marginally higher concerns with *SnapApp*.

Besides the influence of known unlock methods, we assume that security ratings were negatively influenced by misunderstood configurations: Long *Snap times* and misconfigured blacklists may indeed open access to sensitive data, but short *Snap times* for suitable apps can mitigate threats (e.g. shoulder surfing), compared to always using PIN/pattern. Simpler configurations might partially mitigate such concerns.

Appropriate Use of Blacklist Requires further Support

Only five of 18 users configured the blacklist, although half of all users said they would not use *SnapApp* without a blacklist. Configuring the blacklist thus seemed like too much effort for most people. Those who used it tailored the concept to their needs and ideas. While personalised settings are useful, concepts should not fundamentally rely on custom configuration.

These results support default blacklists, including critical common apps, such as the Android system settings, which were automatically blacklisted in our prototype. At the same time, we assume that a real life deployment would lower the experienced configuration overhead as such configurations could happen during installation of each app, not all at once.

In summary, a simple configuration approach is needed to empower users to balance security and usability of the concept according to their needs. Active configuration may also be facilitated by making the current configuration more visible to users, for example by marking blacklisted app icons on the homescreen during short access.

Fast/Convenient Access can Outweigh Privacy Concerns

While users seemed to have rather different views on which apps were worth protecting, we saw a common pattern in allowing *Snaps* for frequently used apps, for which a high number of authentications would add up to a lot of time.

An interesting example are messaging apps: While personal messages are usually perceived as private and partially sensitive, this type of app was the most frequently accessed one for *Snaps*. In contrast, time-constrained access was less often adopted for e-mail based communication. We assume that the perceived usability benefit outruns privacy concerns for scenarios which require short, high-frequency interactions.

We also found differences within app categories, namely *Photography*: Camera apps were used more often during short access (2.05% of apps in short, 1.59% in full), while browsing and viewing pictures occurred more often after PIN (1.01% in short, 2.01% in full). The results indicate that *Snaps* might be useful even for specific functionalities within single apps (e.g. picture taking vs viewing; writing vs reading).

Comparison to Existing Alternatives

We further relate *SnapApp* to common existing concepts for fast access and authentication.

Regarding software solutions, it is worth noting that some unlock screens allow direct access to some apps (e.g. camera accessible by default on stock Android unlock screen), but the space for such shortcuts to apps on the unlock screen is limited. *SnapApp* can work alongside app shortcuts and could be adapted to set a timer for their use as well.

Considering hardware alternatives, modern fingerprint sensors can also offer fast and convenient authentication. However, in contrast to software solutions like *SnapApp*, such sensors are only available on high end devices and are not suitable for some short access cases (e.g. sweaty fingers while jogging, use of gloves). Moreover, fingerprints are personal data that not everyone wants to submit to the system.

CONCLUSION AND FUTURE WORK

This work is motivated by the observation that users spend considerable amounts of time on unlocking their smartphones, although their actual interactions after unlocking are often quite short [9, 12, 27].

We have presented a novel concept to reduce this authentication overhead, called *SnapApp*. As a key component, our approach offers users two unlock methods on one screen: While PIN entry enables full access to the device, users can bypass authentication with a quick sliding gesture that grants access for a limited amount of time (e.g. 30 seconds). To improve security, short access is limited to ten subsequent uses within ten minutes of the last PIN entry. Furthermore, a user-defined blacklist blocks certain applications from being used without secure authentication.

We conducted a 30-day field study with 18 participants, combining data logging, experience sampling, and questionnaires. Our results showed that *Snaps* significantly reduced unlock times. Perceived security was influenced by previous unlock methods. *SnapApp* was rated as more secure by swipe users, but less secure by PIN/pattern users. Although blacklists can improve privacy and security, they were used less actively than what is desirable from a security standpoint. Consequently, we discussed supporting proper blacklist configuration as a main conceptual challenge. Overall, the fast access model was adopted as intended, as *Snaps* were mainly used for short interactions involving insensitive data.

Future work could also support automatically locking the phone if certain functionality is activated during short access, not just blacklisting apps in general. For example, a user might choose that reading social networks is ok without PIN, whereas actively posting new content is not. Finally, the concept could utilise context, for example by changing *Snap time* and blacklist according to the current time and location.

REFERENCES

1. Panagiotis Andriotis, Theo Tryfonas, and George Oikonomou. 2014. Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. In *Human Aspects of Information Security, Privacy, and Trust*. Springer, 115–126.

2. Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. 2010. Smudge Attacks on Smartphone Touch Screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies (WOOT'10)*. USENIX Association, Berkeley, CA, USA, 1–7.

3. Andrea Bianchi, Ian Oakley, Vassilis Kostakos, and Dong Soo Kwon. 2011. The Phone Lock: Audio and Haptic Shoulder-surfing Resistant PIN Entry Methods for Mobile Devices. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 197–200.

4. Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling Asleep

with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 47–56.

5. Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1393–1402.

6. Mauro Cherubini and Nuria Oliver. 2009. A Refined Experience Sampling Method to Capture Mobile User Experience. In *Workshop of Mobile User Experience Research part of CHI'2009*, Y. Nakhimovsky, D. Eckles, and J. Rigelsberger (Eds.).

7. Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. 2012. Touch Me Once and I Know It's You!: Implicit Authentication Based on Touch Screen Patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 987–996.

8. Serge Egelman, Sakshi Jain, Rebecca S. Portnoff, Kerwell Liao, Sunny Consolvo, and David Wagner. 2014. Are You Ready to Lock?. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, New York, NY, USA, 750–761.

9. Denzil Ferreira, Jorge Goncalves, Vassilis Kostakos, Louise Barkhuus, and Anind K. Dey. 2014. Contextual Experience Sampling of Mobile Application Micro-usage. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 91–100.

10. Arpan Gupta, Markus Miettinen, N. Asokan, and Mate Nagy. 2012. Intuitive security policy configuration in mobile devices using context profiling. In *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Confernece on Social Computing (SocialCom)*. IEEE, 471–480.

11. Alina Hang, Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. 2012. Too Much Information!: User Attitudes Towards Smartphone Sharing. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design (NordiCHI '12)*. ACM, New York, NY, USA, 284–287.

12. Marian Harbach, Emanuel von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It's a Hard Lock Life: A Field Study of Smartphone (Un) Locking Behavior and Risk Perception. In *Symposium on Usable Privacy and Security (SOUPS '14)*. 213–230.

13. Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. 2013. CASA: Context-aware Scalable Authentication. In *Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS '13)*. ACM, New York, NY, USA, Article 3, 10 pages.

14. Eiji Hayashi, Oriana Riva, Karin Strauss, A. J. Bernheim Brush, and Stuart Schechter. 2012. 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 '12)*. ACM, New York, NY, USA, Article 2, 11 pages.

15. Amy K. Karlson, A.J. Bernheim Brush, and Stuart Schechter. 2009. Can I Borrow Your Phone?: Understanding Concerns when Sharing Mobile Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1647–1650.

16. Taekyoung Kwon and Sarang Na. 2014. TinyLock: Affordable defense against smudge attacks on smartphone pattern lock systems. *Computers & Security* 42 (2014), 137–150.

17. Nicholas Micallef, Mike Just, Lynne Baillie, Martin Halvey, and Hilmi Güneş Kayacik. 2015. Why Aren't Users Using Protection? Investigating the Usability of Smartphone Locking. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 284–294.

18. Ildar Muslukhov, Yazan Boshmaf, Cynthia Kuo, Jonathan Lester, and Konstantin Beznosov. 2013. 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 '13)*. ACM, New York, NY, USA, 271–280.

19. Stefan Schneegass, Frank Steimle, Andreas Bulling, Florian Alt, and Albrecht Schmidt. 2014. SmudgeSafe: Geometric Image Transformations for Smudge-resistant User Authentication. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*. ACM, New York, NY, USA, 775–786.

20. Julian Seifert, Alexander De Luca, Bettina Conradi, and Heinrich Hussmann. 2010. TreasurePhone: Context-sensitive User Data Protection on Mobile Phones. In *Proceedings of the 8th International Conference on Pervasive Computing (Pervasive'10)*. Springer-Verlag, Berlin, Heidelberg, 130–137.

21. Tetsuji Takada and Yuki Kokubun. 2013. Extended PIN Authentication Scheme Allowing Multi-Touch Key Input. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia (MoMM '13)*. ACM, New York, NY, USA, Article 307, 4 pages.

22. Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*. ACM, New York, NY, USA, 161–172.

23. Emanuel von Zezschwitz, Alexander De Luca, Bruno Brunkow, and Heinrich Hussmann. 2015a. SwiPIN: Fast and Secure PIN-Entry on Smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1403–1406.

24. Emanuel von Zezschwitz, Alexander De Luca, Philipp Janssen, and Heinrich Hussmann. 2015b. Easy to Draw, but Hard to Trace?: On the Observability of Grid-based (Un)Lock Patterns. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2339–2342.

25. Emanuel von Zezschwitz, Paul Dunphy, and Alexander De Luca. 2013a. Patterns in the Wild: A Field Study of the Usability of Pattern and Pin-based Authentication on Mobile Devices. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 261–270.

26. Emanuel von Zezschwitz, Anton Koslow, Alexander De Luca, and Heinrich Hussmann. 2013b. Making Graphic-based Authentication Secure Against Smudge Attacks. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI '13)*. ACM, New York, NY, USA, 277–286.

27. Tingxin Yan, David Chu, Deepak Ganesan, Aman Kansal, and Jie Liu. 2012. Fast App Launching for Mobile Devices Using Predictive User Context. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*. ACM, New York, NY, USA, 113–126.

28. Nan Zheng, Kun Bai, Hai Huang, and Haining Wang. 2014. You Are How You Touch: User Verification on Smartphones via Tapping Behaviors. In *2014 IEEE 22nd International Conference on Network Protocols (ICNP)*. IEEE, 221–232.