Using Icon Arrangement for Fallback Authentication on Smartphones

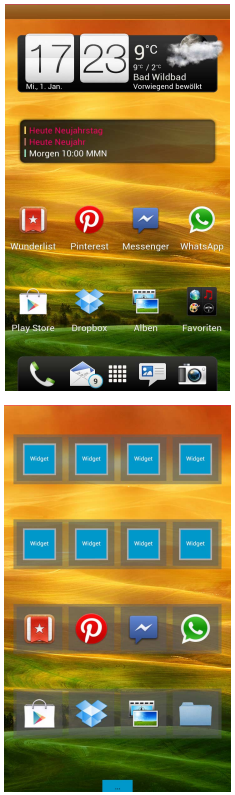

Figure 1. Concept I (Puzzle). Screenshots of original screen (top) and solution selected by user (bottom).

Alina Hang
Media Informatics Group
University of Munich (LMU)
Amalienstr. 17
80333 Munich, Germany
alina.hang@ifi.lmu.de

Alexander De Luca
Media Informatics Group
University of Munich (LMU)
Amalienstr. 17
80333 Munich, Germany
alexander.de.luca@ifi.lmu.de

Heinrich Hussmann
Media Informatics Group
University of Munich (LMU)
Amalienstr. 17
80333 Munich, Germany
hussmann@ifi.lmu.de

CHI 2014, April 26 - May 1, 2014, Toronto, ON, Canada.
ACM 978-1-4503-2474-8/14/04.
http://dx.doi.org/10.1145/2559206.2581169

Abstract
In this paper, we present three concepts for fallback authentication on smartphones that exploit the icon arrangement on home screens. A pilot study with three groups (each with n=6) was conducted to evaluate these concepts in terms of usability and security. The results show that most users made no or only few errors. In turn, adversaries had more difficulties in solving the tasks.

Author Keywords
Fallback authentication; Smartphone; Icon arrangement

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User interfaces – Input devices and strategies, evaluation.

Introduction
Fallback authentication is needed to enable users to regain access to their device and data, when the primary authentication fails, e.g. when the user enters the password incorrectly and has exceeded the number of authentication attempts.

Most solutions for fallback authentication can be found for web services. Popular approaches are the use of email-based password resets or security questions. While the former does not work well on smartphones (i.e. the email-client is usually on the smartphone that

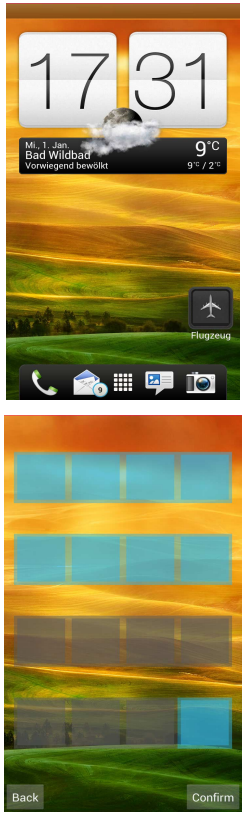

Figure 2. Concept II (Widget Space). Screenshots of original screen (top) and solution selected by user (bottom).

is blocked), it is difficult for the latter to design questions that are easy to be answered by users and hard to guess by adversaries [3], [5].

A common approach on smartphones is the use of personal unblocking codes [6]. However, retrieving this kind of information is difficult in situations where the user is not at home, in a foreign country or when Internet access is not available. Other solutions provided by the most popular operating systems for mobile devices are difficult to use under such circumstances as well. Examples are unblocking Android devices with an online email account or connecting Apple devices to a computer with iTunes.

We present three concepts for fallback authentication on smartphones that exploit the feature that icons on so-called home screens can be arranged by users in different ways. In most Android systems, users can have up to seven home screens that can be personalized and organized, e.g. by placing widgets (apps that run on home screens) or adding shortcuts to apps and folders. Previous work has shown that users implicitly learn the arrangement of their app icons during interaction [2]. Furthermore, users do personalize their home screens and arrange app icons based on different concepts like reachability or frequency [1]. In order to authenticate with the systems introduced in this paper, users have to rebuild their individual icon arrangement.

Brainstorming and Concepts

A brainstorming with 5 users (1 female) was conducted to identify suitable concepts for fallback authentication based on icon arrangements. Users were aged between 20-26 years (∅ 22 years). All users had a background in computer science and owned a smartphone (1 iOS, 4 Android). Based on the results of the brainstorming three concepts were developed:

Concept I – Puzzle:
This concept is based on a puzzle metaphor. Each element (app icon, widget or folder) on the screen is a puzzle tile that the user needs to place at the right position (see figure 1).

Concept II - Widget Space:
This concept is based on the space that widgets fill on a home screen. For example, if a user has two widgets: one at the top left corner (size: 4x2) and the other at the bottom right corner (size: 1x1), the user has to select the first two rows and the last field of the grid to submit a correct solution (see figure 2).

Concept III - App Selection:
This concept shows a pre-assembled home screen with 16 app icons. A random number ([0 … 16]) of icons that do not exist on the given home screen is taken from a library as distractor icons. Users have to select all app icons that are correctly positioned (and thus, actually exist on their home screen, see figure 3).

Implementation

The described concepts were implemented for Android Smartphones with a 4x4 grid (dock not included). All prototypes use the same grid size, since it is a common size for standard Android launchers. Launchers are the main view of the smartphone that manages the organization and launch of apps. The information about each element on its home screens is stored in a SQLite database (e.g. x-and-y coordinates). Since all concepts rely on this information, the database was parsed to log

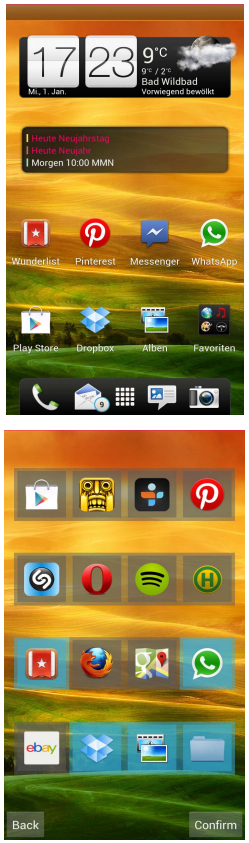

Figure 3. Concept III (App Selection). Screenshots of original screen (top) and solution selected by user (bottom).

the position of each element on the home screen and to compare it with the submitted solution.

Depending on the concept, some features of the prototypes vary: In the first concept, users can drag elements (apps, folders or widgets) from a sliding drawer and drop them on one of the 16 fields on the grid. While for the second concept users can select and deselect fields of the grid, the grid is filled with 16 app icons for the last concept. The app icons can be selected or deselected by touching them.

Threat Model

We assume an adversary who knows the user well [4]. This increases the chance that the adversary is familiar with the user's home screens. Furthermore, we assume that the adversary is in possession of the user's smartphone and has failed to access the device using the primary authentication. Thus, the device gets blocked and the adversary tries the fallback mechanism to gain access.

Pilot Study

We used a between-groups design with the independent variable *concept* (concept I-III). This was done to prevent learning effects with respect to the icon arrangement. For each concept, a within-subject design was used to test two types of screens: main and secondary. While the former screen is the one that users use most often, the latter is the one that has the most elements. The order of main vs. secondary screen was counterbalanced to minimize learning effects.

Study Procedure

We invited users to our lab and asked them to bring a close person who acted as the adversary. Once the users and their attendance arrived, we installed the prototype for the concept they tested on their smartphone and made the necessary settings. Users and adversaries were explained the study procedure and were shown how the prototype works. Adversaries were asked to leave the room, while the corresponding user was completing the tasks. After each task, users were asked to fill out a brief questionnaire. The same procedure was repeated for the adversary. 5 € gift vouchers were provided for participation.

Participants

18 users and 18 adversaries took part in the experiment (6+6 in each group). The data of one pair had to be removed from concept III due to incomplete information from the launcher's database. All participants owned an Android Smartphone with a 4x4 grid layout. This was a prerequisite to participate and necessary to limit the number of independent variables.

Users from group I (concept I; 2 female) were aged between 20-29 years (∅ 23 years); group II (concept II; all male) was aged between 17-29 years (∅ 22 years); and group III (concept III, 2 female) were aged between 20-25 years (∅ 22 years).

Users brought their partner, a close friend, or a colleague who acted as adversaries. Adversaries from group I (all male) were aged between 20-26 years (∅ 23 years); group II (1 female) was aged between 18-24 years (∅ 21 years) and group III (1 female) between 20-34 years (∅ 26 years).

Most users and adversaries had a background in natural sciences (e.g. computer science, physics). Others came from areas like teaching or psychology. The different

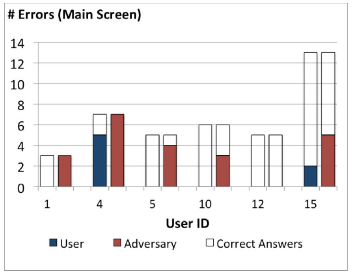

Figure 4. Number of errors for concept I (Puzzle, Main Screen).

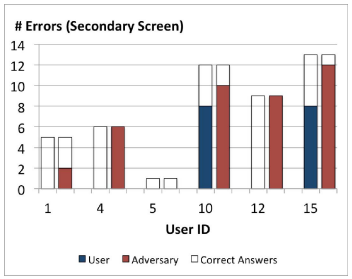

Figure 5. Number of errors for concept I (Puzzle, Secondary Screen).

backgrounds were evenly distributed over the different groups and also between users and adversaries.

Results

PUZZLE (CONCEPT I / GROUP I)

On average, users had 3 to 13 (∅ 7) elements on their main screen and 1-13 (∅ 7) on their secondary screen.

An error was counted when the user placed an element at the wrong position. Users had 0-5 errors (∅ 1), while adversaries made 0-7 errors (∅ 4; see figure 4). In one case, user and adversary made the same amount of errors (ID 12). Both of them had to place 5 elements, one of which was a big widget (4x3). User 4 made the most mistakes. This user could remember the shape that all apps located on the screen built, but not their position or order.

With respect to the secondary screen, users made 0-8 errors (∅ 3), while adversaries made 0-12 errors (∅ 7; see figure 5). However, one adversary did not make any errors. The adversary had to place only one big widget (4x4). Two users had a particularly high number of errors (ID 10 and 15). Those users had few widgets but many app icons and folders (12 elements).

Accuracy is a measure that indicates how well a system works. The following formula is used for calculation:

$$\frac{\sum true\ positives(TP) + \sum true\ negatives(TN)}{\sum all\ authentication\ attempts}$$

It returns a value between 0 and 1. The closer the value is to 1, the better. The accuracy value depends on a selected threshold (a value that decides if an authentication attempt is successful or unsuccessful). It also depicts how many incorrect answers an authentication system would accept in a real deployment. In our case the threshold is the number of errors that one allows users to make [0…16]. The accuracy for all possible thresholds was calculated. The best accuracy values (also in terms of best ratio between TP and TN) will be reported in the following.

The best accuracy value (83.3%, 5 TP, 5 TN) for the main screen is achieved by allowing users to make at most two errors. Allowing this does not negatively influence the number of false positives, but has a positive notion on the number of true positives.

For the secondary screen, the highest accuracy was 75% (4 TP, 5 TN). This value is very low. Thus, other types of errors were considered to calculate the accuracy: Since it is likely that in case users make errors, the distance of the wrongly placed element to the original one will be low [2], we weighted each error by multiplying it with the Euclidean distance of the affected element to its original position.

While the highest accuracy value for the main screen does not change, an accuracy of 91.7% (5 TP, 6 TN) can be achieved for the secondary screen.

WIDGET SPACE (CONCEPT II / GROUP II)

On average, users used 7 of the 16 fields of their main screen for widgets (min = 4; max = 10). 10 fields were used on average for the secondary screen (min = 4; max = 16).

Two types of errors were considered. For each field that users failed to select or that they wrongly selected, an error was counted. Users made almost no errors for the

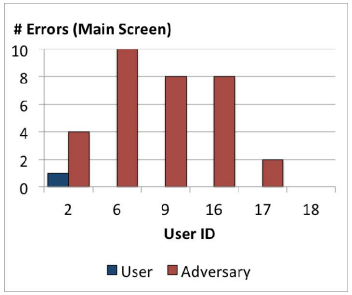

Figure 6. Number of errors for concept II (Widget Space, Main Screen).

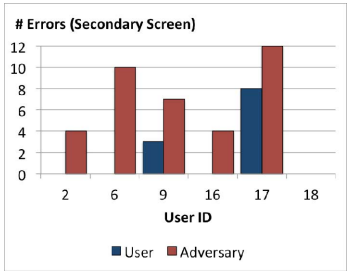

Figure 7. Number of errors for concept II (Widget Space, Secondary Screen).

main screen (min=0; max=1; see figure 6). However, in one case, an adversary had (similar to the user) no errors. This adversary selected the first two rows of the screen. In general, adversaries made 0-10 errors (∅ 5).

Users also had good results for the secondary screen (see figure 7). They made 0-8 errors (∅ 2). One user who made many mistakes had a widget that adapted its size dynamically. Thus, the user selected the right number of fields (2x2), but the widget size information was based on the maximum size it could reach.

Adversaries made 0-12 errors (∅ 6). One adversary had no errors at all. It is the same adversary who had no errors for the main screen and who has used the user's phone before (ID 18).

The highest accuracy value is achieved, by allowing users to make at most one error (91.67%, 6 TP, 5 TN). For for the secondary screen the highest accuracy is 83.3% (5 TP, 5 TN). Users can make up to 3 errors.

APP SELECTION (CONCEPT III / GROUP III)

For this concept, the number of elements that were at the correct position was randomly calculated. For the main screen, users had 0 to 7 correctly positioned elements (∅ 3). In turn, adversaries had between 0-2 (∅ 1) correctly positioned elements. For the secondary screen 2 to 7 elements (∅ 4) were correctly positioned for users, and 0-10 (∅ 4) for adversaries.

Different types of errors were considered for this concept. Each time the user misses to select or wrongly selects an element, an error was counted. For the main screen, users made only 0-2 (∅ 1) errors (see figure 8). Adversaries made between 1-8 errors (∅ 3). Only one adversary had the same number of errors as the corresponding user (ID 8). None of the given elements was correctly positioned and the user only made few selections resulting in this low error rate.

Users made few errors for the secondary screen (see figure 9). They made 0-3 errors (∅ 1). Adversaries made 1-6 errors (∅ 3). Again, the same adversary (ID 8) made only one error as for the main screen. Again, no element was correctly positioned. The user gave only one answer, resulting in the low number of errors.

We also had a closer look at the different types of errors. When users made errors, they either forgot to select an icon or selected an icon that actually existed on the screen (but at wrong position), but never selected apps that were included from the library.

The highest accuracy for the main screen was 80% (4 TP, 4 TN) and allows users to make at most one error. The highest accuracy for the secondary screen is only 70% (3 TP, 4 TN), also allowing at most one error.

Discussion

Most users made only few errors. In turn, adversaries had more difficulties in solving the tasks. However, some adversaries performed equally well. Those adversaries stated to have used the users' device before which might have helped them in solving the tasks. This is one of the worst-case scenarios.

Our study also showed that certain properties of home screen elements can be tricky. While adaptive widgets can increase the number of errors for users, big or common widgets can easily tell adversaries where they are to be placed. Thus, one should add further

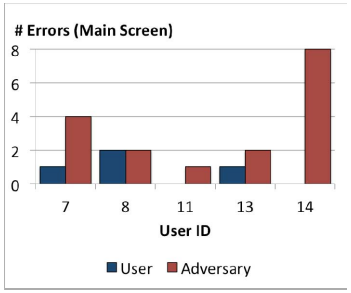

Figure 8. Number of errors for concept III (App Selection, Main Screen).

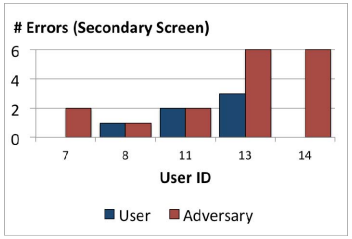

Figure 9. Number of errors for concept III (App Selection, Secondary Screen).

elements (e.g. from a library) that are not located on the screen to concept I, to obfuscate the number of elements to be placed.

Users had difficulties solving the task for concept I when they had too many app icons on the secondary screen. Instead of using the screen with the most elements as secondary screen, one might choose a screen that the users interact with more often (i.e. the screens next to the main screen).

For concept III, it was difficult for users to tell if apps that were located on their home screen were also at the right position. In turn, they could tell if an app is not located on the screen. The factor position of an app should be removed and the task should be reverted, by asking for apps that are not on the home screen.

The accuracies for concepts I and II are promising. Concept I has shown that considering different types of errors can increase the accuracy drastically. Since the sample size for each concept is small, the accuracy values hint at promising concepts, but are not generalizable.

It should also be noted that in a real world deployment not only one single concept will be used for fallback authentication, but a combination of multiple ones. This could be a combination of concepts I-III, but also a combination with other approaches like security questions. This will probably improve the accuracy, since the chances of adversaries to guess the correct answer will decrease with each question.

Conclusion and Future Work

The results of the pilot study give valuable insights on usability and security issues when exploiting icon arrangements on home screens for fallback authentication. The insights help to improve the three concepts and to evaluate them in a consecutive user study with a larger sample size that allows us to use inferential statistics to analyze the differences between each concept and each screen type.

Acknowledgements

We would like to thank John-Louis Gao for his help with the study.

References

[1] Böhmer, M. and Krüger, A. A Study on Icon Arrangement by Smartphone Users. In Proc. CHI 2013, ACM Press (2013), 2137-2146.

[2] Gustafson, S., Holz, C. and Baudisch, P. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In Proc. UIST 2011, ACM Press (2011), 283-292.

[3] Just, M. Designing and Evaluating Challenge-Question Systems. *IEEE Security & Privacy 2*, 5 (2004), 32-39.

[4] Muslukhov, I., Boshmaf, Y., Kuo, C., Lester, J. and Beznosov, K. Know Your Enemy: The Risk of Unauthorized Access in Smartphones by Insiders. In Proc. MobileHCI 2013, ACM Press (2013), 271-280.

[5] Rabkin, A. Personal Knowledge Questions for Fallback Authentication: Security Questions in the Era of Facebook. In Proc. SOUPS 2008, ACM Press (2008), 13-23.

[6] Rysgaard, B. A Method for Protecting User Data Stored in Memory of a Mobile Communication Device, Particularly a Mobile Phone. *European Patent No. EP 1107627* (2001).