A Testbed Tool for Comparing Usability and Security of Mobile Authentication Mechanisms

Karima Boudaoud, Marco Winckler, Zauwali S Paki, Philippe Palanque

A Testbed Tool for Comparing Usability and Security of Mobile Authentication Mechanisms

KarimaBoudaoud[1], Marco Winckler[1], Zauwali S. Paki[1], Philippe Palanque[2]
[1]Université Côte d'Azur, CNRS, I3S, France
[boudaoud, winckler]@univ-cotedazur.fr,
zauwali.sabitu.paki@gmail.com
[1]Université Paul Sabatier, IRIT, France
palanque@irit.fr

Abstract

In this paper, we are concerned by the configuration of authentication mechanisms and how tuning parameters values might affect at the same time the usability and the security of systems. Both usability and security are important properties for interactive systems however, tuning the application (for example to reduce the number of trials) to favor one property (such as security) might decrease another (usability), and vice-versa. In order to investigate the dependencies between usability and security, we propose in this paper a testbed environment for supporting the comparative assessment of authentication mechanisms. The tool presented in this paper allows varying parameters of authentication mechanisms to settle multiple configurations. It integrates a log mechanism for precisely recording the users' interactions. The current implementation features a mobile application that embeds three authentications mechanism, namely: PIN code, Android Pattern Lock and Passface. Nonetheless, the approach can be easily extended to include other authentication mechanisms. The ultimate goal of this testbed is to support user testing of authentication mechanisms and compare the relationship between user's behavior and risk assessment of multiple configurations.

1 Introduction

In contemporary world, the most valuable resource is no longer oil but data and this is one of the reasons why we should care for personal data protection. Currently, lots of personal data (and password granting access to personal data elsewhere) are readily available from personal devices. Mobile devices become so powerful in the last years that they are slowly replacing desktop computers in many users' daily tasks including accessing online services such as managing banking accounts, browsing the Web, managing personal information and professional data including personal contacts and agenda, editing and transporting professional documents. The need of securing mobile devices is

thus obvious. Among the threats, unauthorized access to data is the most common and for that many user authentication mechanisms have been proposed in the last years [2].

It is interesting to notice that authentication mechanisms are a user-dependent issue, for that we have to take users capabilities and skills into account when assessing the technology. Jonathan Grudin [3] found that users would subvert any technology that did not directly benefit them in a group-based technological environment. As highlighted by Renaud [4], this finding appears to apply to authentication mechanisms too: people often work around these mechanisms, which are put there explicitly to protect them, because they do not fully understand the benefits that will accrue from observation of security guidelines. For example, too complex polices (e.g., long sequences of characters, numbers and special codes) are harder to remember so many users are tempted to make use of remembering aids (including post it visible over the computer). In other words, the efficiency of authentication mechanisms also depends on the overall usability of the interactive system and the security policies that implement it.

The usability is defined by the standard ISO 9241-11 (1988) [1] as "the extent to which a product can be used by the target users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Authentication mechanisms impose to users an ancillary task (to identify to the system), which certainly is not the reason why the user is using the system. Authentication takes times, reduces performance and requires from users to memorize additional information on how to log in to the system. It might cause stress and dissatisfaction especially when authentication errors occur. We cannot objectively assert that authentication mechanisms are per see usable, but users might accept to use them up to a certain level without introducing deviant behaviors that might jeopardize security.

Currently, we still know very little about the factors that affect the usability of authentication mechanisms for mobile devices. However, we suggest that the interaction technique (way of entering personal identification) and the security policies implemented as parameters might play a role. In order to investigate this problem, we have implemented a testbed environment for assessing three authentication mechanisms, namely: PIN code, Android Pattern Lock and Passface. This testbed was conceived as an application that is aimed at helping with the setup of usability testing [5] of authentication mechanisms on mobile devices. The testbed has two modes: a setup mode for allowing the setup of the parameters of the experimentation and, a running mode that is used to collect user feedback. The setup mode allows tuning security policies such as the number of trials allowed, the length of passwords, delay between user interactions, availability (or not) of feedback for the user, etc. The user interaction is recorded in the form of a log file that can capture precisely the time of user interactions (which is an important feature to assessing performance), errors and mistakes. User satisfaction is measured in the testbed using a SUS questionnaire [6]. The rest of the paper is organized as follows: section 2, introduces the three authentication mechanisms and the diversity of parameters we are allowed to tune in each implementation; section 3 describes the testbed, in particular the two modes of functioning (setup and running modes); and, lately, section 4, concludes this paper and gives an overview about future works.

2 Overview of Authentication Mechanisms

Hereafter we present the three authentication mechanisms that are most commonly used to identify users of smartphone: PIN code, Android Pattern Lock and Passface. As we shall see, each of these mechanisms implemented a different set of parameters that can be tuned and combined to settle a large set of security policies. It is also worthy of notice that these three authentication mechanisms are based on memorization of textual/graphical elements, which require some cognitive effort and are

very prone to forgetting. Thus, reducing the amount of information to be recalled would ultimately improve the overall usability of the authentication mechanisms.

2.1 PIN Code (Personal Identification Number)

PIN is a special case of a password using number only. The original patent [7] dates back to 1966 and it became very well know from ATM machines. PIN was also adopted by smartphone makers who, in some cases extended the length of the PIN from 4 digits in the original specification up to 17 in the newest Android system. The following parameters can be tuned in PIN implementation:

- *Random numeric keypad*: specifies how numbers are placed on the keypad: in random order or as classic numeric keypad (by default).
- *Pin code size*: represents the number of digits that makes up the PIN code. The default is 4 digits.
- *Number of attempts*: defines the number of trials before the device blocks the user. The default is 3.
- *Input indicator*: provides visual feedback to user actions. There are 4 possible options:
  1. No indicator at all. User does see anything on the phone screen as she/he enters the PIN.
  2. Dots to fill. The system shows a set of little circles equal to the length of the PIN. As the user enters the PIN, the corresponding circle of the entered digit is filled;
  3. Appearing dots. As user enters the PIN, a dot appears on the phone screen.
  4. Show last digit entered. This option allows the user to see the last digit entered. When the subsequent digit is entered, the previous digit is replaced with a dot.

2.2 Pattern Lock

In the Pattern Lock, the user is required to connect dots in a predefined order. Whilst Pattern Lock has many security drawbacks as presented in [8], it is still widely used in mobile phone. The parameters that can be varied for the pattern lock are the following:

- *Number of rows*: is the number of points/nodes laid horizontally on the screen. Default is 3.
- *Number of columns*: is the number of points/nodes laid down vertically. The default is 3. Number of rows and columns determine the total number of possible patterns available.
- *Size of the pattern*: is the number of interconnected points/nodes to create a pattern.
- *Size of the points/nodes*: is the size of the circles used to indicate points/nodes on the screen.
- *Number of attempts*: is the number of failures accepted when drawing the pattern before the phone blocks the user. The default is 3.
- *Vibration*: when this parameter is enabled, the phone vibrates as the user draws the pattern.
- *Stealth*: if this is enabled, the schematic plot of the pattern will be invisible on the phone screen.

2.3 Passface

Passface is a typical example of recognition-based systems that relies on user skill for face recognition [1]. The original implementation relies on human faces but many variants include other types of images. The Passfaces parameters include the following:

- *Number of tiles*: is the number photos/images displayed in a grid. Four options are available: 9 (3x3 grid), 12 (4x3 grid), 15 (5x3 grid), and 18 (6x3). The default is 9.
- *Photo Types*: it might be faces, animals, flowers, places, or other sets. The default is the faces.
- *Size of the Passfaces code*: is the number of photos that make up the password. The default is 5.

- *Number of steps*: is the number of steps/rounds to complete a Passface. If the size of the Passface code is 4, then the image selection and authentication will be done in 4 steps.
- *Image shuffling*: is the re-arrangement of the photos on the grid each time the user authenticates such that the images do not have fixed position on the grid. Image shuffling is enabled by default. This makes the positions of the images really pseudo-random.
- *Use same image multiple times*: determines whether (or not) a user is allowed to use the same photo to compose a Passfaces code. By default, an image is used only once.

3 Testbed implementation

Our testbed is an Android app that simulates the three authentication mechanisms. The design and the implementation of the app is modular, so that it is possible to add as many other authentication mechanisms as needed in the future. This app features two modes: a setup mode and an experiment mode. These two modes correspond to the necessary steps for building a user testing experiment with users. Figure 1 illustrates the main screen of the app where we can distinguish the entries to the following features: *create settings*, which correspond to the setup mode; *run experiments*, which corresponds to the mode used for user testing; *generate and view setups as PDF*, allowing to create a snapshot of the parameters used in each experiment; *manage users ids,* used for creating unique ids for the user testing to protect participants identity; and, *delete setups*, for removing experimental data not longer in use. In addition to that, the apps implements a log file that is presented hereafter.

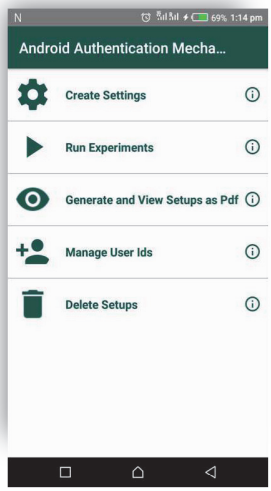


Figure 1: The main screen of the app

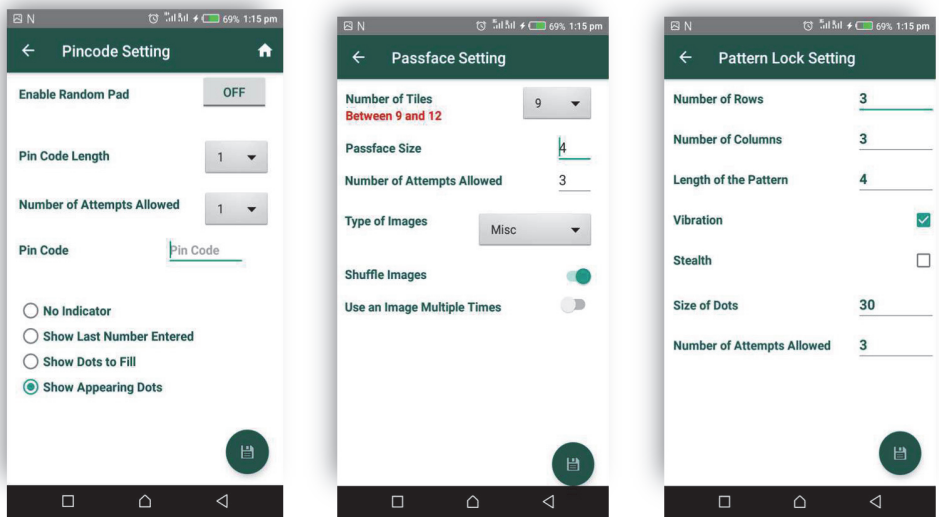
3.1 Setup mode

The main aim of the app is to conduct experiments by varying the various parameters associated with the authentication mechanisms. Therefore, the first step requires creating an evaluation setup with the configuration used in the user testing. So, using the appropriate entries, it is possible to configure the parameters for every authentication mechanism as shown in Figure 2. Every configuration must include a kind of password. The password is included as part of the configuration so that all users will use the same password in an experiment. This has the advantage of allowing a fair comparison of results assuming that the password is variable independent.

For the PIN code, the password is provided directly in the field *Pin code* as shown by Figure 2.a. For the Passface and Pattern Lock one more step is required, and we must use the screen as shown Figure 3 for entering the corresponding passwords.

For the Passface, we should tap the save button located on the bottom right of Figure 2.b. The app displays then the screen in Figure 3.a for the selection of the images. The images to be selected are in steps equal to the size of the Passfaces code as defined in Figure 2.b. After selection of the preferred images, theses image and the associated parameters set are then stored in the app database for use during the experimentation.

For the Pattern Lock, tapping on the save button located on the bottom right of Figure 2.c will open the screen as shown in Figure 3.b. After drawing the pattern, the button "TAKE SCREENSHOT" becomes active and allows the experimenter to take screen capture of the pattern. After taking the screen capture, the button "SAVE SETTING" becomes active so that the parameters set and the screen capture of the pattern can be saved in the app database.



a) PIN code *b) Passface* *c) Pattern Lock*

Figure 2: Setup for parameters of the three authentication mechanisms: a) PIN, b) Passface, and c) Pattern Lock.

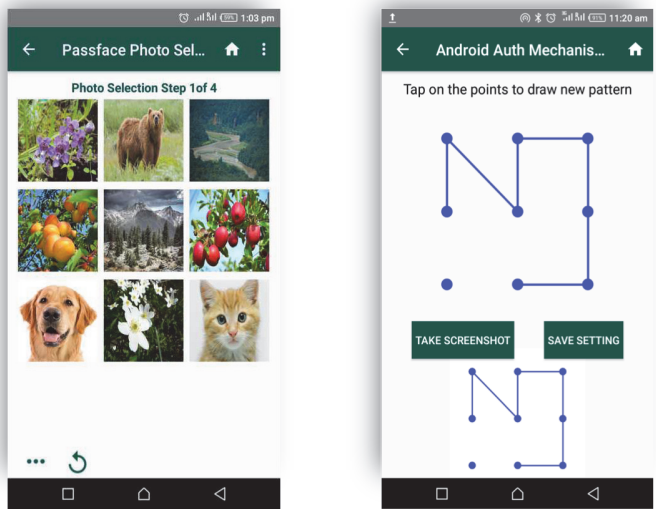
3.2 Running Mode

The goal of the running mode is to launch experiments using predefined configurations. The running mode is accessible from the option "Run experiment" on the main screen (see Figure 1) that leads to the screen shown in Figure 4.a, which presents the list of authentication mechanisms available. By selecting item in that list (Pattern Lock in this example), we move to the list of predefined configurations as shown in Figure 4.b. Once the configuration is selected, the application assigns an ID to the user who will perform the experiment (Figure 4.c). Identifying the users in this way allows preserving the identity of the real participants.

Figure 5 shows the screens used in the next steps of the evaluations. These screens are meant to be used by the user running the experiment. At first (Figure 5.a) the authentication mechanism is displayed; then the corresponding results follow with success (Figure 5.b) or fail (Figure 5.c).

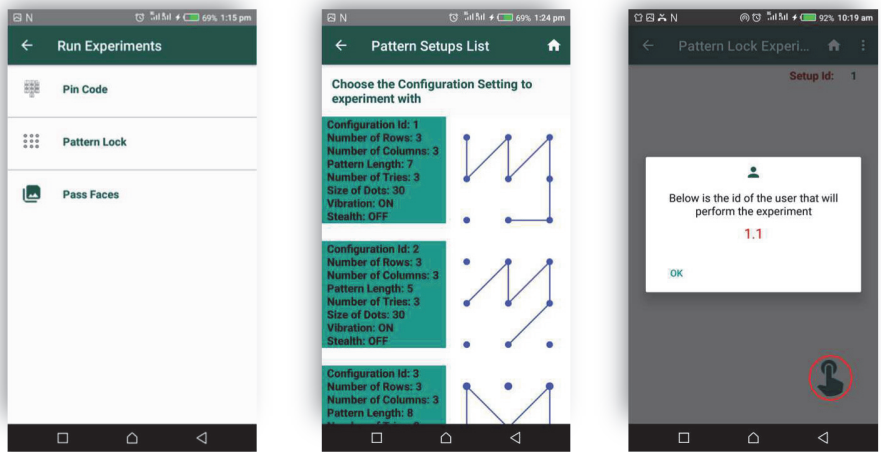
In order to facilitate the experiment, the app allows the experimenter to generate the setups as pdf files so they can be printed and used during the experiment. This can be done by selecting the option

"*Generate and view setups as PDF"* from the main screen (see Figure 1). Having a printed version of the configuration is particularly useful to show the users the password they are going to use during the test. We will typically give it printed to the user along with the mobile device that will be used in the test.



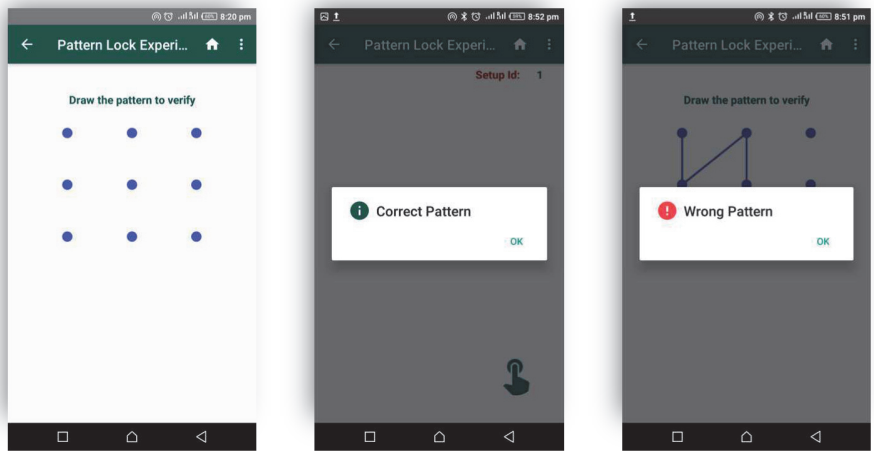
a) Passface *b) Pattern Lock*

Figure 3: Entering a Passface password (a) and Pattern Lock (b).



a) Authentication mechanisms b) List of configurations available. c) User ID.

Figure 4: Main screens in running mode used by evaluator.



a) Authentication mechanism b) Success c) Fail.

Figure 5: Main screens used by the user in the user testing.

3.3 Log Files

The app records the actions of the user during the experiments and stores this data as csv file. The app creates two folders: `Android Auth Mechs Log Files` and `Android Auth Mechs Setups PDFs`. The files `pattern_logs.csv`, `pincode_logs.csv`, and `passface_logs.csv` are the log files for Pattern lock, PIN code, and Passfaces authentication schemes respectively. These files are kept in `Android Auth Mechs Log Files`. All the csv files have the same pro form as illustrated by Table 1 below.

Config Id	User Id	User Actions	Attempt	Time	Date

Table 1: The pro forma for the csv log file for the 3 authentication mechanisms

`Config Id` is a unique identifier for a configuration setup. It is used to identify the configuration for a given experiment because there will be many configurations to experiment with. `User Id` uniquely identifies the user that performs an experiment. It is also just for identification purpose and does not relate to any personal identity of the user. `User Actions` is a column that records all the actions of a user during authentication: information such as when the user starts, when she/he finishes, and all the intermediary actions. `Attempt` column is used to keep information about during which trial the user does those action. For example, some users might be able to authenticate during the first attempt, some during the second attempt and so on. Finally, the `Time` and `Date` columns keep the time and the date of the authentication respectively.

In order to recall the setup of the experiment, the app supports the export of setups as PDF files named as `passfaces_setups.pdf`, `pattern_setups.pdf`, and `pincode_setups.pdf`. These files are available in the `Android Auth Mechs Setups PDFs folder`.

3.4 Managing Setups

After completing an experiment, the app allows the experimenter to get rid of setups that are no longer needed. This feature is also useful in case of mistakes when creating the setup. The

management of setups is available from the main screen (see Figure 1). The option “Delete Setups” gives the list of available configurations that can be deleted according to the needs. Deleting a configuration does not delete the corresponding date from the log files.

4 Conclusion

In this work, we presented a mobile app that can be used as a testbed for assessing suitable configurations for authentication mechanisms. This testbed might have multiple applications. One practical scenario, would be to assess how users behave when using a specific authentication mechanism; in particular, users are not the same and cultural and personal traits might make people more or less keen to adopt more strict security measures.

However, the ultimate goal of this testbed is to help to automate user testing of authentication mechanisms so that we can start to investigate trade-offs between security and usability. Using this tool, we offer an opportunity for researchers to discover the impact of varying the parameters associated with these three authentication schemes in order to find the best acceptable thread-off between security and usability. To the best of our knowledge, there is no similar tool to our testbed.

For now, the application has been used as a proof of concept and tested with colleagues and students. In future work, we will focus on running large scale experiments.

The tool is freely available so that it can be used by the community of researchers interested in usability and security of authentication mechanisms.

References

[1] ISO 9241-11 (1998) Ergonomic requirements for office work with visual display terminals Part 11: Guidance on Usability. This standard has been revised by the ISO 9241-171:2008 Ergonomics of human-system interaction--Part 171: Guidance on software accessibility.

[2] Marcin Rogowski, Khalid Saeed, Mariusz Rybnik, Marek Tabedzki, Marcin Adamski. User Authentication for Mobile Devices. Khalid Saeed; Rituparna Chaki; Agostino Cortesi; Slawomir Wierzchoń. 12th International Conference on Information Systems and Industrial Management, Sep 2013, Krakow, Poland. Springer, Lecture Notes in Computer Science, LNCS-8104, pp.47-58, 2013, Computer Information Systems and Industrial Management.

[3] Jonathan Grudin, “Social Evaluation of User Interfaces. Who Does the Work and Who Gets the Benefit?” in H-J Bullinger and B. Shackel (eds.), Proceedings of INTERACT 1987 IFIP Conference on Human Computer Interaction (Elsevier,1987), 805–811.

[4] Karen Renaud. Evaluating authentication mechanisms (Chapter Six). Security and usability. O’Reilly Media, pages 103-128.

[5] Jeffrey Rubin and Dana Chisnell. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley Publishing, Inc. 2008. 384 pages. ISBN-13: 978-0470185483.

[6] John Brooke. 2013. SUS: a retrospective. J. Usability Studies 8, 2 (February 2013), 29-40.

[7] Ivan, A., Goodfellow, J.: Improvements in or relating to Customer-Operated Dispensing Systems. UK Patent #GB1197183. doi:10.1049/el:19650200 (1966).

[8] 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.

[9] S Brostoff and A Sasse. Are passfaces more usable than passwords? a field trial investigation. In S. McDonald, editor, People and Computers XIV - Usability or Else! Proceedings of HCI 2000, pages 405–424. Springer, 2000.