Pattern Lock Evaluation Framework for Mobile Devices: Human Perception of the Pattern Strength Measure

Agnieszka Bier(✉), Adrian Kapczyński, and Zdzisław Sroczyński

Institute of Mathematics, Silesian University of Technology,
Kaszubska 23, 44-100 Gliwice, Poland
{agnieszka.bier,adrian.kapczynski,zdzislaw.sroczynski}@polsl.pl,
http://www.mmi.edu.pl

Abstract. In this paper we present a multiplatform research framework for acquisition of pattern locks and evaluation of their strength. Along with the design features of the framework itself we provide the analysis of the data set of more than 300 pattern locks acquired by means of this solution. In particular we compare the subjective human rating of pattern memorizability and complexity with the numerical values of pattern strength measure.

Keywords: Authentication · Mobile access control · Pattern strength measure

1 Introduction

This article is devoted to user authentication performed in mobile device environment. The presented research work has been motivated by preliminary results obtained using the specialized research framework for the assessment of the quality of lock patterns [5]. The follow-up of that project was aimed at broadening the research scope, as well as setting new research questions. In this paper we present the methods, tools and results of conducted experiment in the field of mobile device user authentication.

2 State of the Art

Contemporary computing is based on ubiquitous usage of mobile devices, which hold the essential personal data alongside corporate secrets. This is why the security of the mobile terminals becomes vital in almost every software project. The key factor in the mobile security is device locking. To unlock the mobile device the operator should perform a self-identification procedure leading to fast and comfortable access to the allowed functionality. There are some well-known identification procedures taken from personal computers that are not perfectly suitable for the mobile requirements, as alphanumeric passwords for example.

A. Gruca et al. (eds.), *Man-Machine Interactions 5*,
Advances in Intelligent Systems and Computing 659, DOI 10.1007/978-3-319-67792-7_4

The reason is on-screen keyboard rendered at the touch screen, which is often rather small and generates annoying errors in the input.

Regardless of those doubts there are several different methods for the unlock procedure: numeric (PIN codes [22]), alphanumeric (passwords), graphical (touch patterns, gestures over the picture [9]), behavioral [12,13,15,23] (memory games, finger knocks) and biometrical (fingerprints [7,20], face recognition, eye recognition). The advantages and disadvantages of particular methods fit them for the needs of people of different ages, manual skills, disabilities, education level and overall computer experience. The usability of the unlock process is also widely elaborated [1,2] to make it interact smoothly with common tasks done with the use of mobile devices. It is worth noting, that the hardware needed to perform different unlock methods varies, especially when it comes to graphics processing. The algorithms necessary to ensure stability or prevent counterfeiting of such procedures may become relatively complex [8,14].

One of the most popular unlock methods is entering a pattern by pointing and connecting control points (nodes) arranged in a grid at the touch screen. The grid size is 3×3 in most applications, although bigger sizes are considered (for example 4×4 in [6]). This method is often advertised as fast, easy to remember and providing the security level comparable to alphanumeric passcode. Because of its popularity in production environments, the research and elaboration of this security procedure becomes important nowadays [11,16,17].

There are some measures of the quality for the grid-pattern unlock procedure, described in [5,17,21]. The quality measure differs from measuring simple quantities (as length) because its definition is in general based on intuitive robustness to attacks. It also should correspond with human subjective classification.

To develop adequate measure, it is necessary to determine particular quality factors, which contribute to the overall security of a given pattern. The most often used ones are: pattern length [9] and pattern complexity estimated by the number of changes of direction [17], the number of repeated nodes and the number of connections of specific direction [19] or symmetry of the pattern [1,9], statistics of human behaviour—like probability distribution of starting node in [6,11,21], tendency of choosing patterns resembling letters and other common shapes [4], and memorizability [10]. Human-computer interaction factors important during the unlock process are especially thorough elaborated in [2,4,10].

In the following experiments we have used the measure and testing environment introduced in [5]. Because it utilizes multiplatform software development tools [18] the user experience during lock patterns collection is very close to natively spotted in the popular mobile operating systems, contrary to the other pen-and-paper laboratory experiments [4].

3 Research Methods and Tools

In this section research methods and tools are briefly described.

3.1 Research Methods

We have developed the research method in the field of mobile device user authentication. The method consists of two stages. First stage includes three steps, while the second stage includes two steps. During the first stage, authentication factors are generated and evaluated against memorizability and complexity. During the second stage, which is scheduled to take place 7 days after completion of the previous stage, the authentication factors generated during first stage are evaluated.

A detailed description of both stages is given below. In each step the user is asked to follow the requests:

- Stage 1: Step 1. Provide the identifier according to the scheme which consists of DeviceID-SexID-GroupID-UserID, Step 2. Enter the lock pattern and evaluate on the scale of 1 to 5 by complexity and memorizability, Step 3. Download the pattern from stored patterns then evaluate it on the scale of 1 to 5 by complexity and memorizability.
- Stage 2: Step 1. Enter the nickname specified during the step 1 of stage 1 (if cannot recall the UserID from the first stage, provide X as UserID), Step 2. Enter the same pattern as in stage 1 and evaluate its complexity and memorizability again (if cannot recall the lock pattern from the first stage, provide square pattern beginning from top-left corner and the following directions: Down, Right, Up and Left).

3.2 Research Tools

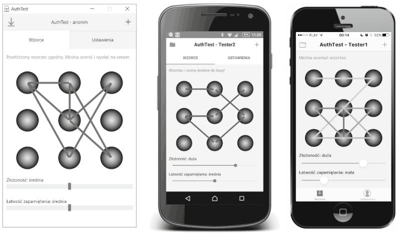
The test environment in conducted experiments consists of the test application and web service. The application mimics common interface implemented in mobile operating systems with all the look-and-feel details (including icons, tabs placement and virtual keyboard appearance) and therefore it is as close as possible to the real interface. This strongly distinguishes our experiment from the others, where web applications or simplified mockups were used.

The test application developed for our experiments is based on multiplatform FMX framework and native cross-compilers available in RAD Studio [18]. This development toolchain maintains single source code, shared between different software platforms while the user interaction designed with it dynamically introduces the proper graphical theme for the given operating system. Therefore it was possible to conduct the research for three different and not compatible operating systems without time consuming re-development of similar software solutions.

Despite the pattern lock codes are used commercially only in Android, we have also provided versions of the test software running at Apple iOS and Microsoft Windows. This way it is possible to detect slight differences in the lock code usage dependant on the origin of the user. Because of that the system does not follow the strict Android lock regulations and length limitations, as our goal was to examine also the most secure lock combinations, even those very long and relatively hard to remember.

Testing with the use of mobile applications gives also opportunity to measure timings, as the movement of the finger is natural. The tests for particular experiments described in the article were performed with the use of three Android devices: Lenovo 7-inch tablet (Lenovo TAB 2 A7-30H), Samsung phone (Samsung Galaxy S III Neo GT-I9301I) and Samsung 10.1-inch tablet (Samsung Galaxy Tab 4 SM-T535) and Windows desktop machines. In general, the test application was also tested by Apple iOS (iPhone and iPad), and Windows touch-enabled convertibles [5].

The design of the user interface of the test application running in different operating systems is shown in Fig. 1.



Powtórzony wzorzec zgodny. Można ocenić i wysłać na serwer

Złożoność: średnia

Łatwość zapamiętania: średnia

Wzorzec i oceny dodane do bazy!

Złożoność: duża

Łatwość zapamiętania: średnia

Można oceniać wzorzec

Złożoność: duża

Łatwość zapamiętania: mała

Fig. 1. The user interface of the multiplatform test application running under control of different operating systems.

The test application works in two different modes: *new*—for input and evaluation of the new unlock pattern and *eval*—for evaluation of the pattern downloaded from the repository. In the first mode the user can enter the pattern of his choice. The length of the pattern is not limited and the nodes can be repeated in the arbitrary manner, but connections cannot overlap nodes. This is different from the standard Android unlock procedure, because we indent to explore some novel, creative unlock schemes.

The system forces the user to repeat the pattern before rating and sending to the repository, which prevents against artificially long or random patterns, actually impossible to memorize by human. However, the user can enter long patterns with repeating parts, which can be the way to reduce the risk of the smudge analysis [3].

In the next step the user can rate the subjective difficulty level for the pattern entered and define the memorizability for it. The encoded pattern supplemented with these two ratings can be finally sent to the repository.

In the second mode the lock pattern is downloaded from the repository, and then the user can rate its complexity and memorizability and finally resend the results to the repository. The user is not obliged to rate every proposal which is downloaded randomly from the repository, as it can be boring to evaluate very short and simple patterns. Instead, the user can continuously request for new download. This way the set of expert ratings can be collected for patterns in the repository. The lock pattern in the second mode is visualized with the use of the animation, which illustrates the subsequent connections from the unlock pattern. This helps the user to understand the way the pattern was created and to determine its complexity and the easiness to memorize it.

The pattern repository is developed as the RESTful webservice over the cloud-based database, which ensures stability, security and scalability for larger experiments in the future. Every database record holds the information about lists of nodes and directions in the particular pattern, ratings (complexity and memorizability), the mode (new or eval), the timestamp, the version of the operating system, times needed to draw the pattern (for the first attempt and the repetition), the user's descriptor called "nick", which is used only to distinguish the group of the study and can be left empty to fully anonymize the experiment.

The node list contains nodes numbered from 0 to 8, while the directions are encoded into one-letter symbols (see Fig. 2). This model may seem somewhat redundant, but it fits well into the used measure of pattern complexity. It can also easily describe layouts different from the current 3 × 3 grid, i.e. larger (for example 4 × 4 [6]) or non-regular ones.

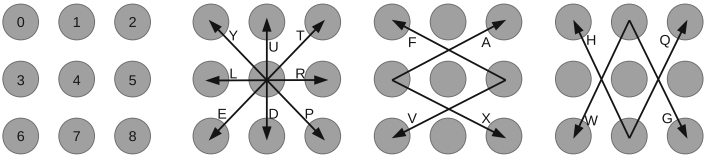


Fig. 2. Nodes numbering scheme and the encoding of directions.

The RESTful service uses JSON universal notation, which can be adapted by different client applications. The API specification introduces four methods to manipulate patterns: URL */patterns*, method: POST, data parameters: nodes, dirs, nick, ratecomplex, ratememory, mode, time1, time2, params (operating system version), returns: complexity measure for the given pattern, inserts the pattern data into the repository; URL */patterns*, method: GET, returns: nodes and encoded directions for the randomly chosen pattern from the repository; URL */measure*, method: GET, parameters: nodes, directions, a (optional coefficient), b (optional coefficient), returns: complexity measure for the pattern; URL */measures*, method: GET, returns: array with all stored patterns and their complexity measures.

4 Experiment

In the following we use the notation introduced in [5]. A pattern P will be viewed as a tuple $(P_0, d_1 d_2 \dots d_k)$, where $P_0 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ is the starting node and $d_1 d_2 \dots d_k$ is the sequence of consecutive directions

$$d_1,\, d_2,\, \dots,\, d_k \in \{L, U, R, D, Y, T, E, P, H, Q, W, G, X, V, F, A\} \tag{1}$$

(see Fig. 2 and [5] for the reference). The choice of P_0 and particular directions follow statistic distribution, resulting from human natural preferences. Patterns which seem more often used than others are considered less secure. The following quality measure for the lock patterns on 3 × 3 grid proposed in [5] was designed to detect the above pattern issues by assigning properly defined weights to particular node and direction choices:

$$m(P_0,\, d_1 d_2\, \ldots\, d_k) = (1 - p(P_0))(1 - \alpha^k)\frac{1}{3k}\sum_{i=1}^{s(k)} w(d_i)\left(\frac{1}{2} + \frac{2^{N_r} - 1}{2^{N_r+1}}\right), \quad (2)$$

where p is the probability distribution of choosing P_0 as the starting node, $\alpha \in [0.75, 0.99]$ is a parameter adjusting measure sensitivity to pattern length, $s(k)$ is the length of the sequence $d_1 d_2 \ldots d_k$ after reduction of consecutive repeating directions and w is the weight function of directions.

The goal of our experiment was to discover the relation between the investigated pattern strength measure and the subjective human perception of a lock pattern. In our calculations $\alpha = 0.8$, and the starting node probability distribution and the direction weight function were assumed as in [5] and are presented in the tables below.

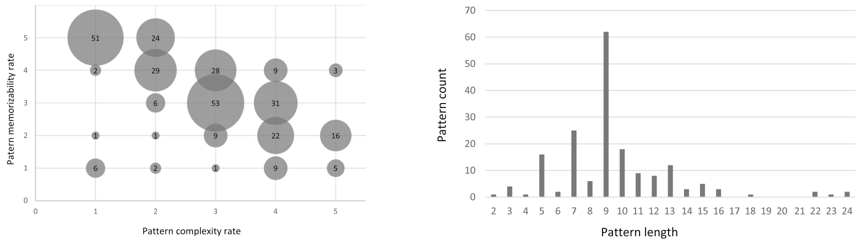
d	L	U	R	D	Y	T	E	P	H	Q	W	G	X	V	F	A
$w(d)$	1	1	1	1	2	2	2	2	3	3	3	3	3	3	3	3

P_0	1	2	3	4	5	6	7	8	9
$p(P_0)$	0.39	0.1	0.12	0.06	0.04	0.06	0.075	0.055	0.1

The experiment was held at Silesian University of Technology during the period 21–29.11.2016. The respondents were chosen among students and instructors of the Faculty of Applied Mathematics. They were asked to enter new pattern lock and rate its complexity (scale 1 (low) - 5 (high)) and memorizability (scale 1 (low) - 5 (high)) or rate a pattern already existing in the database. Also some additional data such as time needed to enter or reenter the pattern, type of device used (tablet, smartphone, PC), or gender, were collected. The analysis of the acquired data was performed with particular interest in typical and exotic cases. Below we present our observations derived from the data set of more than 300 pattern evaluations.

4.1 Main Features of the Data Set

The set of 308 pattern evaluations was first analyzed as a whole with respect to its general statistics. The pairs of indices (C, M), where C denotes complexity rate and M - memorizability rate were considered and counted along the data set regardless of the operation mode. The overall statistics for each pair (C, M) is presented in Fig. 3a. One can observe the natural correlation between the indices C and M, as the majority of cases follow the rule: the higher the complexity rate is - the lower the memorizability rate is. Although there is also a number of outliers, particularly with both low values of M and C, they are all simple, easy to memorize patterns, and a more probable explanation is the misunderstanding the memorizability rate scale rather than some unusual features in the estimated patterns.



(a) (b)

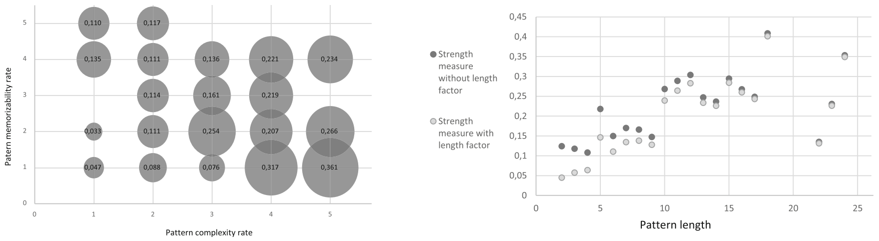
Fig. 3. Subjective evaluation of pattern complexity rate vs pattern memorizability rate (a), pattern length distribution (b).

Another interesting feature of the collected data is the new pattern length statistics. Among 181 newly introduced patterns the longest contains 24 nodes, while the shortest consists of only 2 nodes. The calculated average pattern length is 9.5, which is also close to the most popular length 9. On Fig. 3b we present the distribution of patterns of given length.

4.2 Pattern Strength Measure in Reference to Human Rating and Pattern Numerical Features

In order to verify the correspondence of human subjective ratings with the pattern strength measure, the average measure value for patterns evaluated with pair (C, M) was calculated and analysed. The resulting scheme (see Fig. 4a) reveals a general consistency of the average strength measure values with human perception in the region of low memorizability and high complexity of patterns. The average measure values for patterns rated as $(2, 4)$, $(2, 5)$, $(1, 4)$ and $(1, 5)$ are relatively high: 0.207; 0.266; 0.317 and 0.361, respectively. Outside this region the measure and ratings seem less consistent, however one can observe that the measure follows the complexity rates in the expected manner: the higher complexity rate, the higher value of the measure. Nevertheless, the measure seems less consistent with the memorizability rate for the presented set of data - not necessarily the lower memorizability rate agrees with higher strength measure value. The probable cause of the inconsistency is the fact that the human perception of pattern complexity may be quantified by the pattern length or number of different directions used and repetitions, which agrees with the measure design, while the memorizability remains poorly quantifiable. Also, the above observed confusion regarding the memorizability rate could contribute to this effect.

Figure 4b presents the average strength measure values for patterns of given length. The two series of data show the values of strength measure computed with



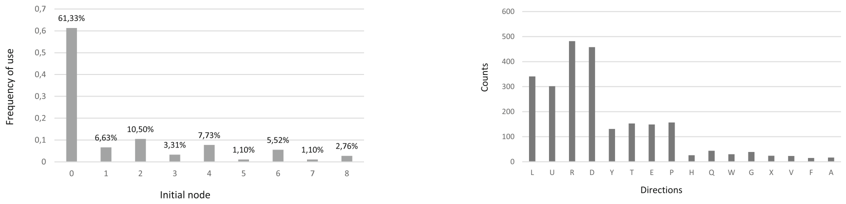
(a) (b)

Fig. 4. Average strength measure values for paired rating values (a), average strength measure vs pattern length (b).

and without the length factor. Clearly, the length factor improves the measure adequacy for short patterns. As expected, the measure values follow the length factor in a desired way in the range of length up to 20 nodes, which is not that clear in the measure without the length factor. However, for patterns of extreme length (more than 20 nodes), a disturbance in the dependency occurs. To understand this phenomena, we analyzed the patterns of length 22, 23 and 34. The reason is that they consist of multiple repetitions of simple patterns (like UURRDDLL) , decreasing the overall strength measure values.

4.3 New Statistical Input for the Measure Evaluation

The last part of our analysis was to develop statistics for the first node and direction choice distributions along the data set of newly introduced patterns. The obtained statistics are presented in Fig. 5.



(a) (b)

Fig. 5. Initial node distribution (a) and direction distribution (b).

Our observations allow for the future adjustment of the measure parameters according to the revealed distributions. Clearly, although the general tendency of initial node and direction distributions is consistent with the assumed parameters, one should adjust their exact numeric values to increase the measure performance in evaluating lock pattern security.

5 Conclusions

This paper was devoted to mobile device user authentication, with emphasis on examination of complexity and memorizability of visual passwords, i.e. lock patterns. The main insights from the research were presented in the previous section, however it shall be noted that drill-down of the data collected during the experiment could lead to a set of detailed observations. Among them, we can notice the tendencies related to picking the initial node, as well as the distribution of directions chosen during entering the pattern. Further research steps will be focused on comparative analysis of data obtained during the preliminary and secondary research experiments. Nevertheless we plan to refine the method, research tool and the performance measure.

References

1. Andriotis, P., Tryfonas, T., Oikonomou, G., Yildiz, C.: A pilot study on the security of pattern screen-lock methods and soft side channel attacks. In: WISEC 2013, pp. 1–6. ACM, Budapest (2013)
2. Aviv, A.J., Fichter, D.: Understanding visual perceptions of usability and security of android's graphical password pattern. In: CSAC 2014, pp. 286–295. ACM, New Orleans (2014)
3. Aviv, A.J., Gibson, K.L., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. Woot **10**, 1–7 (2010)
4. Aviv, A.J., Prak, J.L.: Comparisons of data collection methods for android graphical pattern unlock. In: SOUPS 2015. USENIX Association, Ottawa (2015)
5. Bier, A., Sroczynski, Z.: Evaluation of pattern lock codes strength for increased security in mobile applications. In: Rostanski, M., Pikiewicz, P., Buchwald, P., Maczka, K. (eds.) Proceedings of the 11th Scientific Conference Internet in the Information Society 2016. Academy of Business in Dabrowa Gornicza Press, Dąbrowa Górnicza (2016)
6. Budzitowski, D., Aviv, A.J., Kuber, R.: Do bigger grid sizes mean better passwords? 3 × 3 vs. 4 × 4 grid sizes for android unlock patterns. In: SOUPS 2015. USENIX Association, Ottawa (2015)
7. Cao, K., Jain, A.K.: Hacking mobile phones using 2D printed fingerprints. MSU Technical report, MSU-CSE-16-2 (2016)
8. Cejudo-Torres-Orozco, M., Garcia-Rios, E., Escamillahernandez, E., Nakano-Miyatake, M., Perez-Meana, H.: Counterfeit image detection in face recognition systems using stereo vision and optical flow methods. In: MCASE 2014. Council for Exceptional Children, Missoula (2014)
9. De Luca, A., Hang, A., Brudy, F., Lindner, C., Hussmann, H.: Touch me once and I know it's you!: implicit authentication based on touch screen patterns. In: SIGCHI 2012, pp. 987–996. ACM, Austin (2012)
10. Egelman, S., Jain, S., Portnoff, R.S., Liao, K., Consolvo, S., Wagner, D.: Are you ready to lock? In: SIGSAC 2014, pp. 750–761. ACM, Scottsdale (2014)
11. Goodin, D., Loge, M.: New data uncovers the surprising predictability of android lock patterns; tell me who you are, and I will tell you your lock pattern. http://arstechnica.com/security/2015/08/new-data-uncovers-the-surprising-predictability-of-android-lock-patterns/. Accessed 23 Feb 2017

12. Kapczynski, A., Kasprowski, P., Kuzniacki, P.: User authentication based on behavioral patterns. Int. J. Comput. **6**(1), 75–79 (2014)
13. Kapczynski, A., Sroczynski, Z.: Behavioral HCI-based user authentication. In: Rostanski, M., Pikiewicz, P., Buchwald, P. (eds.) Internet in the Information Society 2015. 10th International Conference Proceedings. Academy of Business in Dabrowa Gornicza Press, Dąbrowa Górnicza (2015)
14. Kwon, T., Na, S.: Tinylock: affordable defense against smudge attacks on smartphone pattern lock systems. Comput. Secur. **42**, 137–150 (2014)
15. Lee, J.D., Im, H.J., Kang, W.M., Park, J.H.: Ubi-RKE: a rhythm key based encryption scheme for ubiquitous devices. Math. Prob. Eng. **2014** (2014)
16. Siadati, H., Gupta, P., Smith, S., Memon, N., Ahamad, M.: Fortifying android patterns using persuasive security framework. In: UBICOMM 2015, p. 81. IARIA XPS Press, Nice (2015)
17. Song, Y., Cho, G., Oh, S., Kim, H., Huh, J.H.: On the effectiveness of pattern lock strength meters: measuring the strength of real world pattern locks. In: CHFCS 2015, pp. 2343–2352. ACM, Seoul (2015)
18. Sroczynski, Z.: Human-computer interaction on mobile devices with the FM application platform. In: Rostanski, M., Pikiewicz, P. (eds.) Internet in the Information Society. Insights on the Information Systems, Structures and Applications. Academy of Business in Dabrowa Gornicza Press, Dąbrowa Górnicza (2014)
19. Sun, C., Wang, Y., Zheng, J.: Dissecting pattern unlock: the effect of pattern strength meter on pattern selection. J. Inf. Secur. Appl. **19**(4), 308–320 (2014)
20. Szczepanik, M., Jóźwiak, I.J., Jamka, T., Stasiński, K.: Security lock system for mobile devices based on fingerprint recognition algorithm. In: ISAT 2015, pp. 25–35. Springer, Szklarska Poreba (2016)
21. Uellenbeck, S., Dürmuth, M., Wolf, C., Holz, T.: Quantifying the security of graphical passwords: the case of android unlock patterns. In: SIGSAC 2013, pp. 161–172. ACM, Berlin (2013)
22. Von Zezschwitz, E., Dunphy, P., De Luca, A.: Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices. In: MobileHCI 2013, pp. 261–270. ACM, Munich (2013)
23. Zargarzadeh, M., Maghooli, K.: A behavioral biometric authentication system based on memory game. Biosci., Biotechnol. Res. Asia **10**(2), 781–787 (2013)