An Anti-shoulder-surfing Authentication Scheme of Mobile Device

Jia-Ning Luo[1], Ming-Hour Yang[2], Cho-Luen Tsai[1]

[1] Department of Information and Telecommunications Engineering, Ming Chuan University, Taiwan
[2] Department of Information & Computer Engineering, Chung Yuan Christian University, Taiwan
deer@mail.mcu.edu.tw, mhyang@cycu.edu.tw, luen19881015@gmail.com

Abstract

Text-based passwords, such as personal identification number (PIN) and Android screen pattern locks, are the most commonly used identity authentication method in smartphones. However, text-based passwords are unable to prevent shoulder-surfing attacks; by directly looking at the passwords entered by users, attackers are able to steal the users' passwords, which poses significant threats to the users.

In this study, a new authentication mechanism was introduced. Such a method enabled users to send out misleading information to attackers when the former entered its text-based passwords; the latter was unable to decipher the true passwords by simply recording or looking at them. The misleading information was the pressure values (i.e., pressures exerted by the users) measured by pressure sensors embedded under the smartphone touchscreens. The systems detected each pressure value entered by the users and determined whether it was to be saved (i.e., as a true password) or omitted (i.e., as misleading information). Regarding this authentication method, because attackers were unable to know the users' pressure values, they were unable to differentiate between true and misleading information and thus had no way of knowing the users' actual passwords. In the end, our authentication mechanism improved the deficiency of current text-based passwords and enhanced system security.

Keywords: Shoulder-surfing attacks, Graphical password

1 Introduction

It is only quite recently that the use of biometrics for authentication purposes on mobile phones [1-4]. However, biometrics authentication mechanisms could contain potential vulnerability [5]. Of all identity authentication methods, text-based passwords are the most commonly used method by system and service providers.

However, text-based passwords are prone to shoulder-surfing attacks [6-7]. Shoulder-surfing attacks signify the practice of spying on the user of a device to obtain his/her passwords; such attacks are less time-consuming and have a high success rate [8]. Methods to protect users from shoulder-surfing attacks have been proposed in numerous studies, in which the methods are used to strengthen the security of the authentication system to prevent users' private information from being exposed. Shoulder-surfing attacks generally comprise the shoulder surfing resistant (SSR) method and the shoulder surfing immune (SSI) method; the former involves misleading people who are shoulder surfing (e.g., by using a spy-resistant keyboard) and the latter entails the use of many passwords and questions, in which users are required to answer all the passwords and questions correctly before they can log in (e.g., portfolio-based systems). However, because SSI-based password systems must store a considerable number of passwords and cannot implement hash storage to store all the answers, most password systems adopt the former method to fend off shoulder-surfing attacks. SSR-related techniques are described below:

Virtual password. For this method, users obtain their password via calculations made using mathematical functions [9] or special patterns [10-11]. This method prevents attackers from passing authentication because they have no knowledge about the function content. However, this method requires the users to spend more time to find their password; the users may also come up with the wrong password because of calculation errors.

Virtual keyboard. For this method, users are asked to enter their password in advance [12]. Text on the keyboard is then removed during authentication, preventing attackers from seeing the text [13]. Nevertheless, this method is not fail-safe as the attackers only need to observe the users a couple more times to find out what their password is. The use of virtual keyboards on mobile devices is effective because attackers are unable to remember the location of all the letters [14]; however, they may solve this problem by video recording the password entered.

Graphical password. Studies show that compared to words, graphics are easier to remember for humans. Therefore, this type of method uses a graphical interface as its verification tool, in which buttons are

[*] Corresponding Author: Ming-Hour Yang; E-mail: mhyang@cycu.edu.tw

DOI: 10.3966/160792642018081904028

employed to move icons to the authentication area or range [15-16]. Nonetheless, this method is not effective against recording attacks. Although the use of a graphical interface, an icon-selection method (i.e., requiring icons to be moved from specific directions or locations), and an icon-moving method (i.e., icons are moved to the authentication area or range by using buttons or sliding the screen) differs from the method in which passwords are preset by users and thus prevents shoulder-surfing attacks, they are not effective against recording attacks.

Second channel. For this method, a second channel is added to general authentication methods to eliminate shoulder-surfing attacks. Examples include telling users a verification code via earphones connected to their mobile phones and asking the users to enter the said verification code [17-23]. Although the content of the messages sent through the second channel is unknown, the verification image on the original screen is prone to recording attacks. However, a disadvantage to this method is that users are required to listen carefully before entering the verification code, resulting in prolonged authentication time.

The above results show that although graphical passwords may discourage shoulder-surfing attacks, they are not useful against recording attacks. In an effort to resist shoulder-surfing attacks, some graphical password systems increase the number of graphics used. However, this adds additional burden on users' memory. Conversely, the use of methods such as biological characteristics and human behavior can effectively prevent shoulder-surfing and recording attacks.

In this study, we introduced a password system that eliminated shoulder-surfing and recording attacks. In general, users show one of two behavioral patterns (i.e., honest or deceptive) when inputting information. For example, the former involves the users operating touchscreens in a habitual manner, whereas the latter involves the users operating the touchscreens in a manner that differs from their habits. Our system was able to identify users' behavioral patterns and separate the two input types into different categories while providing the same user interface, which enabled us to mislead our attackers.

In this study, a pressure sensor was installed on smartphone touchscreens to obtain the pressure value when users pressed on the touchscreens. Next, the users' behavioral patterns were determined by analyzing pressure value changes when the users input information. The results were incorporated into an existing authentication system to improve security. This method was effective because attackers were unable to steal the users' behavioral patterns (i.e., pressure values) by using shoulder-surfing attacks or recording attacks.

The method introduced in this study elevated a system's defense capability against shoulder-surfing attacks, effectively deterred recording attacks, and reduced a system's "false positive rate." The goals of this study are to enhance system security while providing users with a quick and convenient authentication process.

2 An Antishoulder-Surfing Attack-Based Identity Authentication System for Mobile Devices

In this section, we introduce an antishoulder-surfing attack-based identity authentication system suitable for touchscreen-based smartphones. The said system was incorporated into Android screen pattern locks, in which pressure exerted by users' fingers on the smartphone touchscreens was used as the basis for authentication. The method by which the users' pressure values were recorded is then explained. This method effectively prevented attackers from learning about users' true input values by looking at the users' passwords and did not require verification codes be sent via secure channels

2.1 New Methods for Defending Shoulder-Surfing Attacks

In this study, we introduced a new method in which pressure values input by users (measured by using touchscreen pressure sensors) were used as auxiliary inputs in addition to passwords. Shoulder-surfing attackers were thus unable to learn about the said values. Concerning the unit used by the smartphone pressure sensors to measure pressure values, it was *hPa*.

Smartphone touchscreens were employed as the users' input interface. Because the amount of pressure exerted by the users' fingers when operating the touchscreens differed between letters/numbers/symbols, our method recorded the corresponding times and pressure values for each letter/number/symbol input by the users. Our recording method was divided into four stages and are described as follows: (1) Feature Extraction; (2) Key Mapping; (3) Normalization; and (4) Convert data into authentication system outputs.

When a user drew a continuous line on the touchscreen (as shown on the left in Figure 1), segments with a pressure value greater than the threshold value of t were represented in red and a subsequent line was generated by our system, as shown on the right in Figure 1.

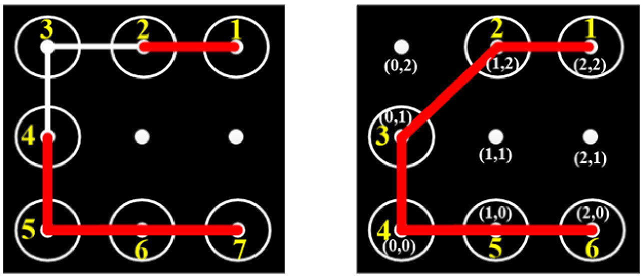

Figure 1. Continuous line drawn by user and subsequent line generated by the system

Feature extraction. The feature extraction process of the proposed system begins by recording the x-y co-ordinates and the touch pressure. Let $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$, $P = \{p_1, p_2, \ldots, p_n\}$ be the x and y co-ordinates and pressure value, respectively, of a signature with length n sampled at times $T = \{t_1, t_2, \ldots, t_n\}$.

Then, a sequence of vectors $W = \{w_1, w_2, w_3, \ldots, w_n\}$, is constructed where each vector element, $w_i = \{x_i, y_i, p_i, t_i\}$ which is a four-touple consisting of the co-ordinates, the pressure attribute, and the time.

In which a line drawn by a user when he/she touched the screen and released his/her finger from the screen (one time) was set as an input W. F represents the series of lines drawn by the user, $F = \{W_1, W_2, \ldots, W_k\}$, where $W_k = \{w_{k_1}, w_{k_2}, w_{k_3}, \ldots, w_{k_n}\}$ was a set of inputs in sequence.

Key-mapping. In this stage, the coordinate (x,y) is mapped to the key-value depend on the specific authentication system's input. For example, an authentication system contains nine key values (as shown in Figure 2), and the key value are defined by a set K={(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)}. The coordinate (x_i, y_i) is mapped to a key k_i in the authentication system by a map function: $k_i = Map(x_i, y_i)$.

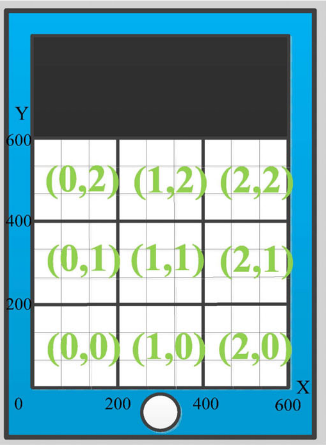

Figure 2. The key number of a specific authentication system

Using the x and y coordinates of w_i, the key number in which w_i was located could be calculated. The size of the mesh was related to the authentication method used and determined by the authentication system. Assuming that the x and y coordinates of w_i were (4,5) and that mesh size was 3*3 (as shown in Figure 3), the mesh number of x and y would be 1 (i.e., 4/3 = 1) and 1 (i.e., 5/3 = 1), respectively. Therefore, w_i would be located in the key (1,1).

0 (0,2)	1 (1,2)	2 (2,2)	3 (3,2)	4 (4,2)	5 (5,2)	6 (6,2)	7 (7,2)	8 (8,2)	9 (9,2)	A (10,2)	B (11,2)
C (0,1)	D (1,1)	E (2,1)	F (3,1)	G (4,1)	H (5,1)	I (6,1)	J (7,1)	K (8,1)	L (9,1)	M (10,1)	N (11,1)
O (0,0)	P (1,0)	Q (2,0)	R (3,0)	S (4,0)	T (5,0)	U (6,0)	V (7,0)	W (8,0)	X (9,0)	Y (10,0)	Z (11,0)

Figure 3. Password interface

Then, an alternate sequence of vectors $V = \{v_1, v_2, \ldots, v_n\}$, is constructed where each vector element, $v_i = \{k_i, p_i, \ldots, t_i\}$ which is a three-touple consisting of the key number, the pressure attribute, and the time.

In which a line drawn by a user when he/she touched the screen and released his/her finger from the screen (one time) was set as an input V. G represents the series of lines drawn by the user, $G = \{V_1, V_2, V_3, \ldots, V_k\}$, where V_k was a set of inputs in sequence.

Normalization. In this stage, continuous and identical key numbers were combined and the average pressure value was calculated. If the key number of v_i equaled v_{i+1}, $v_{i+2}, \ldots, v_j$, the elements are merged and the mean pressure value was substituted into v_i and v_{i+1}, $v_{i+2}, \ldots, v_j$, were deleted from the set, and the new pressure attribute as normalized as follows:

$$g_i = \frac{\sum_{m=1}^{j} g_m \cdot (t_{m+1} - t_m)}{t_{j+1} - t_i}..$$

This produced $V_i' = \{v_1', v_2', v_3', \ldots, v_p'\}$ and G' = $\{V_1', V_2', V_3', \ldots, V_k'\}$.

Each input (i.e., V_i') was separated into light or heavy pressure according to the threshold value t_{u_i}; different threshold values were set for different users. Only the heavy pressure values (i.e., V_i'') were retained and all the V_i'' s were grouped to form Set G''.

$$V_i'' = \{v_i' \mid g_i > t_{u_i}\}$$
$$G'' = \{V_1'', V_2'', V_3'', \ldots, V_q''\}$$

Convert into authentication system output. G'' was converted into authentication system outputs and send to the operating system.

3.2 Improved Text-Based Password

Normally, passwords are entered directly on screens (Figure 3), which is prone to shoulder-surfing attacks. However, by adding our method, even simple text-based passwords can be used to protect users from shoulder-surfing attacks. Such a method allows users to enter their own passwords while sending out misleading information to attackers, that is, the passwords observed by the attackers are not the true passwords.

Feature extraction. Assuming that a user's original password was "password," each text-based password input by the user (i.e., W_i) was subsequently recorded and all W_i were grouped into a set represented by F.

$$w_i = \{x_i, y_i, p_i, t_i\}$$
$$w_i = \{w_1, w_2, w_3, \ldots, w_n\}$$
$$F = \{W_1, W_2, W_3, \ldots, W_m\}$$

Key mapping. All of the coordinates (x_i, y_i) in w_i were converted into the key numbers v_i (keyboard location) to obtain G, a set containing all the key numbers v_i. The key numbers were then converted into "ppasmswotrdde."

$$k_i = Map(x_i, y_i)$$
$$V_i = \{v_1, v_2, v_3, \ldots, v_n\}$$
$$v_i = \{k_i, p_i, \ldots, t_i\}$$
$$G = \{V_1, V_2, V_3, \ldots, V_k\} =$$
$$\{(1,0,0.16,t_1),(1,0,0.17,t_2),(1,0,0.18,t_3),(10,2,0.17,t_4),$$
$$(4,0,0.2,t_5),(10,1,0.08,t_6),(4,0,0.22,t_7),(8,0,0.19,t_8),$$
$$(0,0,0.2,t_9),(5,0,0.1,t_{10}),(3,0,0.21,t_{11}),(1,1,0.18,t_{12}),$$
$$(1,1,0.18,t_{13}),(2,1,0.1,t_{14})\}$$

Normalization. All continuous and identical mesh numbers were combined to form Set V_i' to be included as a part of G', producing "pasmswotrde."

$$V_i' = \{v_1', v_2', v_3', \ldots, v_p'\}$$
$$G' = \{V_1', V_2', V_3', \ldots, V_p'\} =$$
$$\{(1,0,0.16,t_1),(10,2,0.17,t_2),(4,0,0.2,t_3),(10,1,0.08,t_4),$$
$$(4,0,0.22,t_5),(8,0,0.19,t_6),(0,0,0.2,t_7),(5,0,0.1,t_8),$$
$$(3,0,0.21,t_9),(1,1,0.18,t_{10}),(2,1,0.1,t_{11})\}$$

The elements of V_i' with light pressure values were removed and the elements with heavy pressure values were retained using the threshold value t_{u_i} (e.g., t_{u_i} = 0.15 hPa), after which the retained mesh numbers formed Set G'', producing "password."

$$V_i'' = \{v_i' \mid g_i > t_{u_i}\} = \{v_i' \mid g_i > 0.15\}$$
$$G'' = \{V_1'', V_2'', V_3'', \ldots, V_q''\} =$$
$$\{(1,0,0.16,t_1),(10,2,0.17,t_2),(4,0,0.2,t_3),(4,0,0.22,t_5),$$
$$(8,0,0.19,t_6),(0,0,0.2,t_7),(3,0,0.21,t_9),(1,1,0.18,t_{10})\}$$

Convert into authentication system output. Mesh numbers with hard pressure values were converted into a text-based password (i.e., {p,a,s,s,w,o, r,d}), which differed from the password observed by the attacker (i.e., {p,p,a,s,m,s,w,o,t,r,d,d,e}).

During the aforementioned authentication process, because the attacker was unable to observe the pressure values, he/she was unable to know that the user added misleading information while entering the password. In other words, the attacker saw the user enter the password "pasmswotrde," whereas the actual password entered by the user (one that could pass system authentication) was "password." Therefore, the attacker was not able to obtain the user's password by shoulder surfing nor pass authentication.

3.3 Improved Android Screen Pattern Locks

To unlock Android screen pattern locks, users were required to choose four of the nine dots (each dot had to be connected to at least one other dot), as shown in Figure 4; the dots selected need not be the closest to subsequent dots and each dot could be chosen only once. Because this authentication method was prone to shoulder-surfing attacks, it was integrated with the method proposed in this study. First, the mesh numbers of the nine dots from left to right, top to bottom were defined as (0, 2), (1, 2), (2, 2), (0, 1), (1, 1), (2, 1), (0, 0), (1, 0), and (2, 0).

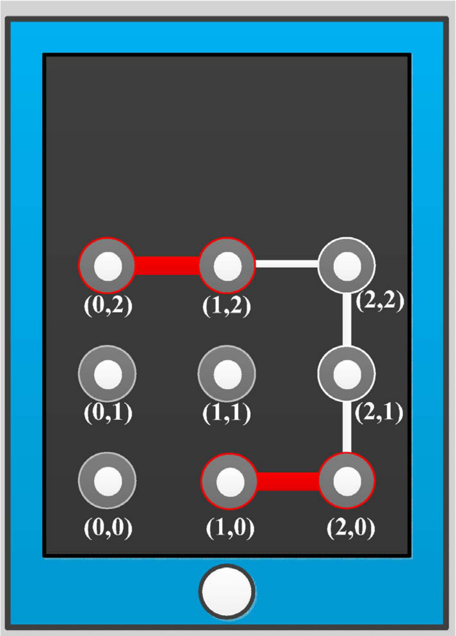

Figure 4. Android pattern lock interface

Feature extraction. In addition to the coordinates drawn by the user, the pressure values of the coordinates were obtained. The lock pattern selected by the user is shown in Figure 4 Because Android screen pattern lock allowed only one input, the user's input (i.e., F) contained only one set (i.e., W_1).

$$W_1 = \{w_{k_1}, w_{k_2}, w_{k_3}, \ldots, w_{k_n}\}$$
$$F = \{W_1\}$$

Key mapping. All the coordinates of W_1 (i.e., w_1 s) were converted into mesh numbers (i.e., v_1 s) to produce Set V_1, which contained all the mesh numbers.

$$k_i = Map\{x_i, y_i\}$$
$$v_i = \{k_i, p_i, \ldots, t_i\}$$
$$V_1 = \{v_1, v_2, v_3, \ldots, v_n\} =$$
$$\{(0,2,0.21,t_1),(1,2,0.18,t_2),(1,2,0.18,t_3),(2,2,0.1,t_4),$$
$$(2,1,0.12,t_5),(2,1,0.12,t_6),(2,0,0.19,t_7),(1,0,0.2,t_8),$$
$$(1,0,0.2,t_9)\}$$
$$G = \{V_1\} = \{(0,2,0.21,t_1),(1,2,0.18,t_2),(1,2,0.18,t_3),$$
$$(2,2,0.1,t_4),(2,1,0.12,t_6),(2,0,0.19,t_7),(1,0,0.2,t_8),$$
$$(1,0,0.2,\Delta t_9)\}$$

Normalization. Continuous and identical mesh numbers were combined, producing the following result:

$$V_1' = \{v_1', v_2', v_3', \ldots, v_p'\} = \{(0,2,0.21,t_1),(1,2,0.18,t_2),$$
$$(2,2,0.1,t_3),(2,1,0.12,t_4),(2,0,0.19,t_5),(1,0,0.2,t_6)\}$$
$$G' = \{V_1'\} = \{(0,2,0.21,t_1),(1,2,0.18,t_2),(2,2,0.1,t_3),$$
$$(2,1,0.12,t_4),(2,0,0.19,t_5),(1,0,0.2,t_6)\}$$

The elements of V_1' with light pressure values were removed and heavy pressure values retained according to the threshold value t_{u_i} (t_{u_i} = 0.15 hPa) to form V_1'', which was incorporated into G''.

$$V_i'' = \{v_i' \mid g_i > t_{u_i}\} = \{v_i' \mid g_i > 0.15\} = \{(0,2,0.21,t_1),$$
$$(1,2,0.18,t_2),(2,0,0.19,t_5),(1,0,0.2,t_6)\}$$
$$G'' = \{V_1''\} = \{(0,2,0.21,\Delta t_1),(1,2,0.18,\Delta t_2),$$
$$(2,0,0.19,\Delta t_5),(1,0,0.2,\Delta t_6)\}$$

Convert into authentication system output. The keys with heavy pressure values were retained and connected to produce the following result (Figure 5), which differed from the pattern observed by the attacker.

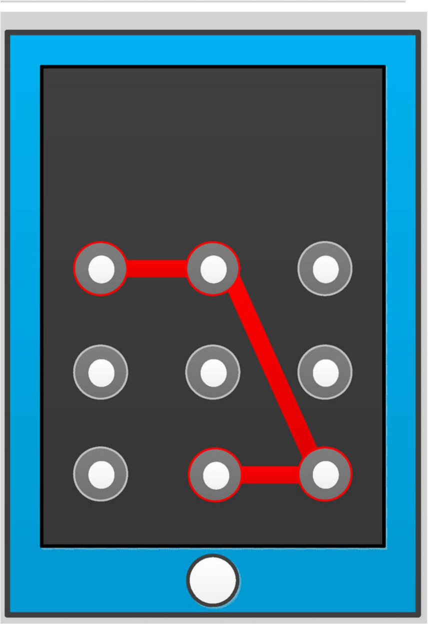

Figure 5. Authentication pattern generated by the system

$$\{(0,2) \rightarrow (1,2) \rightarrow (2,0) \rightarrow (1,0)\}$$

The results showed that despite the attacker seeing the authentication pattern entered by the user, such a pattern differed from the actual pattern generated by the system. This confirms that our method can be applied to Android pattern locks to prevent shoulder-surfing attacks.

4 Experiment and Analysis

Because pressure values record by the touch screen are differed between users and that pressure threshold set for each user differed, instances in which the user considered a pressure value to be light but the system assessed it to be heavy may occur. Therefore, this study analyzed and investigated the threshold value t_{u_i}.

A total of 10 pattern-lock samples exerted by 10 users were randomly selected and the average pressure value for each user was taken (as shown in Figure 6). The pressure values differed significantly between different users, in which light pressure values fell at approximately 0.0987 ± 0.05(hPa) and heavy pressure values fell at approximately 0.1909 ± 0.07(hPa). In fact, pressure values that were considered light for some users were heavy for others, signifying considerable differences in light and heavy pressure values between users. Therefore, different threshold value ranges were set for different users.

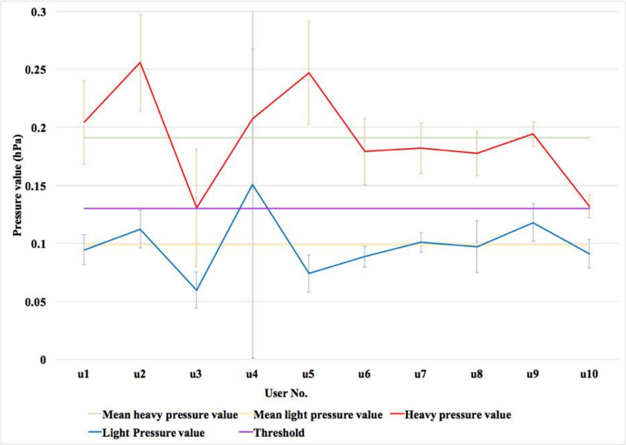

Figure 6. Comparisons between 10 randomly selected pressures exerted by 10 users

Using Android pattern locks, we produced five patterns and three variations for each pattern (as shown in Figure 7). For example, for Pattern A, Variation 1, the attacker observed a pattern of 1→2→3→4→5→6→7→8, whereas the actual pattern required to pass authentication was 2→3→4→5→6→7. To pass authentication, the pressure values of the "correct" dots entered must be greater than the threshold values. The 10 users were then asked to draw the pattern for each variation 10 times to observe their success rate, with the threshold value set at 0.13 (hPa). Patterns comprising pressure values greater than the threshold value were represented in red to enable convenient experimental observations.

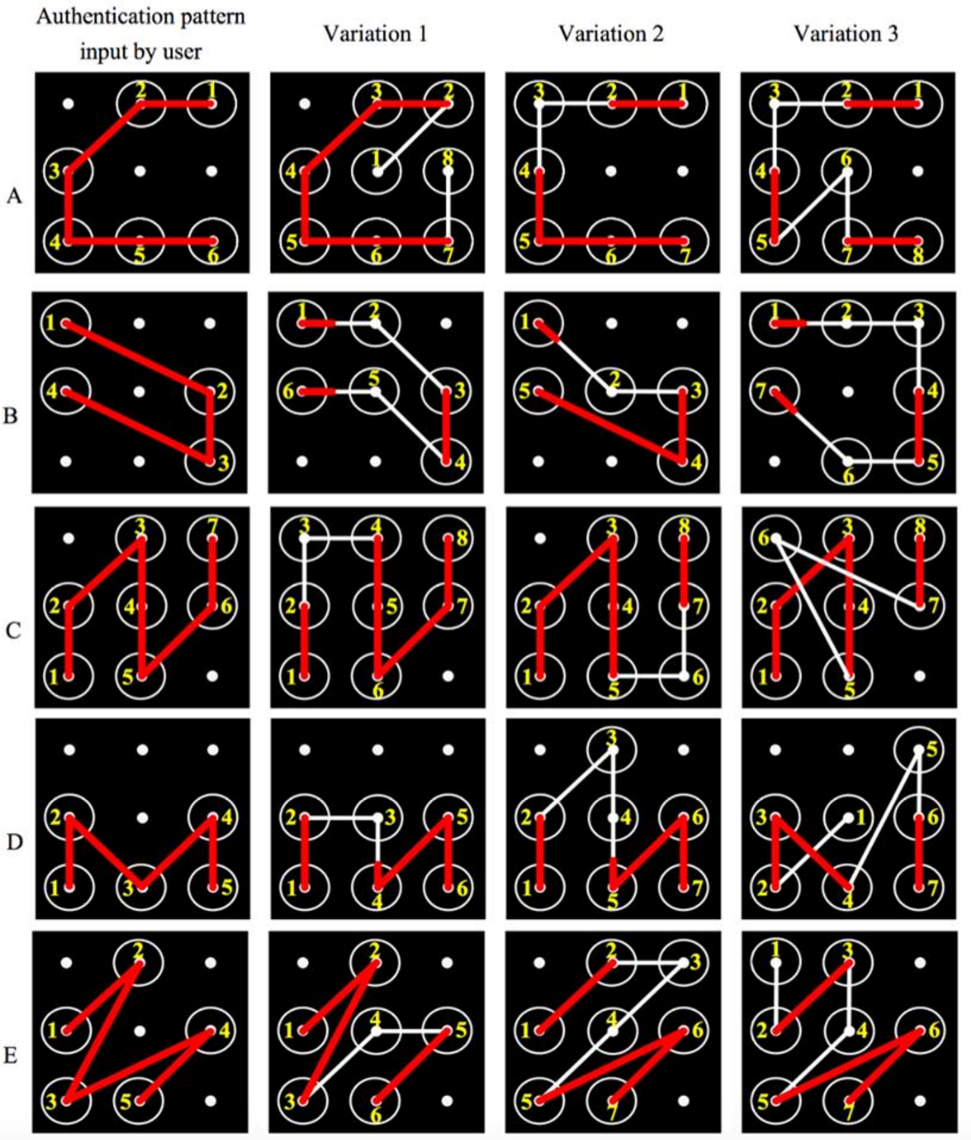

Figure 7. Five patterns and corresponding variations

During the experiment, the number of times that the users correctly entered the variations and the amount of time that the users spent to draw each pattern were recorded. These data were then used to calculate the average success rate of all users as well as that of a single user for each and every pattern (as shown in Figure 8 and Figure 9). The success rate for each variation is shown in Figure 8, in which complex patterns and/or patterns containing two successive dots located far from each other (i.e., top-left dot to mid-bottom dot) showed lower success rates than those of others; the reason why the latter displayed lower success rates was that users were more prone to touching the mid-left and/or center dot when moving from the top-left dot to the mid-bottom dot). Figure 9 shows the success rates of the users for the different patterns, in which three users demonstrated lower success rates. We found that the average heavy pressure value of Users 3 and 10 were comparatively lower, which contributed to higher error rates when drawing the patterns for authentication. User 4 showed a lower success rate because the user displayed higher standard deviations in pressure values compared with those of other users during the pressure experiment. This indicated that the said user possessed unstable pressure values and that the fluctuations in pressure values were more likely to cause authentication failure.

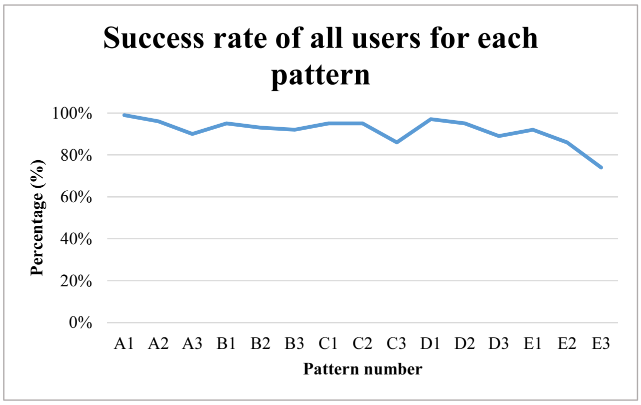

Figure 8. Success rate of all users for each pattern

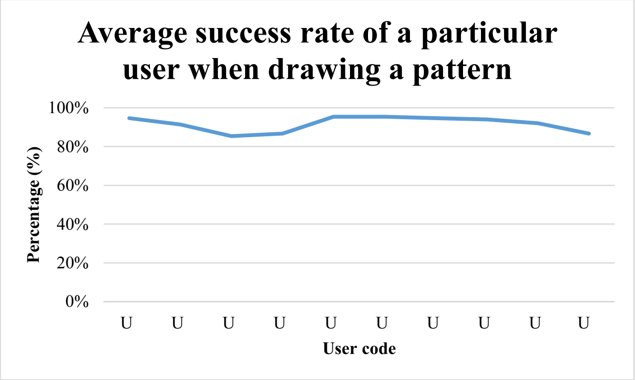

Figure 9. Average success rate of a particular user when drawing a pattern

Table 1 shows that in general, users who employed our method were able to log in successfully 92% of the time. By removing the variation with the lowest success rate for each pattern, the aforementioned log in success rate was elevated to 94%. Likewise, by removing the two users whose average pressure values were similar to the threshold value as well as that who exhibited unstable pressure values, the log in success rate was increased to 94%. These results showed that the success rates increased by 2% by removing the "oddities" and that users employing our method were able to achieve high success rates.

Table 1. Number of successes and average time

Patterns	All Patterns	When variation with the worst success rate is removed for each pattern
Success rate	92%	94%

Users	All users	When the three users with the lowest success rates are removed
Success rate	92%	94%

Concerning the amount of time that the users spent to log in to the system, it was divided into "average amount of time that all users spent to draw each pattern" (Figure 10) and "average amount of time that a particular user spent to draw a pattern" (Figure 11). The amount of time that the users spent to draw the variations showed that the more complex the variations were, the more time that the users spent to draw them. On average, each pattern took the users approximately 2.5-4.5 s to complete. The time spent by the users had no effect on the success rate (i.e., an increase in time spent did not result in a higher success rate). For example, on average, Users 1 and 6 spent less time than User 5 to draw the patterns; however, the former displayed similar success rates compared with that of the latter. Such a finding revealed that the primary factor influencing success rate was pressure value. Figure 4.6 shows that on average, the users spent approximately 2.7-3.5 s to draw the patterns, indicating that the users required merely minimal time to complete authentication and that the use of our method did not create extra burden on the users. The users also were not required to learn new methods; they merely had to combine our method to the authentication method that they were familiar with.

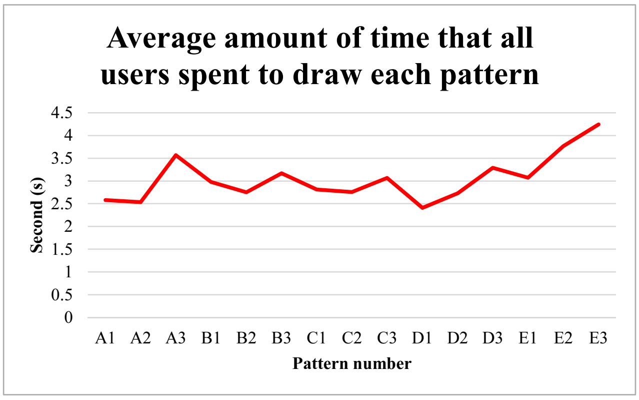

Figure 10. Average amount of time that all users spent to draw each pattern

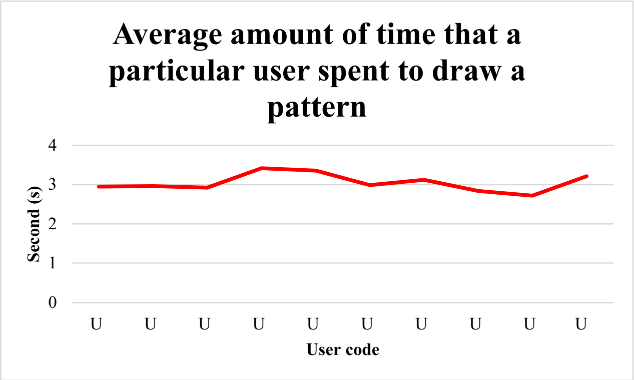

Figure 11. Average amount of time that a particular user spent to draw a pattern

Figure 12 shows the possible guesses that an attacker may make after observing a user enter the pattern shown in A1 (Figure 12), whereas Figure 13 shows the possible guesses that an attacker may make after observing a user enter the pattern shown in C2 (Figure 13). Because the number of possible combinations is not solitary regardless of the patterns observed, the attackers are unable to decipher the true patterns by simply engaging in shoulder-surfing attacks. Figure 14 shows the possible guesses that an attacker may make after recording a user enter his/her password on three separate occasions (i.e., patterns A1, A2, and A3) and then using the repeated dots to infer the possible authentication patterns. Conversely, Figure 15 shows the possible guesses that an attacker may make after recording a user enter his/her password on three separate occasions (i.e., patterns B1, B2, and B3) and then using the repeated dots to infer the possible authentication patterns. Because B1, B2, and B3 contained only four repeated dots (which is also the minimum number of dots required to unlock Android screen pattern locks), they produced only two possible combinations. This result showed that the lower the number of repeated dots is, the lower the number the possible combination becomes. Therefore, when users enter their passwords, they are recommended to include a higher number of repeated dots to increase the number of possible combinations to elevate the level of security against recording attacks.

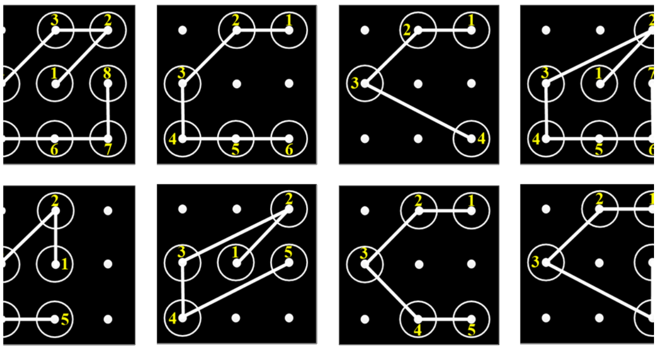

Figure 12. Attacker's guess of possible authentication patterns using the pattern from A1

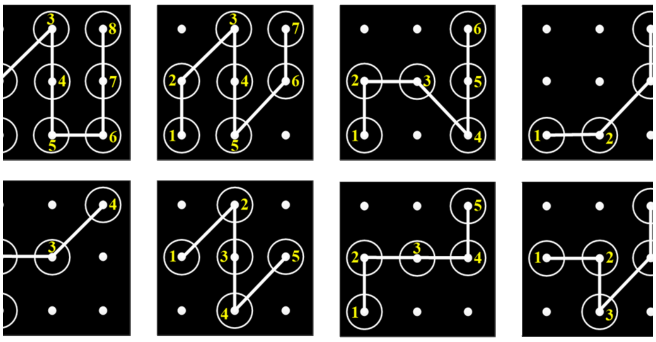

Figure 13. Attacker's guess of possible authentication patterns using the pattern from C2

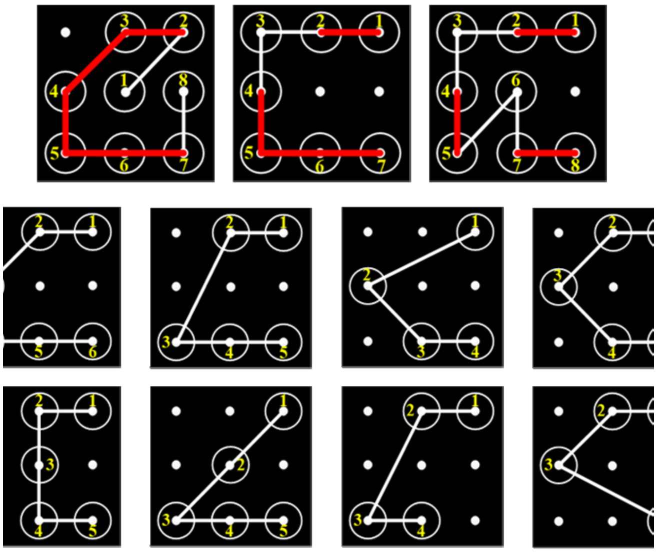

Figure 14. Attacker's guess of possible authentication patterns using the patterns from A1, A2, and A3

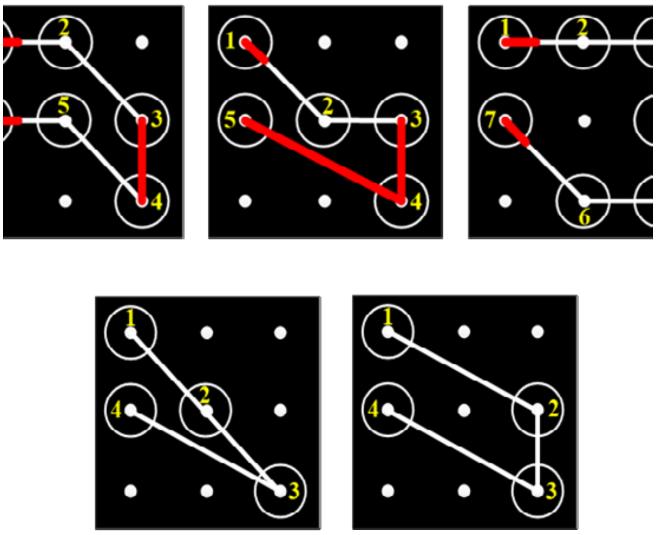

Figure 15. Attacker's guess of possible authentication patterns using the pattern from B1, B2, and B3

5 Conclusion

In this paper, we propose a novel authentication mechanism based on the measurement of smart phone's pressure sensor. Such pressure values were combined with identity authentication methods currently employed by smartphones to protect users from shoulder-surfing attacks. Because attackers are unable to differentiate between true and misleading information, the proposed authentication system can effectively defend users from shoulder-surfing attacks.

We integrate the proposed system with Android screen pattern locks, because the number of possible authentication patterns is more than one according to the pattern observed by an attacker, he/she is unable to decipher the user's true authentication pattern. This means that the attacker is unable to see the user's password even if he/she engages in shoulder surfing. Concerning our system authentication process, the success rate of users who passed authentication while incorporating misleading information into their passwords was 92%. In addition, we found that when users include a higher number of repeated dots in their patterns or use the same input pattern, it increases the number of possible combinations and subsequently elevates the level of security against recording attacks.

The method introduced in this study can be used by any touchscreen-based authentication system operated by fingers. The method protects users from leaking their text-based passwords and Android screen pattern locks when they are shoulder surfed by attackers, elevating their information security. In addition, attackers are unable to decipher the true passwords even if they have successfully observed the complete passwords.

Acknowledgements

This research was partially supported by the National Science Council of Taiwan under grant no. MOST 107-2218-E-011-012-, MOST 107-2221-E-033-010-, and MOST 107-2221-E-130-001-.

References

[1] P. Corcoran, C. Costache, Smartphones, Biometrics and a Brave New World, *IEEE Technology and Society Magazine*, Vol. 35, No. 3, pp. 59-66, September, 2016.

[2] A. F. Abate, M. Nappi, S. Ricciardi, Smartphone Enabled Person Authentication based on Ear Biometrics and Arm Gesture, *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics*, Budapest, Hungary, 2016, pp. 3719-3724.

[3] M. J. Coakley, J. V. Monaco, C. C. Tappert, Keystroke Biometric Studies with Short Numeric Input on Smartphones, *Proceedings of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems*, New York, NY, 2016, pp. 1-6.

[4] A. D. Luca, A. Hang, F. Brudy, C. Lindner, H. Hussmann, Touch Me Once and I Know It's You! Implicit Authentication based on Touch Screen Patterns, *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Austin, TX, 2012, pp. 987-996.

[5] A. Roy, N. Memon, A. Ross, MasterPrint: Exploring the Vulnerability of Partial Fingerprint-based Authentication Systems, *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 9, pp. 2013-2025, September, 2017.

[6] I. Altiok, S. Uellenbeck, T. Holz, GraphNeighbors: Hampering Shoulder-Surfing Attacks on Smartphones, *Sicherheit 2014*, Vienna, Austria, 2014, pp. 25-35.

[7] Y. Chen, W. Ku, Y. Yeh, D. Liao, A Simple Text-based Shoulder Surfing Resistant Graphical Password Scheme, *Proceedings of the 2013 IEEE International Symposium on Next-Generation Electronics*, Kaohsiung, Taiwan, 2013, pp. 161-164.

[8] M. K. Lee, Security Notions and Advanced Method for Human Shoulder-surfing Resistant PIN-entry, *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 4, pp. 695-708, April, 2014.

[9] M. Lei, Y. Xiao, S. V. Vrbsky, C. C. Li, L. Liu, A Virtual Password Scheme to Protect Passwords, *Proceedings of the 2008 IEEE International Conference on Communications*, Beijing, China, 2008, pp. 1536-1540.

[10] H. Shin, D. Kim, J. Hur, Secure Pattern-based Authentication against Shoulder Surfing Attack in Smart Devices, *Proceedings of 2015 7th International Conference on Ubiquitous and Future Networks*, Sapporo, Japan, 2015, pp. 13-18.

[11] K. Kiruthika, D. Jennifer, K. Sangeetha, C. Jackulin, R. Shalini, A Secure Pin Authentication Method against Shoulder Surfing Attacks, *International Journal of Engineering and Computer Science*, Vol. 5, No. 10, pp. 18707-18713, October, 2016.

[12] D. Choi, C. Choi, X. Su, Invisible Secure Keypad Solution Resilient against Shoulder Surfing Attacks, *Proceedings of the 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, Fukuoka, Japan, 2016, pp. 514-517.

[13] S. Rajarajan, K. Maheswari, R. Hemapriya, S. Sriharilakshmi, Shoulder Surfing Resistant Virtual Keyboard for Internet Banking, *World Applied Sciences Journal*, Vol. 31, No. 7, pp. 1297-1304, July, 2014.

[14] T. Kwon, S. Na, S. Park, Drag-and-Type: A New Method for Typing with Virtual Keyboards on Small Touchscreens, *IEEE Transactions on Consumer Electronics*, Vol. 60, No. 1, pp. 99-106, February, 2014.

[15] S. Kim, J. Kim, S. Kim, H. Cho, A New Shoulder-surfing Resistant Password for Mobile Environments, *Proceedings of the ACM ICUIMC 2011*, Seoul, Korea, 2011, pp. 1-8.

[16] T. Perkovic, M. Cagalj, N. Rakic, SSSL: Shoulder Surfing Safe Login, *17th International Conference on Software, Telecommunications & Computer Networks (SoftCOM 2009)*, Hvar, Croatia, 2009, pp. 270-275.

[17] A. Bianchi, I. Oakley, V. Kostakos, D. S. Kwon, The Phone Lock: Audio and Haptic Shoulder-surfing Resistant PIN Entry Methods for Mobile Devices, in *Proceedings of the 5th International Conference on Tangible, Embedded, and*

Embodied Interaction, Funchal, Portugal, 2011, pp. 197-200.

[18] H. Saevanee, P. Bhatarakosol, User Authentication using Combination of Behavioral Biometrics over the Touchpad Acting Like Touch Screen of Mobile Device, *2008 International Conference on Computer and Electrical Engineering (ICCEE 2008)*, Phuket, Thailand, 2008, pp. 82-86.

[19] Y. Kita, F. Sugai, M. Park, N. Okazaki, Proposal and Its Evaluation of A Shoulder-surfing Attack Resistant Authentication Method: Secret Tap with Double Shift, *International Journal of Cyber-Security and Digital Forensics*, Vol. 2, No. 1, pp. 48-55, March, 2013.

[20] N. Chakraborty, S. Mondal, SLASS: Secure Login against Shoulder Surfing, *Recent Trends in Computer Networks and Distributed Systems Security*, Trivandrum, India, 2014, pp. 346-357.

[21] H. Yi, Y. Piao, J. H. Yi, Touch Logger Resistant Mobile Authentication Scheme using Multimodal Sensors, in: H. Jeong, M. S. Obaidat, N. Yen, J. Park (Eds.), *Advances in Computer Science and its Applications. Lecture Notes in Electrical Engineering, Vol 279*, Springer, Berlin, 2014, pp. 19-26.

[22] T. Chang, C. Tsai, J. Lin, A Graphical-based Password Keystroke Dynamic Authentication System for Touch Screen Handheld Mobile Devices, *Journal of Systems and Software*, Vol. 85, No. 5, pp. 1157-1165, May, 2012.

[23] Y. Meng, D. Wong, L. Kwok, Design of Touch Dynamics based User Authentication with An Adaptive Mechanism on Mobile Phones, in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, Gyeongju, Korea, 2014, pp. 1680-1687.

Biographies

Jia-Ning Luo holds a Ph.D. degree in Computer Science of National Chiao Tung University, Taiwan. He specializes in network security, operating systems, network administration and network programming. He is currently working on NFC-based protocols; RFID ownership transfer; eWallet security.

Ming-Hour Yang received her master degree in Information & Telecommunications Engineering at Ming Chuan University, Taiwan. His research mainly focuses on network security and system security with particular interests on security issues in RFID and NFC security communication protocols. Topics include: mutual authentication protocols; secure ownership transfer protocols; polymorphic worms; tracing mobile attackers.

Cho-Luen Tsai received her master degree in Information & Telecommunications Engineering at Ming Chuan University, Taiwan.