PAPER

A Series of PIN/Password Input Methods Resilient to Shoulder Hacking Based on Cognitive Difficulty of Tracing Multiple Key Movements

Kokoro KOBAYASHI[†], *Student Member*, **Tsuyoshi OGUNI**[††], *Nonmember*, *and* **Masaki NAKAGAWA**[†a)], *Fellow*

SUMMARY This paper presents a series of secure PIN/password input methods resilient to shoulder hacking. When a person inputs a PIN or password to a smartphone, tablet, banking terminal, etc., there is a risk of shoulder hacking of the PIN or the password being stolen. To decrease the risk, we propose a method that erases key-top labels, moves them smoothly and simultaneously, and lets the user touch the target key after they stopped. The user only needs to trace a single key, but peepers have to trace the movements of all the keys at the same time. We extend the method by assigning different colors, shapes, and/or sizes to keys for enhancing distinguishability, which allows all the keys to be moved instantaneously after key-top labels are erased and the user to touch the target key. We also introduce a "move backward/forward" function that allows the user to play back the movements. This series of methods does not have the highest security, but it is easy to use and does not require any changes to the server side. Results of a performance evaluation demonstrate that this method has high resistance to shoulder hacking while providing satisfactory usability without large input errors.

key words: *PIN code, password, user authentication, shoulder hacking, cognitive difficulty*

1. Introduction

A Personal Identification Number (PIN) is a secret sequence of digits to authenticate the user and protect against illegal access to the information or resources possessed by the user. A similar role is played by a password, which is composed of a secret sequence of characters. We can consider PINs as a type of password here and discuss PINs inclusively. In daily life, passwords are increasingly being used to authenticate user access to ATMs, to pay by a credit card, to open up a smartphone/tablet, to enter computer and network services, and so on.

With the proliferation of smartphones and tablets these days, passwords are increasingly being input in public spaces such as trains and buses where they are likely to be exposed to public eyes. Moreover, ATMs are now prevalent in convenience stores, shopping centers, and other public places where security is lower than within a bank, so again passwords are more open to the risk of being peeped by other persons.

An instance where a password is peeped by others over the victim's shoulder (or from the reflection off glass) is called shoulder hacking or surfing. Once this happens, the information or resource possessed by the user is subject to illegal access or attack.

In this paper, we propose a series of methods for secure password input against shoulder hacking that requires less mental load for the user while incurring cognitive difficulties for peepers. It is not resilient to video recording, but can easily be made so by introducing another secret or calculation. Moreover, it does not require any changes to the hardware and software on the server side. Its effectiveness is demonstrated through evaluation experiments.

This paper combines two preceding conference publications [20], [21] and formulates them into a series of methods with an added evaluation. Sect. 2 presents related works and clarifies the position of our approach among others. Our basic method and its extensions are presented in Sect. 3, and Sect. 4 reports their evaluation. Sect. 5 considers the results. Sect. 6 describes a further extension and its evaluation. We conclude in Sect. 7 with a brief summary and mention of future work.

2. Related Works

Several technologies have been proposed or invented to protect password input from shoulder hacking. Randomizing key allocations every time a key is pushed may prevent the key from being read by the positions of the user's arm and finger, thus providing resilience against the so-called replay attack, but it forces the user to search for the key every time [1]. Tanaka et al. proposed a method to allocate keys from 0 to 9 circularly while marking the 0 key in a different color and rotating the keys every time a number is input [2]. This allows the user to find the target key easily while defending from the replay attack. Makita et al. proposed another replay-attack resilient method that displays the input panel partially and has the user scroll it to show and push the desired key [3]. This is more suitable for larger keyboards than the smaller ten numerical keypads. Kakinuma et al. proposed another method within the category of graphical passwords [4] that utilizes a sequence of colors as a password and lets the user touch the color appearing in a presented picture in the sequence of the password. The picture is altered every time so that guessing the password from the finger and arm position is difficult. However, its sole depen-

Manuscript received June 27, 2019.
Manuscript revised December 21, 2019.
Manuscript publicized April 6, 2020.
[†]The authors are with Tokyo University of Agriculture and Technology, Koganei-shi, 184–8588 Japan.
[††]The author is with NTT DATA, Tokyo, 108–0075 Japan.
a) E-mail: nakagawa@cc.tuat.ac.jp
DOI: 10.1587/transinf.2019EDP7181

dence on color excludes users with color vision deficiencies. Moreover, some software additions must be made to adapt it for use in conventional password authentication systems. In summary, the methods described above are resilient to the replay attack but not to peeping at the keys being pushed. Hidaka et al. proposed a technique that is resistant to key-logging, peeping and even video recording, provided that the four arrow keys to move to the intended key are smaller and can be hidden from a camera [5].

Sakurai et al. proposed a method that is not only resilient to the replay attack but also to peeping [6]. It classifies characters for passwords into several groups, and for every character in a password, the user searches for the character, finds the group that includes it, and selects a random number assigned every time to the group. While this method is peeping-resilient, it is not resilient to video recording in so far as the groups are fixed. Moreover, the robustness to random attack is decreased since all characters within a group are accepted as the correct character. They additionally proposed an extension that lets the user input the sum of two random numbers assigned to two consecutive characters in the password, which makes it harder for peepers to guess the password. However, the user has to search for password characters and perform quick mental calculations, which increases his/her mental load. KyuChoul et al. invented another method [7] that randomizes the key arrangement for a password and then lets the user push the key displaced to the fixed direction with the fixed distance from the target key for each password character. The direction and distance of the displacement is identified from the first character "*" of the password. This method is resilient to replay and peeping attacks, but not to video recording.

Takada et al. proposed a video recording resilient method that introduces "fakePointer" in addition to a password [8], [9]. fakePointer is a mask that may point to several keys. The user manipulates a specific position in the mask to point to a password character and repeats this to input the password. Its specific position is secret and peepers cannot identify which key is selected. However, since the characters are limited in fakePointer, the password is confined within a certain sequence. To avoid this, the method is extended to interleave false characters, which can be detected by the system. It is resilient to peeping and video recording but introduces another secret to remember. Kita et al. proposed another video recording resilient method that displays graphical password keys on a 4x4 grid and the user input keys on positions shifted from the target keys by a secret amount [10]. The drawback of this method is that the user needs to make a mental calculation to locate the shifted positions every time. Watanabe et al. proposed another video recording resilient method that introduces "cursor camouflage" [11]. It shows multiple dummy cursors moving in random directions, and while the user can find the real cursor by comparing with the mouse movement, potential peepers cannot identify it. It is resilient to peeping and video recording but it imposes a burden on the user to find the real cursor. Luca et al. proposed a similar method [12].

Information theoretic methods have also been proposed [13]–[15]. They are resilient even to video recording, but introduce additional secrets and require complex mental operations.

Another stream of authentication is emerging in the form of biometric information such as fingerprint, iris, retinal pattern, finger or palm vein pattern, face, speech, and handwriting [16], [17]. Biometric information is unlikely to be forgotten or stolen compared to passwords or physical objects, and implementing it is both easy and user-friendly. Therefore, it is getting popular for a variety of devices and services. Once such information is stolen, however, it cannot be recovered. Moreover, authentication of the true user may fail due to noises, and false users may pass through a gate as a result of various errors. After passing through authentication by biometrics, however, the user still has to type in identification and password details to enter many services.

Another stream of research has focused on reducing the mental load of the user, though most of the methods are not resilient to video recording. Roth et al. proposed a method [18] that colors half of the ten numerical keys black and the other half white. The user selects the color of the key that he/she wants to input, and then the system scrambles the keyboard to show a different coloring and the user selects the color again. When this is repeated four times, the key is identified uniquely. User testing showed that this method is resilient to peeping, but it takes about ten times as long as entering simple PINs on a number pad. In order to enhance its recording resilience, they proposed reducing color inputs to less than four times and making the PIN number unidentifiable uniquely. The authentication is allowed if the correct one is within the probable candidates. Tan et al. proposed a software keyboard that displays 42 keys and two "Interactor Tiles" at the bottom of the keyboard [19]. Just as each key on a standard keyboard represents two characters, each key is randomly assigned a lowercase letter (on the top row with red background), an uppercase letter (middle with green background), and either a number or a symbol (bottom with blue background). Rather than having a fixed shift state for the entire keyboard, each key has a randomly assigned shift state, indicated by the red line under the active character. In order to select a character, the user first locates the key containing the character to be typed. Next, the user clicks on one of the Interactors to cycle through shift states and move the red underline to the desired character. Finally, the user drags the Interactor to the key on which the desired character resides. Upon the start of the drag interaction, the system blanks all key-top labels. Without knowing where the user is going to drop the Interactor, adversarial observers have to memorize the locations of all characters on the keyboard. The keyboard re-randomizes characters and the user repeats the process to select the next character. The results of a user study conducted on a digital whiteboard showed that, when 8-character passwords were input, the security level was highly improved (a magnitude) while the input time was just doubled in comparison with a common soft keyboard. As their future work, they appended an idea to

move multiple keys, but it has not been evaluated yet.

In this paper, we propose a series of methods for secure password input against replay-attack and peeping in shoulder hacking. It requires less mental load for the user while incurring cognitive difficulties for peepers.

We assign colors, shapes, and/or various sizes to keys in a keypad/keyboard, erase key-top labels, move them simultaneously, and let the user touch the target key. Peepers have cognitive difficulty in tracing the movements of all the keys at the same time, but the user only needs to trace a single key and touch it. An extension of this method is to move all the keys instantaneously after erasing key-top labels and let the user touch the target key. Another extension is to introduce a "move backward/forward" function for the user to confirm the traces of movements. It is not resilient to video recording, but it can easily be made so by introducing another secret or calculation in similar ways as [6], [8], [9], [18]. A simple example is to let the user touch a key displaced by an agreed distance from the correct key. In this paper, however, we limit our focus to presenting a new dimension for defending against shoulder hacking.

3. Basic Method and Extensions

In this section, we propose a series of password input methods that require less mental load for the user while incurring cognitive difficulties for peepers, thus providing resilience to replay-attack and peeping. We have named this series of methods Secure Pad.

3.1 Basic Method

When there is no risk of shoulder hacking, the user inputs a password character by touching displayed keys directly. When there is a risk, however, the user triggers the function of Secure Pad by tapping the "shuffle" button. Secure Pad then erases the key-top labels, moves them smoothly and simultaneously, and lets the user touch the target key after they stopped, as shown in Fig. 1. Meanwhile, the shuffle button is renamed as "retry" to let the user retry the process if the target is lost. Peepers are expected to find it cognitively difficult to trace the movements of over four objects at the same time [22], [23], but the user needs only to trace a single target key and touch it without having to remember another secret or to make any calculation. Therefore, we discard key-movement candidates when fewer than four keys overlap while moving. This can be used without any special hardware and without any changes to the server side.

We should point out that Secure Pad does not provide the highest security; specifically, it is not resilient to video recording. It can easily be made so, however, by introducing another secret or calculation in similar ways as [6], [8], [9], [18].

3.2 Extensions

We can assign different colors, shapes, and/or sizes to keys

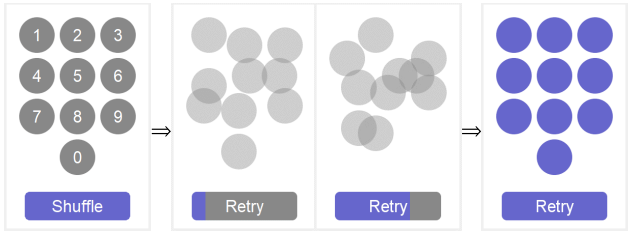

Fig. 1 Secure Pad display: (a) initial state, (b) erasing key-tops, (c) moving them smoothly and simultaneously, and (d) stopped state for accepting key-tap. The retry button initiates another cycle when the target key is lost.

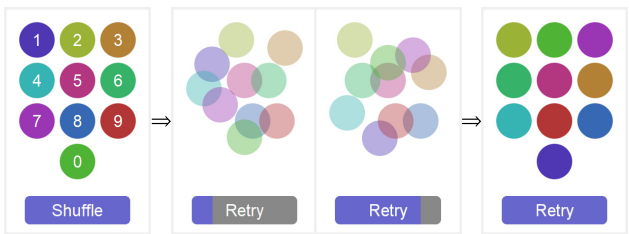

Fig. 2 Secure Pad display with various colors.

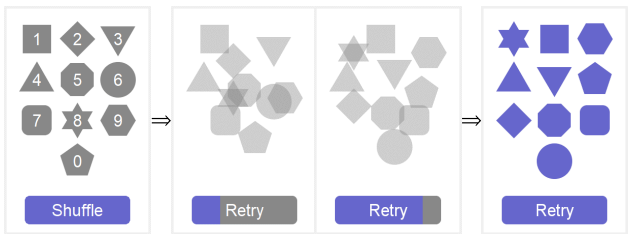

Fig. 3 Secure Pad display with various shapes.

for enhancing distinguishability, as shown in Figs. 2 and 3. Note that colors should not be exclusive extensions because many people have difficulty distinguishing colors.

Enhanced distinguishability due to different colors, shapes, and/or sizes allows all the keys to be moved instantaneously after key-top labels are erased and the user to touch the target key.

3.3 Dimensions of Extensions

To summarize the above extensions, the several dimensions of variations for Secure Pad are listed in Table 1. Key color may include variations of texture, figure, or even pictures on the tops of keys. However, a set of colors undistinguishable by people with color weakness should be avoided. In such a case, gray level variations could be utilized. We can combine variations of key color, key shape, key size, and key movement to enhance distinguishability, but this may lower the difficulty of peepers to trace key movements.

The combination of variations can be applied for both the ten numerical keypads and the alphanumeric (QWERTY) keypads. Figure 4 shows the color and shape variations applied for the latter.

Table 1 Dimensions of variations for Secure Pad.

Dimension	Variation	Detail
Key color	Single color	Single color for all keys.
	Multiple colors	Different color for each key.
Key shape	Single shape	Single shape for all keys (e.g., circle, square).
	Multiple shapes	Different shape for each key (e.g., circle, polygon, star).
Key size	Single size	Single size for all keys.
	Multiple sizes	Different size for each key.
Key movement	Smooth	Move all keys smoothly and simultaneously.
	Instantaneous	Move all keys instantaneously and simultaneously.

Fig. 4 Secure Pad for the QWERTY keyboard.

4. Evaluation

This section presents our evaluation of the variations of Secure Pad through experiments on the robustness to peeping and usability.

4.1 Variations of Secure Pad

The purpose of the experiment is to evaluate the robustness to peeping by others and the user's ease of use of the Secure Pad variations. We prepared 12 variations of Secure Pad for the ten numerical keys (ten keys in short) and the QWERTY keys with regard to key color, key shape, and key movement as well as two benchmark key configurations, as shown in Table 2.

For color variations, we divided the hue into ten (for ten keys) or 36 (for QWERTY) at equal intervals while fixing the brightness and saturation (as all the participants in the experiment had normal color vision). Then, we assigned these different hues randomly to the keys. For shape variations, we utilized circles, upward triangles, downward triangles, squares, rounded squares, diamonds, pentagons, hexagons, octagons, and stars. We felt that more than ten different shapes would be too confusing. When combining color and shape variations for QWERTY, we chose three or four colors, at approximately equal hue intervals, and assigned them for each shape. We do not examine key size dimension here because we assumed it would have the same or less effect as the key color and key shape. As for key movement, we considered straight movement and set the duration of the smooth movement to 1 sec considering the balance between the difficulty of peepers tracing multiple keys and the user's ease of tracing the target key and time to input a password. For instantaneous movement, keys must be clearly distinguishable, and combination with single color, single shape, and single size is meaningless. When instantaneous movement was used for the QWERTY keys, we only tested the combination with multiple (36) colors and multiple (ten) shapes because color or shape variations alone seems hard to distinguish with 36 keys.

Table 2 List of keypads for evaluation.

Type \ Dimension	Key set	Key color	Key shape	Key movement
Benchmark 1	Ten keys	Single (Blue)	Single (Circle)	No movement
Benchmark 2	QWERTY			
Secure Pad 1	Ten keys	Single	Single	Smooth
Secure Pad 2		Multiple		
Secure Pad 3		Single	Multiple	
Secure Pad 4		Multiple		
Secure Pad 5		Multiple	Single	Instantaneous
Secure Pad 6		Single	Multiple	
Secure Pad 7		Multiple		
Secure Pad 8	QWERTY	Single	Single	Smooth
Secure Pad 9		Multiple		
Secure Pad 10		Single	Multiple	
Secure Pad 11		Multiple		
Secure Pad 12		Multiple	Multiple	Instantaneous

4.2 Details of Experiment

We formed a pair of participants—one as a user and one as a peeper—and changed their roles for each type of Secure Pad. Peepers were allowed to stand at the easiest distance from the display for peeping, which was about 30 cm on average. This is similar to the conditions on a crowded train, so the experiment should illuminate the worst-case scenario for peeping resilience. Table 3 lists the profiles of participants. The PINs used for Secure Pad with ten keys (Secure PIN Pad) were 4-digit numeric strings, and the passwords used for Secure Pad with the QWERTY keys (Secure QWERTY Pad) were 4-character alphanumeric strings. They were randomly generated for each pad and each role in a pair. We denote the sequence of actions where the user inputs a PIN or password and the peeper tries to read it as a trial. We asked each pair and role to perform three trials with the same password (note that PIN is included) on each type of Secure Pad. When the peeper succeeded in reading the password completely at the first or second trial, the subsequent trials are considered "success" and are skipped. In contrast, when the user retried inputting a single character three times, the input and the peeping conditions were marked as "failure" and the user was forced to input the next character. In each trial, we recorded whether the password was successfully peeped and the time required for actions. The experiment was performed using a 7-in 1024 x 600 tablet oriented horizontally without tilt.

Each pair took part in the following procedure:

(1). Listen to the explanation on how to use Secure Pad and the procedure for the experiment.
(2). Do a few practice runs on Secure Pad 1 and Secure Pad

Table 3 Experiment participants.

Pair no.	Age	Gender
1	22	Male
	23	Male
2	22	Male
	23	Male
3	22	Female
	21	Male
4	54	Female
	55	Male
5	59	Female
	59	Male

12 (QWERTY).

(3). Perform the trials on Benchmark 1 and Benchmark 2.

(4). Perform the trials on various types of Secure Pad. In order to eliminate bias due to the order of use, the types of Secure Pad used were randomized for each pair.

(5). Answer a simple questionnaire after completing the experiment.

4.3 Results

We present the results on the robustness to peeping, ease of input, input time, and verification.

4.3.1 Robustness to Peeping

Table 4 shows the average numbers of successful peeping of individual characters and the average rate of peeping all four characters for each type of keypad. Although the password was typically peeped in the 1st or 2nd trials with the benchmark keypads, which do not feature moving keys, Secure Pad was robust to peeping even in three trials with many types. With Secure PIN Pad for numeric keys, only one PIN was peeped in two trials, some were peeped in the third trial, and the number of characters successfully peeped was less than half in three trials. With Secure QWERTY Pad, no password was peeped in three trials, and only less than a single character was peeped with some types on average (at most two characters).

4.3.2 Ease of Input

Table 5 shows the average number of characters successfully input and the number of retries performed on each type of keypad. The former divided by four shows the input success rate. Participants in their 20s had no large difference in this rate between Secure Pad and the benchmarks, and their numbers of retries were small. In contrast, the input success rates were lower and the numbers of retries increased on Secure Pad for participants in their 50s. Moreover, instantaneous movement was liable to cause input failure. As the color and/or the shape variations were added under the same condition, however, the input success rate was improved and the number of retries decreased.

Table 4 Average number of peeped characters and rate of all four characters peeped.

Type \ Measure	Number of characters peeped			Rate of all four characters peeped		
	1st trial	2nd trial	3rd trial	1st trial	2nd trial	3rd trial
B1 (Ten, Kc:Sin, Ks:Sin, M:No)	4.00	4.00	4.00	1.00	1.00	1.00
B2 (Qw, Kc:Sin, Ks:Sin, M:No)	3.10	3.80	4.00	0.60	0.90	1.00
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	0.90	1.50	1.50	0.00	0.00	0.20
SP2 (Ten, Kc:Mul, Ks:Sin, M:S)	0.60	0.90	1.50	0.00	0.00	0.10
SP3 (Ten, Kc:Sin, Ks:Mul, M:S)	1.20	1.30	1.90	0.00	0.10	0.10
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	0.80	1.40	1.80	0.00	0.00	0.10
SP5 (Ten, Kc:Mul, Ks:Sin, M:I)	0.00	0.30	0.70	0.00	0.00	0.00
SP6 (Ten, Kc:Sin, Ks:Mul, M:I)	0.50	1.00	0.70	0.00	0.00	0.00
SP7 (Ten, Kc:Mul, Ks:Mul, M:I)	0.50	0.60	1.10	0.00	0.00	0.00
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	0.00	0.10	0.10	0.00	0.00	0.00
SP9 (Qw, Kc:Mul, Ks:Sin, M:S)	0.10	0.10	0.30	0.00	0.00	0.00
SP10 (Qw, Kc:Sin, Ks:Mul, M:S)	0.00	0.00	0.20	0.00	0.00	0.00
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	0.10	0.20	0.30	0.00	0.00	0.00
SP12 (Qw, Kc:Mul, Ks:Mul, M:I)	0.10	0.10	0.20	0.00	0.00	0.00

Under "Type", B and SP denote Benchmark and Secure Pad, Ten and Qw denote Ten keys and QWERTY, KC:Sin and KC:Mul denote key color being single and multiple, KS:Sin and KS:Mul denote key shape being single and multiple, and M:No, M:S, and M:I denote movement being no movement, smooth, and instantaneous, respectively.

4.3.3 Input Time

Table 6 shows input time, where the "Time" column shows the average time (in sec) taken to input all four characters on each type of keypad and the "Time/char." column shows the average time per character from pushing the shuffle button to key input. Note that B1 and B2 do not have the shuffle button, so there is no value for the latter column. For Secure Pad, the value in the Time column does not equal four times the value in the Time/char. column since the former includes the time from key input to the next shuffle and that for retries.

With Secure Pad, it takes lager input time. It is from 2.8 to 11.5 times compared with the benchmarks (27.34 sec on SP8 v.s. 9.70 sec on B2 to 38.62 sec on SP4 v.s. 3.34 sec on B1 by participants of 50s). Moreover, Secure QWERTY Pad took a long time for users in their 50s (discussed in more detail later).

4.3.4 Verification

We performed paired t-testing on the number of characters successfully peeped, the rate of all four characters being

Table 5 Average number of successfully input characters and number of retries.

Type \ Group of participants & measure	All participants		Age: 20s		Age: 50s	
	No. of chars.	No. of retries	No. of chars.	No. of retries	No. of chars.	No. of retries
B1 (Ten, Kc:Sin, Ks:Sin, M:No)	3.97	N/A	4.00	N/A	3.92	N/A
B2 (Qw, Kc:Sin, Ks:Sin, M:No)	3.87	N/A	3.94	N/A	3.75	N/A
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	3.80	0.13	3.78	0.00	3.83	0.33
SP2 (Ten, Kc:Mul, Ks:Sin, M:S)	4.00	0.07	4.00	0.00	4.00	0.17
SP3 (Ten, Kc:Sin, Ks:Mul, M:S)	3.97	0.03	4.00	0.00	3.92	0.08
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	4.00	0.00	4.00	0.00	4.00	0.00
SP5 (Ten, Kc:Mul, Ks:Sin, M:I)	3.77	0.30	3.94	0.28	3.50	0.33
SP6 (Ten, Kc:Sin, Ks:Mul, M:I)	3.67	0.33	3.67	0.22	3.67	0.50
SP7 (Ten, Kc:Mul, Ks:Mul, M:I)	3.93	0.10	3.94	0.00	3.92	0.25
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	3.80	0.27	3.94	0.00	3.58	0.67
SP9 (Qw, Kc:Mul, Ks:Sin, M:S)	3.80	0.33	3.94	0.28	3.58	0.42
SP10 (Qw, Kc:Sin, Ks:Mul, M:S)	3.80	0.27	3.94	0.06	3.58	0.58
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	3.90	0.10	3.94	0.11	3.83	0.08
SP12 (Qw, Kc:Mul, Ks:Mul, M:I)	3.93	0.10	3.94	0.06	3.92	0.17

Table 6 Average input time (sec).

Type \ Group of participants & measure	All participants		Age: 20s		Age: 50s	
	Time	Time/ char.	Time	Time/ char.	Time	Time/ char.
B1 (Ten, Kc:Sin, Ks:Sin, M:No)	2.81	–	2.51	–	3.34	–
B2 (Qw, Kc:Sin, Ks:Sin, M:No)	4.60	–	1.53	–	9.70	–
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	15.75	2.06	10.80	1.77	22.76	2.47
SP2 (Ten, Kc:Mul, Ks:Sin, M:S)	18.60	2.18	9.54	1.76	32.18	2.81
SP3 (Ten, Kc:Sin, Ks:Mul, M:S)	20.88	2.10	9.89	1.75	36.45	2.60
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	21.35	2.21	9.84	1.75	38.62	2.91
SP5 (Ten, Kc:Mul, Ks:Sin, M:I)	19.89	1.71	10.41	1.39	34.12	2.20
SP6 (Ten, Kc:Sin, Ks:Mul, M:I)	21.80	2.04	13.12	1.69	34.81	2.56
SP7 (Ten, Kc:Mul, Ks:Mul, M:I)	20.17	1.49	9.42	1.06	36.30	2.14
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	20.55	2.31	15.75	1.97	27.34	2.79
SP9 (Qw, Kc:Mul, Ks:Sin, M:S)	22.85	2.26	15.63	1.87	34.65	2.84
SP10 (Qw, Kc:Sin, Ks:Mul, M:S)	22.28	2.31	14.00	1.89	34.70	2.94
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	22.59	2.33	12.05	1.83	38.40	3.08
SP12 (Qw, Kc:Mul, Ks:Mul, M:I)	24.79	2.56	16.70	2.24	36.92	3.05

peeped, the input time, and the number of characters successfully input between each type of keypad and the benchmarks (B1 or B2). The number of characters successfully peeped and the rate of all four characters being peeped were significantly smaller with $p < 0.001$, which supports the peeping resilience of Secure Pad in these respects. On the other hand, the input time was significantly larger with $p < 0.05$, while the number of characters successfully input was not significantly different with $p > 0.01$.

We also preformed paired t-testing between the smooth movements (SP2, SP3, SP4 and SP11) and the instantaneous movements (SP5, SP6, SP7 and SP12) under the same conditions. Specifically, we took the *n-th* (n=1 to 3) trial of a pair of participants for SP2 and the *n-th* (n=1 to 3) trial of the same pair of participants for SP5. We repeated this for SP3 and SP6, for SP4 and SP7 and for SP11 and SP12. Then, we applied paired t-testing for all of these pairs. The number of characters successfully peeped and the rate of all four characters being peeped by the instantaneous movements were significantly smaller than those by the smooth movements ($p < 0.05$), which shows that the peeping resilience of the instantaneous movements is stronger than that of the smooth movements. On the other hand, the number of characters successfully input by the smooth movements was significantly larger ($p < 0.01$), which shows that the ease of use of the smooth movements is better compared to that of the instantaneous movements. For the input time, no significant difference was observed ($p > 0.05$).

4.3.5 Feedback from the Participants

We received the following opinions from the participants after the experiment:

- When a single color and shape is used, neither inputting nor peeping are easy.
- Instantaneous movement is difficult both to trace and to peep from a single observation.
- Ease of peeping depends on the distance of key movement.
- Without shape variation, the user is not confident in deciding the target key.
- When the target key and surrounding keys are similar in shape and color, the user is confused in tracing the target key.
- When movements cross over, both tracing and peeping are difficult.
- It takes time to input all four characters.

4.4 Considerations

The experimental results, as shown in Table 5 and discussed in Sect. 4.3.1, demonstrate that Secure Pad is robust to peeping.

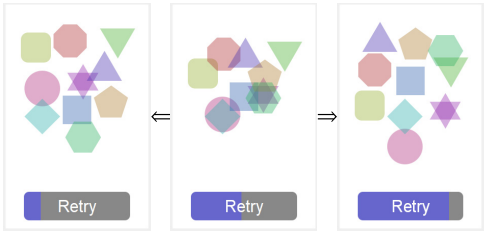

Fig. 5 Secure Pad display with various shapes.

However, the success rate of inputting a password character dropped when a single color and shape were specified. In addition, instantaneous movement was liable to cause input failure, but failures could be prevented by the color and shape information. Likewise, the number of retries decreased when there were more color and shape variations. These results suggest that the user's mental load is not excessively increased by the color and shape information, compared with the benchmark keypads that do not feature moving keys.

As for the input time, it took several times with Secure Pad than with the benchmarks. This is the price of enhancing the security, the same as with other methods [10], [16]. In Secure Pad, however, users can touch keys without having to move them, which means they can shorten the input time when there is no need to worry about security. It took users in their 50s a longer time with the Secure QWERTY pad, presumably because two of them were not accustomed to using the QWERTY keyboard.

A comparison between the smooth movements and the instantaneous movements shows that the instantaneous movements have higher peeping resilience but the input success rate deteriorates. Each has advantages and disadvantages so that an appropriate method can be chosen according to the required peeping resilience and the ease of use.

5. Further Extension and Evaluation

In the above versions of Secure Pad, users had to retry the key movements whenever they lost the target. One way to decrease the number of retries would be to provide a move backward/forward (move b/f for short) function, but the surface of Secure Pad is a little too small to add an extra button or slider. Therefore, we combine this function with the "retry" button to let the user confirm movement traces by sliding it to the left (move the keys backward) or right (move them forward). The button works as a retry button when it is tapped and works as a moving b/f slider when it is dragged, as shown in Fig. 5. Its leftmost side corresponds to the initial key arrangement and its rightmost side to the final arrangement after key movements. We expect this modification to increase the input success rate while decreasing the number of retries.

6. Evaluation of the Extension

We evaluated whether the number of successfully input characters increased when the retry is dispensed and to what extent the resilience to peeping attack is maintained by the move b/f function.

Table 7 List of keypads for evaluation.

Type \ Dimension	Key set	Key color	Key shape
Secure Pad 1	Ten keys	Single	Single
Secure Pad 4		Multiple	Multiple
Secure Pad 8	QWERTY	Single	Single
Secure Pad 11		Multiple	Multiple

Table 8 Participants.

Pair no.	Age	Gender
1	22	Male
	23	Male
2	22	Female
	21	Male
3	59	Female
	59	Male

6.1 Details of Experiment

We selected four types of Secure Pad for the ten keys and QWERTY with and without the same color variations and the same shape variations as the basic method, as shown in Table 7 We formed a pair of participants in the same way as the experiment for the basic method. Table 8 shows the profiles of participants.

In order to simulate the situation where the user loses the traces of key movements, the user was asked to start key movements but not to observe them while the peeper followed the traces until they stopped. Then, the user operated the retry slider to view the key movements and touched the target key. The peeper observed the key movements and the key touch. This setting again seems most favorable for the peeper and the experiment is expected to illuminate the worst-case scenario for peeping resilience. Aside from this setting, the evaluation follows the same process as for the basic method described in Sect. 4.2.

6.2 Results

We present the results on the robustness to peeping, ease of input, and input time.

6.2.1 Robustness to Peeping

Table 9 lists the average number of successfully peeped characters and Table 10 lists the average rate of peeping all four characters for each type of keypad. With Secure PIN Pad for numeric keys, 2.33 and 2.17 characters were peeped on average in three trials on SP1 and SP4, respectively. With Secure QWERTY Pad, 0.33 and 0.50 characters were peeped on average in three trials on SP8 and SP11, respectively.

With Secure PIN Pad for numeric keys, 0.13 PINs were peeped on average in three trials on SP1 while no PIN was

Table 9 Number of characters peeped (difference from Table 4).

Type	1st trial	2nd trial	3rd trial
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	1.00 (+0.10)	1.83 (+0.33)	2.33 (+0.83)
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	1.00 (+0.20)	1.50 (+0.10)	2.17 (+0.37)
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	0.00 (0.00)	0.17 (+0.07)	0.33 (+0.23)
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	0.00 (-0.10)	0.50 (+0.30)	0.50 (+0.20)

Table 10 Rate of all four characters peeped (difference from Table 4).

Type	1st trial	2nd trial	3rd trial
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	0.00 (0.00)	0.00 (0.00)	0.13 (-0.07)
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	0.00 (0.00)	0.00 (0.00)	0.00 (-0.10)
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

peeped on SP4. With Secure QWERTY Pad, no password was peeped in three trials on both SP8 and SP11.

In total, the number of characters peeped increased slightly until the second trial, and then largely until the third trial, but the total rate of all characters peeped remained almost the same.

6.2.2 Ease of Input

Table 11 shows the average number of characters successfully input on each type of keypad. There was no failure of input among participants in their 20s on all types of Secure Pad, which is better than those without the move b/f function. In contrast, there was increased failure of input among participants in their 50s (fewer successfully input characters among four). There was no retry since the move b/f was provided.

6.2.3 Input Time

Table 12 shows the average time per character from the user starting to move b/f until key input. Here, we only measure "Time/char", since most pairs spent various lengths of time to prepare for the simulated situation of losing the traces of key movements.

With Secure PIN Pad for numeric keys, it took 15.59 and 11.01 seconds on average on SP1 and SP4, respectively. With Secure QWERTY Pad, users spent 20.58 and 13.41 seconds on average on SP8 and SP11, respectively. On all types of Secure Pad, it took about 7 to 27 seconds to input a character with the move b/f function. This is far longer than retry, which took about 2 seconds (as shown in Table 7).

6.2.4 Feedback from the Participants

We received the following opinions from the participants after the experiment:

Table 11 Average number of successfully input characters (difference from Table 5).

Type	1st trial	2nd trial	3rd trial
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	3.94 (+0.14)	4.00 (+0.22)	3.83 (0.00)
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	4.00 (0.00)	4.00 (0.00)	4.00 (0.00)
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	3.56 (-0.24)	4.00 (+0.06)	2.67 (-0.91)
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	3.89 (-0.01)	4.00 (+0.06)	3.67 (-0.16)

Table 12 Average input time (time/char.) (sec).

Type \ Group of participants	All participants	Age: 20s	Age: 50s
SP1 (Ten, Kc:Sin, Ks:Sin, M:S)	15.59	9.96	23.01
SP4 (Ten, Kc:Mul, Ks:Mul, M:S)	11.01	6.84	16.58
SP8 (Qw, Kc:Sin, Ks:Sin, M:S)	20.58	15.03	27.15
SP11 (Qw, Kc:Mul, Ks:Mul, M:S)	13.41	8.18	20.47

- The move b/f function allows the user to control the speed of movement.
- This function is useful to replay the movements, especially when keys cross over.
- This is cognitively easier, since the retry requires the user to remember the shape and/or color of the target key again.
- Peeping becomes easier since peepers can observe key movements twice or more.
- Peeping becomes difficult since the speed of movement is changed by the user.

6.3 Considerations

Due to the move b/f function, the retry function is no longer necessary. However, it takes a far larger time than the retry, as discussed in 6.2.1. It is quicker to retry than to invoke the move b/f function when the user loses the key movements. Moreover, the ease of input was degraded for older people, as discussed in 6.2.2. This may be due to the overlaid functions of retry and move b/f along with some usability issues for older people (the slider might be too small for them, etc.).

At the same time, there were some positive opinions about controlling the speed of movement, confirming crossover movements, and less mental load for retrying. Although it takes time, it is only necessary when the user has lost the target key.

As for the most important issue, peeping resilience, the number of characters peeped increased slightly until the second trial and largely until the third trial, but the overall rate of all characters being peeped did not increase. It seems safest if we restrict the use of the move b/f function to twice.

At present, it is difficult to say which function is better than the other. Therefore, we consider providing them

both as alternative options. A more detailed user experience study for a certain period of time will be a future work.

7. Conclusion

On the basis of a survey on existing secure PIN/password input methods with respect to resilience to various levels of attacks, user's mental load, restrictions to hardware, requirements for the server side, and so on, we proposed a series of replay-attack and peeping resilient PIN/password methods that we call Secure Pad. The key idea is to associate colors and shapes with keys, erase key-top labels, move them smoothly and simultaneously or instantaneously, and let the user touch the target key. The user only needs to trace a single key, but peepers have to trace the movements of all the keys at the same time.

We conducted an experiment to evaluate the resilience, ease of input, and input time and found that Secure Pad is robust to peeping even over three trials. Although the success rate of inputting a password character dropped in the case of single color and shape, especially for older people, we found that when color and shape variations were added under the same condition, the input success rate improved and the number of retries decreased. As for the input time, it took several times longer with Secure Pad compared with the benchmarks featuring no key movement. This is the price of enhancing security, as with other methods. In Secure Pad, however, users can touch keys without moving them, which shortens the input time when there is no need to worry about security.

We compared the smooth and the instantaneous movements with the result that the instantaneous movements have higher peeping resilience but a worse success rate of input. An appropriate method can be chosen based on the required peeping resilience and the ease of use.

We also introduced a move backward/forward function for the user to replay the movements instead of relying on the retry function. Although the former takes more time and degrades the resilience to the peeping of each password character, it eases the process of key input and does not degrade the resilience to the peeping of the whole password. Therefore, both functions (move backward/forward and retry) can be provided as alternative options.

As a whole, the proposed series of PIN/password input methods achieves high resilience to shoulder hacking while providing satisfactory usability without large input errors.

There remain some issues for future research. Although we considered trials up to three to peep the PIN/password, it would be useful to extend this limit to figure out how many trials are necessary for successful peeping. We also need to make a user experience study on the move backward/forward function to compare it with the retry function. There are also a few issues pointed out by the users, including speed and crossover of movements and arrangement of different colors and shapes among keys, which need to be addressed. Moreover, movements along curvilinear or polygonal lines should also be considered.

Acknowledgements

This research is being partially supported by JSPS KAKENHI Grant Number (S) 18H05221. We would like to thank all of the people who joined the evaluation experiments.

References

[1] T.G. Willeby, "Secure key entry using a graphical user interface," U.S. Patent Application No. US 20020188872 A1.
[2] S. Tanaka and S. Takahashi, "暗証番号入力装置及び暗唱番号入力方法" (in Japanese), Japanese Patent Application No. 2002-134808.
[3] K. Makida, "パスワード入力装置及びパスワード入力方法" (in Japanese), Japanese Patent Application No. 2005-340699.
[4] Y. Kakinuma and K. Maruyama, "Color distance based authentication smartphone lock screens" (in Japanese), Proc.76th National Convention of IPSJ, vol.1, pp.121–122, March 2014.
[5] N. Hidaka, S Naguchi, and M. Okamoto, "Input Password Only with Arrow Keys," Proc. 9th Symposium On Usable Privacy and Security, July 2013.
[6] S. Sakurai and W. Takahashi, "Authentication methods for mobile phones" (in Japanese), IPSJ SIG Technical Reports, no.122 (CSEC-19), pp.49–54, Dec. 2002.
[7] A. KyuChoul and A.Y. Ha, "Password security input system using shift value of password key and password security input method thereof," US 20130047237 A1.
[8] T. Takada, "フェイクポインタによる暗証番号入力装置及び暗唱番号入力方法" (in Japanese), Japanese Patent Application No. 2007-175073.
[9] T. Takada, "fakePointer: A user authentication scheme that makes peeping attack with a video camera hard," IPSJ Journal, vol.49, no.9, pp.3051–3061, Sept. 2008.
[10] Y. Kita, F. Sugai, M. Park, and N. Okazaki, "Proposal and its Evaluation of a Shoulder-Surfing Attack Resistant Authentication Method: Secret Tap with Double Shift," Int. J. Cyber-Security and Digital Forensics, vol.2, no.1, pp.48–55, March 2013.
[11] K. Watanabe, F. Higuchi, M. Inami, and T. Igarashi, "CursorCamouflage: Multiple dummy cursors as a defense against shoulder surfing," SIGGRAPH ASIA 2012 Emerging Technologies, Nov. 2012. DOI:10.1145/2407707.2407713
[12] A.D. Luca, E.v. Zezschwitz, L. Pichler, and H. Husmann, "Using fake cursors to secure on-screen password entry," Proc. CHI 2013, Paris, France, pp.2390–2402, April 2013. DOI:10.1145/2470654.2481331
[13] T. Matsumoto and H. Imai, "Human identification through insecure channel," Eurocrypt '91, D.W. Davies, Ed., vol.547 of Lect. Notes Comput. Sci., Springer Verlag, pp.409–421, 1991. DOI:10.1007/3-540-46416-6_35
[14] X.-Y. Li and S.-H. Teng, "Practical human-machine identification over insecure channels," J. Combinatorial Optimization vol.3, no.4, pp.347–361, Dec. 1999. DOI:10.1023/A:1009894418895
[15] N.J. Hopper and M. Blum, "Secure human identification protocols," Proc. ASIACRYPT '01, C. Boyd, Ed., vol.2249 of Lect. Notes Comput. Sci., Springer Verlag, pp.52–66, Dec. 2001.
[16] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," Commun. ACM vol.43, no.2, pp.90–98, Feb. 2000. DOI:10.1145/328236.328110
[17] H. Sakano, "A state of art of biometric person authentication technology" (in Japanese), Japanese Journal of Forensic Science and Technology, vol.12, no.1, pp.1–12, June 2007. DOI:10.3408/jafst.12.1
[18] V. Roth, K. Richard, and R. Freidinger, "A pin-entry method resilient against shoulder surfing," Proc. 11th ACM Conference on Computer and Communication Security, Washington DC, USA, pp.236–245,

Oct. 2004.
DOI:10.1145/1030083.1030116
[19] D.S. Tan, P. Keyani, and M. Czerwinski, "Spy-resistant keyboard: More secure password entry on public touch screen displays," Proc. OZCHI 2005, Canberra, Australia, pp.1–10, Nov. 2005.
[20] K. Kobayashi, T. Oguni, and M. Nakagawa, "A PIN code/password input method resilient to shoulder hacking using difficulty of tracing multiple button movements" (in Japanese), Proc. Computer Security Symposium 2017, pp.728–733, Oct. 2017.
[21] K. Kobayashi, T. Oguni, and M. Nakagawa, "Usability improvement of an anti-shoulder-hacking PIN code/password input method exploiting tracing difficulty of multiple button movements" (in Japanese), Proc. IPSJ Interaction 2018, pp.565–568, March 2018.
[22] J. Intriligator and P. Cavanagh, "The spatial resolution of visual attention," Cognitive Psychology, vol.43, no.3, pp.171–216, Nov. 2001.
DOI:10.1006/cogp.2001.0755
[23] Z.W. Pylyshyn and R.W. Storm, "Tracking multiple independent targets: Evidence for a parallel tracking mechanism," Spatial Vision, vol.3, pp.179–197, 1998.
DOI:10.1163/156856888X00122

Kokoro Kobayashi was born on 28 April, 1992. He received B.Eng. and M.Eng. degrees in Computer and Information Sciences from the Tokyo Univ, of Agri. and Tech. (TUAT), Japan in 2018. He is currently pursuing a Ph.D. degree at TUAT. His research interests include privacy, security and user interfaces for personal devices.

Tsuyoshi Oguni received B.Eng and M.Eng. degrees from the Tokyo Univ, of Agri. and Tech. (TUAT) in 1996 and 1998, respectively. Since 1998, he has been working at NTT DATA. When, he was a student, he worked on pen/touch-based user interfaces, especially user interfaces for large electronic whiteboards and educational applications. He is currently working on Banking systems.

Masaki Nakagawa is a Professor of Media Interaction in Dept. of Computer and Information Sciences, TUAT. He has been working on pen/touch-based user interfaces, handwriting recognition, and related topics. He received the Minister of Education and Science award of Japan and the contribution award from Tokyo Metropolitan Government in 2016. He is a fellow of IAPR (International Association of Pattern Recognition), IEICE (Institute of Electronics, Information and Communication Engineers, Japan) and IPSJ (Information Processing Society of Japan).