Shoulder-Surfing Resistant Graphical Authentication based on One-Time Shape-Pattern Generation

Kentaro Ishii*

School of Network and Information, Senshu University

Abstract

We propose a graphical authentication method based on one-time shape-pattern generation aiming at mitigating the shoulder surfing problem. In the proposed method, one-time correct and dummy graphics are generated based on predefined generation rules. The user of the proposed method is authenticated by choosing the correct graphic from the presented group of graphics. Since different graphics are displayed each time, it can be difficult for another person to acquire necessary information for authentication even if conducting shoulder surfing. We present the proposed method using shape-pattern generation with in-depth consideration and evaluation. The result of the evaluation experiment, where one participant did authentication, while the other participant did shoulder surfing, showed that the proposed method achieved high genuine user pass rates and high non-genuine user rejection rates.

1 Introduction

In password/passcode-based authentication or draw-a-secret authentication used in the usual context, a non-genuine user can infer password/passcode or secret pattern, which is key information for authentication, by shoulder surfing of the authentication scene. In particular, in the use of touch input devices like smartphones, observers can easily know the input from touch location [1].

This paper aims to mitigate this problem and proposes a shoulder-surfing resistant authentication method based on one-time shape-pattern generation. The proposed method generates a one-time correct graphic and dummy graphics on each time of authentication and presents a set of the graphics on the screen based on a generation rule decided in advance. A user will be authenticated if the user selects the correct graphic predefined times (Figure 1). The genuine user of the system, which knows the generation rule, can select the correct graphic. On the other hand, the individual graphics changes each time, so it is expected to be difficult for non-genuine users to find the correct graphic even if they perform shoulder surfing.

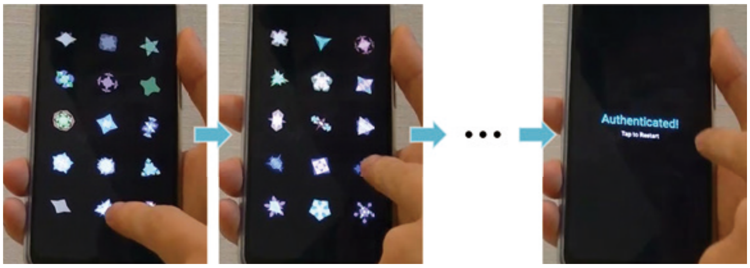

Figure 1: Proposed graphical authentication. The user is authenticated if the user selects the correct graphics predefined times.

We propose a method that employs one-time shape-pattern generation rule for defense to shoulder surfing. This paper describes the proposed method and its evaluation experiment, after summarizing related work of image-based authentication. Our contribution is proposing a novel approach of generating one-time images for shoulder surfing defense of image-based authentication.

[*] kenta@pc.fm.senshu-u.ac.jp

We hypothesize that the proposed method is easy enough to use for genuine users and has resistant ability for non-genuine users and conduct evaluation experiments. In addition, we discuss usability of the proposed method in terms of answer time by comparing with prior shoulder surfing resistant methods.

2 Related Work

Human image recognition ability is considered to be higher than character recall ability like ordinary password authentication [2], and image-based authentication methods are considered to have a lower memory load than ordinary password authentication methods. Image-based authentication can be classified into three types: “cognometric” for selecting the image itself, “locimetric” for selecting a specific location in the image, and “drawmetric” for performing a drawing operation of a registered figure.

Déjà Vu [2] and PassFaces [3] were proposed as a cognometric image-based authentication. In Déjà Vu, five correct images are determined in advance from computer-generated random art figures, and authentication is performed by selecting five correct images from 25 presented images. In PassFace, five correct images are randomly given from a face image database, and authentication is performed by selecting one correct image from nine presented images five times. However, the effect of memory load reduction for these methods may be limited because they use irrelevant images to the user. Awase-E is another cognometric authentication that allows users to register correct and dummy images [4]. The user of this system can use recognizable images based on individual episodic memory. These above methods, however, suffer by shoulder surfing, because the exact correct image is presented on the authentication screen. Others can easily obtain the key information for authentication by shoulder surfing. We deal with a technique to prevent the theft by shoulder surfing in cognometric image-based authentication.

As shoulder surfing resident methods, combination of cognometrics and locimetrics were proposed [5, 6, 7]. These methods use cognometrics to search for multiple images displayed on the screen, and locimetrics to click coordinates in a convex hull formed by connecting the searched images. It is hard to guess the searched images from the coordinates where the user clicks, and this is basically a mechanism of shoulder surfing defense. Wiedfenbeck et al. reported that the average time required for one authentication trial was 71.66 seconds [5]. The remaining two papers did not state the time required for authentication but seem to be nearly the same. This study differs from these studies in that our method uses cognometrics only to reduce user load and answer time. We discuss answer time based on the results of the experiment.

In addition, locimetric image-based authentication by eye gaze measurement [8] and drawmetric image-based authentication by adding dummy strokes or erasing drawn strokes [9] has also been proposed. As locimetric and drawmetric method, the principle of image-based authentication is different from this study. In addition, our proposed method differs from the former proposal that requires additional hardware and the latter proposal that does not focus on shoulder surfing defense.

Tessellation is used for some image-based authentication methods to mitigate the shoulder surfing problem. The system presents a tessellated image automatically generated from the correct image [10]. For non-genuine users that do not know the original correct image, it is difficult to identify the original correct image from the presented tessellated images. In addition, this tessellation technique is also applied to locimetric image-based authentication, in which the user specifies the certain location in the tessellated image [11]. This study differs from the former proposal in that our method presents different images each time, while the system in [10] presents the same tessellated image for the correct answer. In addition, this study differs from the latter proposal in that our method uses cognometrics that means a recognition-based method, and it is expected that the response generation at the time of authentication is easier than the locimetric method.

3 Proposed Graphical Authentication

3.1 Basic Graphic Generation Algorithm

The proposed method uses a basic graphic generation algorithm described in this section. Both correct and dummy graphics are generated by the basic generation algorithm. Figure 2 shows examples of graphics generated by the basic graphic generation algorithm. We arrange a graphic generation method proposed by Miyashita et al. [12] to build the algorithm.

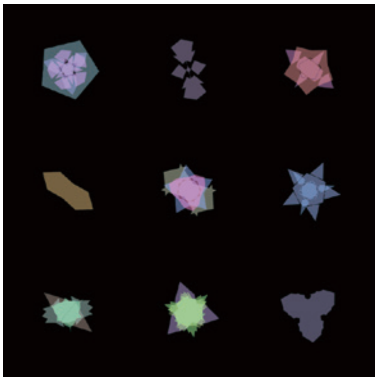

Figure 2: Graphics generated by the basic algorithm.

The following is the procedure of the basic graphic generation algorithm. Each operation has parameters that are randomly determined, which results in different shape patterns each time.

1. Prepare a regular polygon with a random number of vertices.
2. Create a new vertex at the middle point of each edge of the polygon, and move the new vertex by a random positive or negative distance toward the direction from the center of the polygon to the new vertex (Figure 3; The moving distance of each vertex is the same).
3. Repeat step 2 a random number of times.
4. Fill the polygon with a random color.
5. Rotate the polygon by a random angle on its center.
6. The random number of polygons generated by step 1 to 5 are superposed with constant transparency.

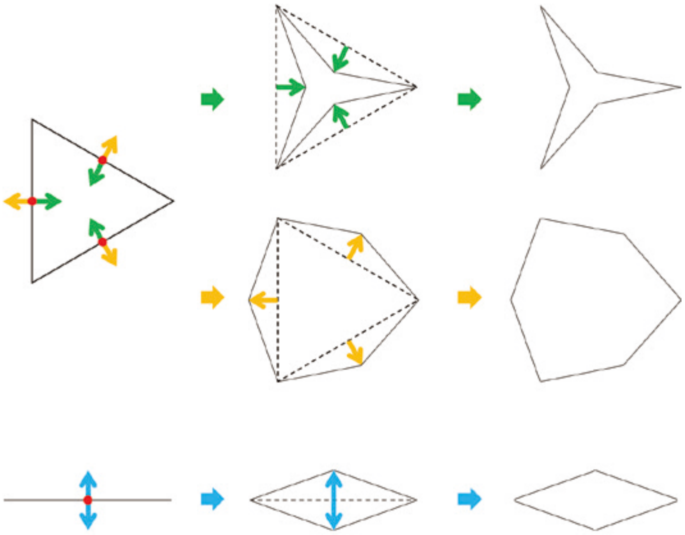

Figure 3: Edge middle point deformation.

Step 1 start with a regular polygon as an initial shape, and we decide that the radius of the regular polygon or the distance from the center to each vertex depends only on the number of superpose described in step 6. The number of vertices is a random parameter. In the current implementation, the number of vertices is an integer from 2 to 5. Although a digon, which means the polygon has two vertices, is not generally defined, we define a digon has two edges between two vertices (referred as A and B): from A to B and from B to A. This definition works in the subsequent operations as if the edges are initially at the same position but move to different directions. The radius or the distance from the center to each vertex was set to 150 pixels for the first superpose, 125 pixels for the second superpose, and 100 pixels for the third superpose.

Step 2 is the most important operation in our graphic generation. The distance of new vertex movement is a random parameter, which can be either positive or negative. If the parameter is positive, new vertices move farther from the center; if it is negative new vertices move nearer to the center. Intuitively, if the parameter is positive, the polygon shape expands; if it is negative, the shape shrinks (Figure 3). In case the initial polygon is a digon, a rhombus is generated by the first operation because the midpoints of the two same positions move in opposite directions by the definition in step 1. (Figure 3 bottom). In the current implementation, the distance of movement is a random real number from minus radius of the initial polygon to plus radius, but we set an additional constraint of new vertices not moving to farther position than the initial vertices. This operation is repeated as step 3. Each time the operation is performed, the number of vertices doubles, and the polygon becomes more complicated. In the current implementation, the number of repetitions is an integer from 2 to 4.

Step 4 colorizes the polygon with a single color of three random parameters of 256-scale RGB color space. In the current implementation, each component of RGB is an integer from 128 to 255.

Step 5 changes the angle of deviation of the polygon. Since a regular polygon generated in step 1 always has a vertex at the position of angle 0, which is directly to the right based on the definition of screen coordinate axes, the polygons generated by step 1 to 4 look oriented in the same direction. Therefore, this rotating operation aims to make different orientation, and the angle of rotation is a random parameter. In the current implementation, the angle of rotation is a real number between 0 and 2π, that is, the orientation of the polygon appears without any constraints.

Step 6 compose a final graphic by superposing one or more colored polygons generated by step 1 to 5 with the fixed transparency, which results in various patterns of graphics. The number of superpose is a random parameter. In the current implementation, the transparency is 127 in 256-scale, and the number of superpose is an integer from 1 to 3.

3.2 Authentication Graphic Generation Rules

Based on the basic graphic generation algorithm, we define an authentication graphic generation rule that generates a combination of a correct graphic and dummy graphics. The correct graphic is the graphic that the user should select for authentication, and the dummy graphics are the graphics that the user should not select. There are two features that an authentication graphic generation rule should have: those who know the rule can distinguish the correct graphic from the dummy graphics, and those who do not know the rule cannot guess the rule from a set of generated graphics. The latter feature is necessary to prevent that a non-genuine user is authenticated, even after performing shoulder surfing.

We define authentication graphic generation rules by constraining random parameters of the basic graphic generation algorithm described in section 3.1. For instance, the number of vertices of the initial polygon, which is random in the basic algorithm, is fixed to 3 for correct graphics. On the other hand, for dummy graphics, the number of vertices of the initial polygon is a random number other than 3. According to this method, in principle, we can define authentication graphic generation rules for each random parameter of the basic algorithm. There are six random parameters in the basic algorithm: the number of vertices of the initial polygon, the moving distance when adding vertices, the number of repetitions of adding vertices, the color of the polygon, the rotation angle, and the number of superposing. For each of the six random parameters, we examine if constraining the parameter can lead two desirable features of genuine users distinguishing the correct graphic and non-genuine users not guessing the rule.

3.2.1 Examination of Random Parameters

The number of vertices of the initial polygon greatly affects the shape of a generated graphic. Intuitively, most generated graphics have shapes that can be associated with initial polygons. For instance, if the number of vertices of the initial polygon is 3, most generated graphics have shapes that can be associated with triangles or trigons. With the basic graphic generation algorithm, the generated graphic is point symmetric to the center, and rotational movement of certain angle determined by the initial polygon will produce the same polygon. Therefore, we consider that the user can identify the initial polygon and conclude that the number of vertices of the initial polygon is a suitable parameter for defining an authentication graphic generation rule.

The moving distance when adding vertices greatly affects the shape of a generated graphic. For example, if moving distance of the first deformation is a large positive value, the generated graphic has an inflated shape, and if moving distance of the first deformation is a small negative value, the generated graphic has a shrunk shape. We can define a generation rule as the correct graphic should take a certain fixed range of moving distance, and the dummy graphics should take other moving distances. However, when the constraint of the correct graphic is a large positive value for instance, only one graphic has an inflated shape while the others have a shrunk shape, which is a big difference even for those who did not know the rule. The same is true for the constraint with small negative value and constraint with values close to 0. We conclude that the moving distance when adding vertices is not a suitable parameter for defining an authentication graphic generation rule.

The number of repetitions of adding vertices does not significantly affect the shape of a generated graphic. This is because the more vertices are added, the more in detail changes appear, and the current implementation guarantees at least two repetitions. Another reason is that the moving distance when adding vertices is random, so the shape hardly changes when the distance is close to 0. We conclude that the number of repetitions of adding vertices is not a suitable parameter for defining an authentication graphic generation rule.

The color of the polygon directly affects the appearance of a generated graphic. The user seems to quickly identify graphics with the specific color. Considering the possibility of quick input and less information given to others, we conclude that the color of the polygon is a suitable parameter for defining an authentication graphic generation rule.

The rotation angle itself does not greatly affect the appearance of a generated graphic. However, the graphic looks a little prominent if it is facing the right direction. This means in two ways; the orientations of multiple polygons are all the same, or the orientations of all polygons are straight up or down. The former rule requires twice or more superposed graphic and depends also on the number of superposing described later, so we exclude the former rule. The latter rule is valid for both single and multiple polygon graphics and defined by the rotation angle only. We conclude the latter rule of the rotation angle is a suitable parameter for defining an authentication graphic generation rule.

The number of superposing sufficiently affects the appearance of a generated graphic. Since each polygon to be superposed has different initial radius, shape, color, and rotation angle determined by the basic graphic generation algorithm, the combination of different characteristic makes a significant effect. Intuitively, the more polygons are superposed, the more complicated graphic is generated. We concluded that the number of superposing is a suitable parameter for defining an authentication graphic generation rule.

3.2.2 Definition and Interpretation

As we consider that four of the random parameters of the basic graphic generation algorithm are suitable parameters, we can define four categories of generation rules: the number of vertices of the initial polygon, the color of the polygon, the rotation angle, and the number of superposing. According to candidates of each parameter of the current implementation, we define a total of 12 authentication graphic generation rules. Figure 4 shows sets of graphics generated by the authentication graphic generation rule, each set representing each of the four categories.

Category 1 Correct graphics: The number of vertices of the initial polygon is n. Dummy graphics: The number of vertices of the initial polygon is other than n. There are four rules in this category: n is one of {2, 3, 4, 5}.

Category 2 Correct graphics: c is the largest among the component in the RGB color space. Dummy graphics: A component other than c is the largest among the component in the RGB color space. There are three rules in this category: c is one of {R, G, B}.

Category 3 Correct graphics: The rotation angle of each polygon is θ. Dummy graphics: The rotation angle of each polygon is other than θ. There are two rules in this category: θ is either of $\{\pi/2, 3\pi/2\}$. (According to the definition of screen coordinate axes, $\theta = \pi/2$ is right downward, and $\theta = 3\pi/2$ is right upward.)

Category 4 Correct graphics: The number of superposing is n. Dummy graphics: The number of superposing is other than n. There are three rules in this category: n is one of {1, 2, 3}.

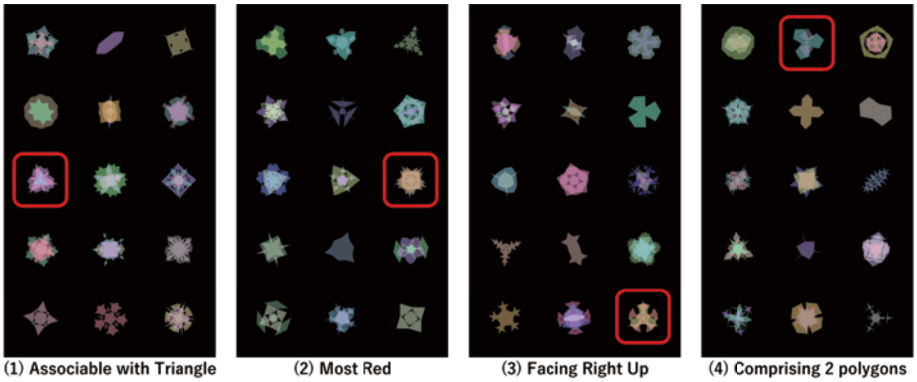

Figure 4: Graphics generated by the generation rules and interpretations of the generation rules. Each set of shape patterns includes one correct graphic and 14 dummy graphics. Users do not necessarily need to know the basic generation algorithm. There is no big differences among graphics in each set nor graphic sets of the categories.

Intuitively, the correct graphic generated by the above rules can be interpreted as follows. Therefore, users do not need to know the internal structure of the program nor parameters used in the rules, which we think this is important. In addition, there is no significant difference among graphics in each set nor graphic sets of categories (Figure 4).

Category 1 The correct graphic can be associated with {digon, trigon, tetragon, pentagon}.

Category 2 The correct graphic is the most {red, green, blue}.

Category 3 The correct graphic is facing right {downward, upward}.

Category 4 The correct graphic is comprising {1, 2, 3} polygons.

3.3 Application Software

We implement application software for PC and Android smartphone by incorporating authentication graphic generation rules described in section 3.2 and authentication user interface (figure 5). In this application, an authentication screen contains one correct graphic and 14 dummy graphics generated based on one of authentication graphics generation rules. When the user taps one of the presented graphics, other 15 graphics will appear on the screen. The user needs to select the correct graphic four times in a row to be authenticated. Although 15 graphics (14 dummies) and four times are parameters determined by user preferences, we use these settings for the following evaluation.

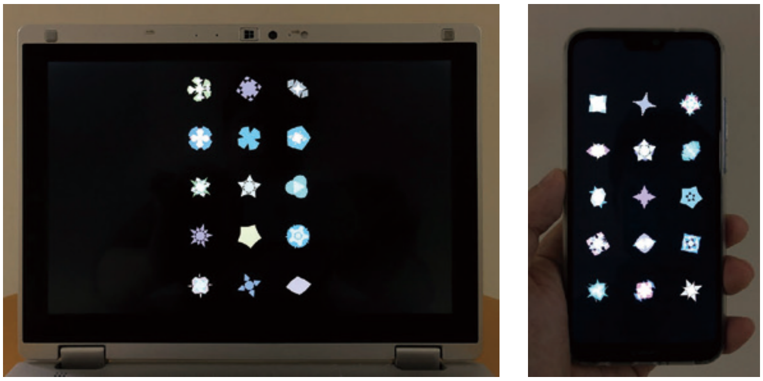

Figure 5: Authentication application running on a PC and smartphone.

4 Evaluation Experiment

We conduct two kinds of evaluation: (1) investigation on rules and prior knowledge and (2) comparison with existing methods. In both experiments, we invite two participants for one round of the experiment. One participant plays a role of a genuine user, and the other participant plays a role of a non-genuine user performing shoulder surfing. We use an Android application version of the proposed method for both experiments.

For the first experiment, the experimenter assigns a generation rule to each session and provide only intuitive interpretation to the participant as a first-time genuine user. This design aims at verifying easiness to understand the idea of the proposed method. We also control prior knowledge of the participant of a non-genuine user, where the non-genuine user performs shoulder surfing in three conditions: "system inexperienced", "rule category inexperienced", and "rule category experienced".

On the other hand, for the second experiment, the participants of genuine users can determine the applied generation rule by themselves like a usual context. Since the experimenter provides complete explanations of the proposed method and existing methods for the second experiment, the participants of non-genuine users has a maximum level of prior knowledge at the time of experiment sessions.

Participants are invited by solicitation at a cafeteria and lounge of a university. After the experimenter explains the purpose of the experiment, which is evaluating an authentication method, the participants can decide if they participate by themselves. The participants received 500 yen, roughly 4.5 dollars, for taking part in the experiment.

4.1 Investigation on Rules and Prior Knowledge

4.1.1 Procedure

One round of the experiment includes six sessions, where we use all different authentication graphic generation rules. Assuming two participants are A and B, we design the order of a player of a genuine user is A → A → B → B → A → B, which means the players of a genuine and a non-genuine users alternate during the experiment. We aim at acquiring data under different conditions regarding prior knowledge of our authentication system among the first two sessions, the middle two sessions, and the last two sessions.

In the first two sessions, the participant B as a non-genuine user performs shoulder surfing without experience of using the authentication system. The participant B is not told that the authentication system is operating based on an authentication graphic generation rule nor that a rule exists.

In the middle two sessions, the participant A as a non-genuine user already knows that the authentication system is operating based on an authentication graphic generation rule because the participant A was a genuine user and told it from the experimenter in the first two sessions. Shoulder surfing is performed under the condition of knowing the rule-based operation of the system. However, we dare to design that the two rules used in the middle two sessions are selected from different categories of the two rules used in the first two sessions, which means the non-genuine user has an experience of the system but does not have an experience of the selected rule category.

In the last two sessions, the participant A or B as a non-genuine user already knows that the authentication system is operating based on an authentication graphic generation rule, and the rules used in the last two sessions are selected from the same categories that the non-genuine user experienced as a genuine user in the previous sessions. Note that we use a different rule or parameter from the same category, for example trigon for this session and tetragon for a previous session. Shoulder surfing is performed under the condition of knowing a similar rule.

In summary, we compare three conditions of "no experience of the system & not told rule-based operation", "experience of the system & no experience of the rule category", and "experience of the system & experience of the rule category". We refer the three conditions to "system inexperienced" condition, "rule category inexperienced" condition, and "rule category experienced" condition, respectively. We also refer this factor to "prior knowledge".

The order of categories (out of 4) or parameters (out of 2-4) of rules used in the experiment is counterbalanced among participants. The following is the procedure of each session, so the following procedure is repeated six times for one round of experiment.

1. The experimenter explains an authentication graphic generation rule used in this session to the participant of a genuine user. Note that the explanation is performed by intuitive interpretation described in section 3.2.2, not by the algorithm and parameters.

2. The participant of the genuine user practices the authentication application six times.
3. The participant of the genuine user tries authentication three times. At the same time, the participant of a non-genuine user performs shoulder surfing near the genuine user.
4. After that, the participant of the non-genuine user tries authentication three times.
5. The participant of the non-genuine user fills in a simple questionnaire of how the participant thinks during shoulder surfing.

4.1.2 Data to Acquire

The data acquired through the experiment are correct/incorrect for each graphic selection (a tap), answer time for each graphic selection (a tap), authentication success/failure for each authentication trial (four taps), and answers to questionnaires (a session).

Correct/incorrect and answer time are acquired by automatic recording by the authentication application for each answer trial. In addition, success/failure is also acquired by automatic recording by the authentication application for each authentication trial. Both a genuine and a non-genuine users perform three authentication trials per session, and four answer trials are included per authentication trial. Therefore, we obtain 12 correct/incorrect taps and answer time per session, and three success/failure authentications per session.

Only the participant of the non-genuine user answers the questionnaire at the end of a session. The question sentence is “How did you think when you performed shoulder surfing?”, and the participant answers by free writing. Therefore, we obtain one free writing per session.

4.1.3 Results and Analysis

52 university students of 26 pairs participated in the experiment. However, two participants in different pairs asked to stop the experiment without completing all sessions, because they had color vision deficiency and found out that it was hard to answer authentication challenge of a rule of the color category. We used data from 48 participants of 24 pairs, as the above two pairs were excluded for evaluation analysis.

Each of 48 participants to be evaluated had three sessions of a genuine user and three sessions of a non-genuine user, so we acquired a total of 144 sessions for both genuine and non-genuine users. Since both a genuine and a non-genuine users performed three authentication trials per session, and four answer trials per authentication trial, we had a total of 432 authentication trials and 1728 answer trials for genuine and non-genuine users. As the rule categories were counterbalanced, we had 36 sessions, 108 authentication trials, and 432 answer trials for each of the four categories. Since the number of rules in a category differs depending on the rule category, the number of sessions also differs depending on the rule category: 9 sessions, 27 authentication trials, and 108 answer trials for category 1 (initial polygon); 12 sessions, 36 authentication trials, and 144 answer trials for category 2 (color) and category 4 (superposing); 18 sessions, 54 authentication trials, and 216 answer trials for category 3 (orientation). For each prior knowledge condition, we had 48 sessions, 144 authentication trials, and 576 answer trials.

Authentication and Correct Answer Rates Table 1 top shows the percentages of successful authentication attempts for each rule. In general, authentication rates for genuine and non-genuine users were largely different, although we allowed non-genuine users to freely perform shoulder surfing. Considering shoulder surfing for three consecutive authentication trials is more severe condition than practical use, we consider the proposed method effectively worked.

Looking into individual rules, double and triple of category 4 for genuine users had lower authentication rates than the others. This suggests that double and triple were somewhat difficult for genuine users. On the other hand, trigon of category 1 and single of category 4 for non-genuine users had higher authentication rates than the others. This suggests that trigon and single were somewhat easy for non-genuine users to steal.

To investigate the performance in more detail, we consider the percentages of correct answer attempts for each rule. As shown in Table 1 middle, correct answer rates were higher than authentication rates except for the case of 100.0%, because authentication rates in principle equal to results of logical AND of four correct answers. In this way, we found out that trigon and single still had high values compared with the others, the differences were not as large as authentication rates. The possibility of selecting the correct answer by chance is 1/15, so the chance level of a correct answer rate is 6.7%. We performed a series of binomial tests to examine if correct answer

Table 1: Authentication rates (%), correct answer rates (%), and average answer times (msec.) for each generation rule.

	Category 1				Category 2			Category 3		Category 4			Total
	digon	trigon	tetragon	pentagon	red	green	blue	down	up	single	double	triple	
Authentication rate (genuine)	100.0	88.9	100.0	100.0	94.4	94.4	94.4	98.1	94.4	94.4	72.2	83.3	92.8
Authentication rate (non-genuine)	25.9	37.0	22.2	11.1	5.6	16.7	5.6	11.1	7.4	33.3	22.2	22.2	17.1
Correct answer rate (genuine)	100.0	96.3	100.0	100.0	97.2	98.6	98.6	99.5	98.6	98.6	93.1	94.4	98.0
Correct answer rate (non-genuine)	44.4	50.0	30.6	38.9	21.5	34.7	27.1	21.3	21.8	48.6	41.7	30.6	32.6
Average answer time (genuine)	4109	5228	14995	6047	2939	4172	10879	6153	5079	3219	8431	9677	6535
Average answer time (non-genuine)	9729	8577	10150	3605	7349	8548	9904	7029	10082	5975	16647	9339	8956

rates for non-genuine users were significantly different from the chance level of 6.7%. As a result, there were significant differences in all rules ($p < 0.01$; $g = 0.146 \sim 0.433$). This result indicates that the participants of non-genuine users did not a random guess. Therefore, although there is a difference depending on the rules, it is possible that the participant of non-genuine users could narrow down the possibility of the applied rule even in failure authentication trials.

Table 2 top shows the percentages of successful authentication attempts for each prior knowledge condition. For genuine users, there are no big difference among prior knowledge conditions, as they were told intuitive interpretation of the applied rule in the same manner. For non-genuine users, rule category experienced condition had a higher authentication rate than system inexperienced condition and rule category inexperienced condition. We performed a chi-square test and found significant differences in authentication rates among the conditions ($p < 0.01$; $V = 0.162$). This indicates that shoulder surfer who have experienced similar rules can recognize the applied rules better with the proposed method. We obtained a similar result for correct answer rates for each prior knowledge condition as well (Table 2 middle). A chi-square test revealed that there were significant differences in correct answer rates among the conditions ($p < 0.01$; $V = 0.173$).

Table 2: Authentication rates (%), correct answer rates (%), and average answer times (msec.) for each prior knowledge condition.

	System inexperienced	Rule category inexperienced	Rule category experienced	Total
Authentication rate (genuine)	92.4	93.8	92.4	92.8
Authentication rate (non-genuine)	11.8	13.9	25.7	17.1
Correct answer rate (genuine)	97.6	98.4	98.1	98.0
Correct answer rate (non-genuine)	28.8	25.2	43.9	32.6
Average answer time (genuine)	6196	5629	7779	6535
Average answer time (non-genuine)	8206	8800	9862	8956

Answer Time Table 1 bottom shows average answer times for the users to select a graphic for each rule. In general, genuine users responded in shorter time than non-genuine users. We think this is reasonable, because the rule that genuine users knew help to find the correct graphic. However, differences were not so large.

Looking into individual rules, digon, red, green, and single for genuine users had relatively shorter answer time, but we also found large variation among participants in original data. Specifically, when we defined the rules of category 2 (color), we expected for users to quickly find the correct graphic, but the result of blue did not show such an indication. On the other hand, when comparing within a category, digon and single seemed to easier for genuine users to identify the correct graphic.

Table 2 bottom shows average answer times for the users to select a graphic for each prior knowledge condition. For both genuine and non-genuine users, answer time did not change much among the prior knowledge conditions. It seems that authentication graphic generation rule affects answer time more than prior knowledge.

Questionnaire Investigating the obtained answers, we first found out that the participants of non-genuine users in system inexperienced condition often predicted that some rules exist. As described in section 4.1, the participants who first played a role of non-genuine users were not told that a rule exists, but many of them mentioned rule based search, such as "I searched for a graphic that had most corners." (with concrete strategy) or "I found it difficult to find a pattern." (without concrete strategy). In addition, there were also descriptions that did not conclude existence of a rule but explored the possibility, such as "I tried to find regularity." Counting all types of the above, 17 out of 24 answers for the first session indicated that a rule might exist. Although we did not expect this, we guess that the participants thought it would be indistinguishable without some rules. This is an open issue that should be investigated in future experiments.

On the other hand, responses from the third session and after, where the non-genuine users were already told that a rule exists, still contained many statements considering concrete rules. Furthermore, rule estimation based on non-visual or meta information like how the genuine user answered appeared, such as "Since the genuine user answered quickly, I thought it was a simple graphic." or "I thought it was a rule I could answer quickly because the genuine user answered quickly."

In comparison with the data of authentication success/failure, non-genuine users guessed almost right about the rule in case the non-genuine users succeeded authentication multiple times. In case the non-genuine users succeeded authentication only once, about a half questionnaire guessed right about the rule, but the other half questionnaire did not understand the rule. This supports that some participants could narrow down the possibility of the applied rule even if they did not completely understand the rule.

4.2 Comparison with Existing Methods

4.2.1 Procedure

We compare the proposed method with four existing methods: passcode authentication, draw-a-secret authentication, random art figure method [2], and face image method [3]. The first two methods are representatives from practical use, and the last two methods are cognometric graphical authentication using fixed image figures.

First, the experimenter explains how the proposed method and all four existing methods works to both two participants and demonstrates actual authentication using a smartphone as the participants request. For the proposed method, the participants are told each of 12 authentication figure group generation rules and that they can set a preferred rule from the 12 rules. For passcode and draw-a-secret authentication, we use built-in implementations on the smartphone. For random art and face image authentication, we implement applications on the smartphone with similar conditions to the proposed method (15 images per screen and four taps per authentication; Figure 6). Random art images are generated in the same way as existing research [2], and face images are generated by StyleGAN [13, 14], which is a version of Generative Adversarial Networks (GAN) [15] and can generate non-existent faces and avoid that a known face appears in the experiment.

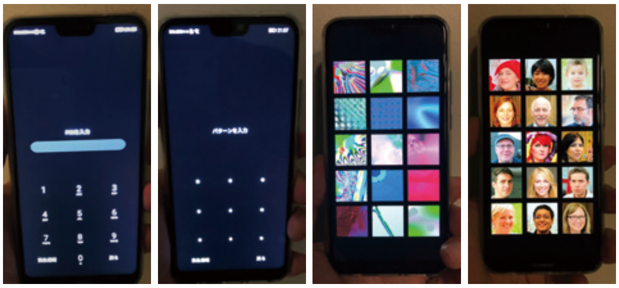

Figure 6: Existing authentication methods: from the left, passcode-based, draw-a-secret, random art figure, and face image.

Like the first experiment, one participant playing a role of a genuine user do authentication while the other participant playing a role of a non-genuine user perform shoulder surfing. But, in this experiment, passcode, secret pattern, fixed pass-images, or authentication figure group generation rule for each method is set by the participant of the genuine user, and we allow the genuine user to try and change the settings as the participant want before each experimental session. We also allow the non-genuine user to try the authentication method to be tested using a backup smartphone.

Five sessions are continuously conducted with the same role of the genuine and non-genuine users for all five methods. After that, we exchange the roles of two participants and proceed five more sessions. The order of methods used in the experiment is partially counterbalanced within the two practical existing methods and three cognometric graphical methods. The following is the procedure of each session, so the following procedure is repeated 10 times for a round of experiment.

1. The experimenter specifies an authentication method and instructs both participants to try the specified authentication method. The experimenter also instructs the participant of a genuine user to determine a passcode, secret pattern, fixed pass-images, or authentication figure group generation rule for authentication.

2. The participant of the genuine user practices the authentication application as many times as desired.

3. The participant of the genuine user tries authentication three times. At the same time, the participant of a non-genuine user performs shoulder surfing near the genuine user.

4. After that, the participant of the non-genuine user tries authentication three times.

4.2.2 Data to Acquire

The data acquired through the experiment are authentication success/failure and authentication time for each authentication trial (four taps except for draw-a-secret) for all methods. For the three graphical authentication methods, we also acquire correct/incorrect for each graphic selection (a tap). For the two practical authentication methods using passcode and draw-a-secret, although we can divide a user's action into multiple taps of digits or line segments drawn, we consider that the combination makes sense, and each tap or line segment does not. So, we do not measure correct/incorrect for each tap or line segment for the two practical methods.

For the three graphical authentication methods, data are acquired by automatic recording by the authentication application as well as the first experiment. For the two practical authentication methods, since we do not have a way of automatic recording, we perform video recording during the sessions and manually measure authentication success/failure and authentication time using the video. Both a genuine and a non-genuine users perform three authentication trials per session, and we obtain three authentication success/failure and authentication time per session. For the three graphical methods, we additionally obtain 12 correct/incorrect taps per session.

4.2.3 Results and Analysis

12 university students of 6 pairs participated in the experiment. Each of 12 participants had sessions of a genuine user and sessions of a non-genuine user for all five methods, and both genuine and non-genuine users performed three authentication trials per session, so we acquired 12 sessions and 36 authentication trials for both genuine and non-genuine users for each authentication method.

Table 3 shows the authentication success rates, correct answer rates, and average authentication times for each authentication method. Regarding the authentication success rates and correct answer rates, participants of genuine users generally made successful authentication in all methods. Data from participants of non-genuine users showed differences among the methods. Authentication success rates of non-genuine users were low for the proposed method, followed by random art figure and face image authentication, followed by passcode and draw-a-secret authentication. Authentication success rate and correct answer rate of the proposed method roughly match the result of the rule category experienced condition in the first experiment.

On the other hand, authentication time was short for passcode and draw-a-secret authentication, followed by random art figure and face image authentication, followed by the proposed method. Regarding the graphical authentication methods, the participants of genuine users tended to respond in a shorter time than the participants of non-genuine users, which shows the same tendency as the first experiment. On the other hand, the average authentication time of the proposed method was twice as short as the result of the first experiment (26.1s vs. 12.8s). We think this is because all the rules were explained in advance and the participants selected the preferred rule by themselves. Although an expert user (an experimenter) can be authenticated in shorter time, which is about 5-6 seconds and similar to the fixed graphical methods, the proposed method requires more time for the first-time users like the participants.

Table 3: Authentication rates (%), correct answer rates (%), and average authentication times (msec.) for each authentication method.

	Proposed method	Random art figure	Face image	Passcode	Draw-a-secret
Authentication rate (genuine)	100.0	100.0	88.9	100.0	100.0
Authentication rate (non-genuine)	27.8	61.1	50.0	100.0	72.2
Correct answer rate (genuine)	100.0	100.0	97.2	-	-
Correct answer rate (non-genuine)	49.3	81.9	73.6	-	-
Average authentication time (genuine)	12783	5465	6336	2002	1626
Average authentication time (non-genuine)	15927	8337	10672	1533	2087

5 Discussion

The proposed method has an advantage in preventing a non-genuine user's authentication even if shoulder surfing is performed. For first-time users, the system is usable by a simple explanation. Furthermore, the average time required for first-time users for an authentication trial is 26.1 seconds if using a randomly selected rule as in the first experiment and 12.8 seconds if using a preferred rule as in the second experiment. Compared to the average authentication time of 71.66 seconds by Wiedenbeck et al. [5], which utilizes the combination of cognometric and locimetric method, our recognition-based method showed better performance in authentication time. On the other hand, 26.1 or 12.8 seconds is a bit long for daily use. Although some of trial was made in less than 10 seconds or even less than 5 seconds, this is still a limitation of our authentication method with shoulder surfing defense. Considering this fact, we consider that possible application scene of the proposed method is in a public place of many people to switch from the normal authentication method, such as the situation that a person needs a bank transfer on a train or a bus with a high occupancy. As another scenario, we consider the use for a public terminal that generally has a big touch screen for input without a normal mouse and keyboard.

Also as shown in the first experiment, participants who have experienced the rules of the same category can find the applied rule more easily. The experimental settings seem to be harder condition than practical use for the genuine user in that shoulder surfing is performed three times in a row. But, even if the non-genuine user has a maximum level of prior knowledge of the proposed method as shown in the second experiment, the proposed method blocked three quarters of the attack trials. Ideally, the method has more rejection performance even if the non-genuine user has a maximum level of prior knowledge, and we will discuss a possible solution to this issue in the following section.

6 Possible Variation for Further Safety

In this section, we present a variation of the proposed method to overcome a limitation of the original method. The key idea is inline rule notification, and we will show its effectiveness for further safety by a user study.

The authentication graphic generation rule used in the original method is always fixed to a certain rule. Since non-genuine users seem to narrow down the possibility of the applied rule during an authentication trial by the selected graphic of each answer trial, we think it is effective to use and switch multiple rules in one authentication trial. Furthermore, we consider applying multiple rules at random, rather than in a predetermined order. In this case, genuine users need to know the applied rule each time of answer trials. Inline rule notification is a technique for the system to notify the randomly applied rule to the user by a presented graphic. In other words, the idea of inline rule notification is to embed a graphic to notify the rule in the presented answer graphics.

Figure 7 shows an example of an authentication screen that implements inline rule notification with explanation note written in red. In the setting of this example, the rule is notified by the graphic presented at the center, which we call a notifier graphic, and the notifier graphic can be associated with a square or tetragon. The genuine user sees the tetragon to know the applied rule is tetragon and can find the correct graphic from the remaining 14 graphics that can be associated with a tetragon. Both the notifier and remaining graphics are one-time graphics generated by an authentication graphic generation rule as the same as the original method, and the remaining graphics has only one correct graphic.

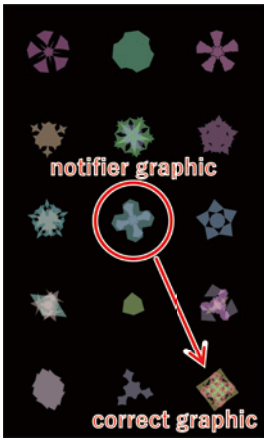

Figure 7: Inline rule notification. A genuine user has specified where to notify and which rules to be used beforehand, and the user is notified the randomly selected generation rule from the notifier graphic in authentication phase.

The location of a notifier graphic, notifier and rule candidates of random selection are preference settings of a genuine user, which must not be disclosed to others. Notifier candidates are a set of rules used for generating the notifier graphic. Rule candidates are a set of rules used for generating the remaining graphics. Note that notifier candidates and rule candidates need to define correspondence but can be either the same or different. For example, if a digon appears as a notifier graphic, the user can define either digon rule or trigon rule is applied for generating the remaining graphics. For the latter case, although the notifier graphic is a digon, the user needs to find a trigon on the screen. This preference can be set across categories, such as digon notifier for red rule. We also note that it is necessary to use multiple locations for rule notification. For example, it is good to set different notifier locations depending on the number of times of answer trials in an authentication trial. Since the correct graphic is not placed at the location of the notifier graphic, if the notifier is set to be always the same location, the notifier location can be easily estimated as the location that the genuine user does not select.

Our user study showed that the use of inline rule notification provides more safety to shoulder surfing attacks than the original method. Before the user study, we asked a person that was not the experimenter nor participants to determine preference settings by repeatedly using the application and trying various settings of inline rule notification. As a result, the setting used in the user study was determined as 9 graphics on a screen at a time, 3 answer trials for an authentication trial, exact same notifier candidates and rule candidates. For the user study, we prepared a video where the experimenter repeatedly tried authentication simulating shoulder surfing by the angle of the video camera. We also edited raw video footage to extract 10 consecutive authentication trials selected at random and added a text that indicates the number of times of the total 10 trials (Figure 8).

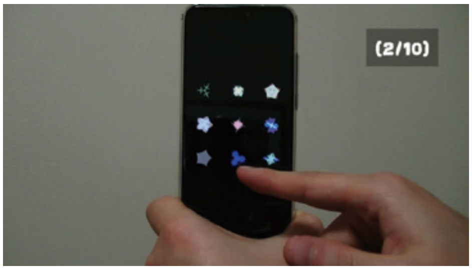

Figure 8: Video screenshot used for the user study of inline rule notification.

In the user study, the experimenter explains all 12 authentication graphic generation rules of 4 category and the mechanism of both the original method and inline rule notification. After the participant reports to reach maximum comprehension, the experimenter introduces the video clip of simulating shoulder surfing, and the participant is asked to try the smartphone application of the same setting as the video as many times as the participant wants. The participant is also told that the participant can ask questions other than settings to the experimenter, and the session will end if the participant is successfully authenticated or gives up.

24 university students participated in the user study. From the 24 participants, we acquired a total of 201 authentication trials. Of these trials, only one trial was a successful authentication. In other words, only one participant could be authenticated by the system with inline rule notification, and the other 23 participants could not be authenticated. The participant made a successful authentication, which was the 20th participant, the participant seemed to find a strategy to overcome this challenge during video watching. After that, the participant made a couple of questions including "Is it true that the location of the correct graphic never be the location of the notifier graphic?" This is true and actually a good strategy to narrow down notifier location. The participant spent 25 minutes to repeatedly watch the video after the questions and made a successful authentication after 5 failure trials. The other participants in this evaluation experiment basically reported that they could not find any cues of settings. We consider this variation with inline rule notification can be solved theoretically by repeated shoulder surfing and guessing attacks, but it is very hard in a practical trial number and time.

7 Conclusion

This paper proposed a novel approach of generating graphics to image-based authentication based on one-time shape-pattern generation aiming at mitigating the shoulder surfing problem. In the proposed method, a one-time correct graphic and dummy graphics are generated based on predefined shape-pattern generation rules. Since different graphics are displayed at different location each time, it can be difficult for another person to acquires the shape-pattern generation rule even if conducting shoulder surfing. As a result of examining the generation algorithm, we defined 12 authentication graphic generation rules categorized into four groups for the proposed method.

The results of the experiments showed effectiveness of the proposed method. The proposed method is fairly usable in terms of first-time user and authentication time. We also introduced inline rule notification for further shoulder surfing defense as a possible solution to the limitation of the proposed method shown in the evaluation experiment.

References

[1] Aviv, A.J., Davin, J.T., Wolf, F., Kuber, R.: Towards Baselines for Shoulder Surfing on Mobile Authentication, *Annual Computer Security Applications Conference*, pp.486–498 (2017).

[2] Dhamija, R., Perrig, A.: Déjà Vu: A User Study Using Images for Authentication, *USENIX Security Symposium* (2000).

[3] Tari, F., Ozok, A.A., Holden, S.H.: A Comparison of Perceived and Real Shoulder-surfing Risks between Alphanumeric and Graphical Passwords, *Symposium on Usable Privacy and Security*, pp.56–66 (2006).

[4] Takada, T., Koike, H.: Awase-E: Image-Based Authentication for Mobile Phones Using User's Favorite Images, *Human-Computer Interaction with Mobile Devices and Services*, pp.347–351 (2003).

[5] Wiedenbeck, S., Waters, J., Sobrado, L., Birget, J.C.: Design and Evaluation of a Shoulder-Surfing Resistant Graphical Password Scheme, *International Working Conference on Advanced Visual Interfaces*, pp.177–184 (2006).

[6] Man, S., Hong, D., Matthews, M.: A Shoulder-Surfing Resistant Graphical Password Scheme - WIW, *Security and Management*, pp.105–111 (2003).

[7] Zhao, H., Li, X.: S3PAS:A Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication Scheme, *International Conference on Advanced Information Networking and Applications Workshops*, pp.467–472 (2007).

[8] Forget, A., Chiasson, S., Biddle, R.: Shoulder-Surfing Resistance with Eye-Gaze Entry in Cued-Recall Graphical Passwords, *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp.1107–1110 (2010).

[9] Zakaria, N.H., Griffiths, D., Brosto., S., Yan, J.: Shoulder Surfing Defence for Recall-based Graphical Passwords, *Symposium on Usable Privacy and Security*, Article No.6 (2011).

[10] Harada, A., Isarida, T., Mizuno, T., Nishigaki, M.: A User Authentication System Using Schema of Visual Memory, *International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pp.338–345 (2006).

[11] Yamamoto, T., Harada, A., Isarida, T., Nishigaki, M.: Advantages of User Authentication Using Unclear Images –Automatic Generation of Decoy Images–, *IEEE International Conference on Advanced Information Networking and Applications*, pp.668–674 (2009).

[12] Miyashita, Y., Higuchi, S., Sakai, K., Masui, N.: Generation of fractal patterns for probing the visual memory, *Neuroscience Research*, Vol.12, No.1, pp.307–311 (1991).

[13] Karras, T., Laine, S., Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.4401–4410 (2019).

[14] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and Improving the Image Quality of StyleGAN, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.8110–8119 (2020).

[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets, *International Conference on Neural Information Processing Systems*, pp.2672–2680 (2014).