Secure Human Authentication with Graphical Passwords

Zayabaatar Dagvatur[1], Aziz Mohaisen[2], Kyunghee Lee[3], DaeHun Nyang[1]

[1] Computer Science Department, InHa University, Republic of Korea
[2] Department of Computer Science, University of Central Florida, United States of America
[3] Computer Science Department, University of Suwon, Republic of Korea
zayabaatar@eit.edu.mn, mohaisen@cs.ucf.edu, khlee@suwon.ac.kr, nyang@inha.ac.kr

Abstract

Both alphanumeric and graphical password schemes are vulnerable to the shoulder-surfing attack. Even when authentication schemes are secure against a single shoulder-surfing attack round, they can be easily broken by intersection attacks, using multiple shoulder-surfing attacker records. To this end, in this paper we propose a graphical password-based authentication scheme to provide security against the intersection attack launched by an attacker who may record the user's screen, mouse clicks and keyboard input with the help of video recording devices and key logging software. We analyze our scheme's security under various threat models and show its high security guarantees. Various analysis, usability studies and comparison with the previous work highlight our scheme's practicality and merits.

Keywords: Security, Usability, Shoulder-surfing, Graphical passwords, Authentication

1 Introduction

Password-based schemes are one of the most common authentication techniques today. In its simplest form, a password-based authentication technique requires a user identifier and a password, both of which are often strings of alphanumeric. While this approach is popular due to its convenience and usability features, it is vulnerable to attacks due to the short and predictable passwords. Having little knowledge of password cracking technique leads users to create and use predictable passwords [1]. Addressing such vulnerability by increasing the length of the alphanumerical password results in poor usability, thus calling for alternatives that are more secure against guessing attacks without sacrificing usability. In search for such alternative, harder-to-guess graphical password-based techniques are proposed in the literature. Since such research has focused on countering guessing attacks, guessing attack is not the biggest threat to graphical password-based authentication schemes.

Both alphanumerical and graphical password schemes are vulnerable to shoulder-surfing attacks [2-3], where an adversary who is capable of recording a user's login procedure can identify the user's password. To defend against this attack, a number of graphical password schemes were proposed in the literature. They succeeded in hiding secrets from observers.

While addressing simple forms of the attack on graphical password-based authentication schemes is possible, advanced attacks are challenging. For example, an attacker who can record user authentication sessions over time can mount intersection attacks and find a user's secret by looking for common images from various authentication sessions.

We design a graphical password authentication scheme secure against the intersection attack while maintaining security against guessing attack within acceptable range. We show our scheme is secure against intersection attacks, because: (1) occurrence frequency of user secret elements and that of decoy elements is configurable to be the same. (2) The response from the user is different for each authentication attempt, even when the same graphical password is used. (3) The possible passwords that can be inferred from a user's response aren't unique. Our scheme is also user-friendly.

2 Related Works

A number of graphical password schemes have been developed [4-8] as an alternative to alphanumerical passwords. Those schemes emphasize that humans are good at recognizing previously seen images rather than recalling a secure alphanumeric password (i.e., often random strings for better security), which is notoriously difficult to remember. Indeed, several psychological and user studies support such observation, which hinders usability for security [5, 9]. Thus, memorability of a graphical password also offers larger possible password space, which exceeds that of alphanumeric passwords and consequently increases the security of the password. A typical graphical password scheme requires a user to select or recognize password images among a larger group of decoy

*Corresponding Author: DaeHun Nyang; E-mail: nyang@inha.ac.kr
DOI: 10.3966/160792642019072004023

images.

Most graphical password based authentication schemes are vulnerable to shoulder-surfing. Especially, graphical password schemes like DAS [4] and Passpoint [5] require users to click secret image or to draw a secret pattern, thus are vulnerable to this attack. Also, a majority of the recognition-based graphical password schemes are also subject to the shoulder-surfing attack [10].

A number of researches are proposed to defend against shoulder surfing attack [4, 11-15, 17, 23-25], and they are secure against human observer and good for single round authentication. But in most of the cases [6, 17, 23-24], the adversary can obtain the secrets after recording a few authentication rounds. A good example is provided in [16], in which user's secret images of CAS scheme [15] are recovered using a SAT solver after observing a few successful logins.

Biddle et al. [10] surveyed a number of graphical password schemes. For the evaluation of authentication schemes, Bonneau et al. [18] developed a comprehensive framework considering usability, deployability and security. Nyang et al. [19] used Bonneau et al.'s framework to evaluate an authentication scheme resistant to keyloggers.

3 Threat Model

We model a shoulder surfer as an adversary who records the user's screen and keyboard inputs with the help of external recording devices and key logging software, respectively [20]. As such, the adversary can record both challenge set and user response of the authentication mechanism. In addition, we assume that the adversary who recorded multiple authentications of a user is able to mount a statistical intersection attack using all previous user inputs and challenge sets. Subsequently, the adversary is able to collect the whole password pool through repeated observations of the input system.

3.1 Challenge Set Only Intersection Attack

After observing several successful authentication sessions, attackers may use intersections of challenge sets to reveal password elements. In some graphical password schemes, secret images of the user appear more often in the challenge than the decoy images, which makes them vulnerable to this simple intersection attack.

3.2 Known Response Intersection Attack

A shoulder surfer capturing users' login screen can obtain all cell pairs by looking up user's response in the response table discussed in section 5.1.2. The adversary can then mount intersection attacks to reveal the user's secret. In this type of attack, the adversary tries to find out users' secrets using a challenge grid and user's responses.

3.3 Intersection Attack with Known Response

Here we define another type of intersection attack that aims to obtain a subset equivalent to a secret subset. The attacker tries to get the images that belong to one of the subsets defined in section 5.3. If the attacker successfully finds a subset, an attacker can pass the test without knowing the exact secret subset of a given user. The algorithm of this attack is shown in Figure 4. And the security analysis of this attack against the proposed scheme is discussed in section 6.2.3.

4 Vulnerabilities of Existing Schemes

Zakaria et al. [15] have improved the recall-based graphical passwords such as DAS [4] and BDAS [16] by drawing additional strokes and removing user drawn strokes to resist shoulder-surfing. While they may have indeed succeeded in defending user secret from a human observer, this scheme is not secure against the adversary defined in section 3 who has the ability to record user's mouse clicks. The CHC scheme by Sobrado and Birget [11] and the improved one by Wiedenbeck et al. [12] are not easily broken in a single observation by a powerful adversary who records every user action. However, these schemes are vulnerable to simple intersection attack defined in 3.1, because the secret images appear more often than the decoy images. This kind of vulnerability is common in early graphical password systems and the literature suggests using same decoys at each authentication attempt [21].

Hong and Mang's WIW schemes (in [13] and [14]) ask the user to remember 4×n icons (4 different variations of n icons) and the corresponding 4×n text codes of each icon. The authors assumed that the variations of the icons are hard for computer vision (spyware) to recognize. However, we notice that the spyware does not need to recognize the icons, and only needs to have the ability to distinguish the icons to launch the attack. Since the text codes (user responses) are linked with the secret images, the adversary defined in section 3 can mount intersection attack to get the user's secret images.

The spin wheel based scheme in [23] is hard for human observers to find the secret, but it is noticeably easy for the adversary defined in section 3.1. The adversary gets clues (aligned characters) about the secret in every authentication. And by intersecting the challenge of two or more authentication, he can learn the secret whether or not the characters are randomly filled.

The PassMatrix scheme in [25] is secure against attackers who are not able to perceive the login indicator from the challenge. However, the adversary defined in section 3 is able to discover the login

indicator and therefore can learn the secret.

5 Shoulder Surfing Resistant (SSR)

A user in our scheme shares some secret images (such as those shown in Figure 1) with a server and the user is challenged with a grid that is composed of password images and non-password images. To be authenticated, the user must find two cells that have the shared secrets. There are only two user secrets in the challenge grid. Upon finding the two cells, the user is asked to find the response, using the location of the two cells, from a response table.

(a) Secret image 1 (b) Secret image 2

(c) Secret image 3 (d) Secret image 4

Figure 1. Example graphical password

5.1 Overview

Here we give a brief description of the structure of the challenge grid and the authentication procedure. For clarity, an authentication example is given.

5.1.1 Challenge Grid

The challenge in our authentication protocol is an m by n grid, which consists of $m \times n$ cells as shown in Figure 2. Every cell of the challenge grid has k images and an index number on the right of it. In Figure 2, the grid is $m \times n = 4 \times 4$ and $k = 4$.

5.1.2 Response Table

A response table (as shown in Figure 3) is an $m \times n$ by $m \times n$ static (constant) table of characters where the rows and columns represent the cell index number of the challenge grid.

Figure 2. Challenge sent to user: A 4×4 challenge grid with four images in each cell

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	-	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	1
2	A	-	C	D	E	F	G	H	I	J	K	L	M	N	O	B	2
3	B	C	-	E	F	G	H	I	J	K	L	M	N	O	A	D	3
4	C	D	E	-	G	H	I	J	K	L	M	N	O	A	B	F	4
5	D	E	F	G	-	I	J	K	L	M	N	O	A	B	C	H	5
6	E	F	G	H	I	-	K	L	M	N	O	A	B	C	D	J	6
7	F	G	H	I	J	K	-	M	N	O	A	B	C	D	E	L	7
8	G	H	I	J	K	L	M	-	O	A	B	C	D	E	F	N	8
9	H	I	J	K	L	M	N	O	-	B	C	D	E	F	G	A	9
10	I	J	K	L	M	N	O	A	B	-	D	E	F	G	H	C	10
11	J	K	L	M	N	O	A	B	C	D	-	F	G	H	I	E	11
12	K	L	M	N	O	A	B	C	D	E	F	-	H	I	J	G	12
13	L	M	N	O	A	B	C	D	E	F	G	H	-	J	K	I	13
14	M	N	O	A	B	C	D	E	F	G	H	I	J	-	L	K	14
15	N	O	A	B	C	D	E	F	G	H	I	J	K	L	-	M	15
16	O	B	D	F	H	J	L	N	A	C	E	G	I	K	M	-	16
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Figure 3. Response table. There are 15 characters (A to O) in each row and column. The user has to pick one character from this table

For the actual use of our scheme, a response table can be printed and distributed to users in advance or can be presented and viewed during the authentication process. The system does not have to make response tables for every user, and the same response table can be shared among users, thus improving usability of the proposed scheme.

5.1.3 Authentication

The user has to find two cells that have his secret images from the challenge grid. He also has to remember the indexes of the cells. After recognizing the secrets, the user checks the response table to look up the response character using two indexes of the cells where the secrets are located. Accordingly, the user inputs the response character to be authenticated.

5.1.4 Authentication Example

We assume a user with four secret images as highlighted in Figure 1. The server challenges the user with the grid shown in Figure 2. First, the user finds the location of his two secret images (the headphone) and (the notebook) in cells 11 and 14, respectively. The user remembers the two indexes (11 and 14) and looks them up in the response table. The user finds the corresponding answer of "H" in the 11th row and 14th column in the response table. The same response can be found in the 14th row and 11th column. The user inputs the response to the challenge as "H" and is authenticated by the server.

5.2 Response Table Construction

When the system is installed for the first time, the server constructs an $m \times n$ by$m \times n$ response table. The values in the response table can be alphabetic characters, special characters or numbers. The response table is constructed such that all of the following conditions are satisfied. (i) It should be diagonally symmetric. (ii) There should be $m \times n$ copies of the same value. (iii) Every row (and every column) has only one instance of every response value.

In Figure 3, $m=n=4$ and every alphabetic character occurs exactly 16 (4×4) times. The purpose of the table is to obscure the relationship between the user secret and the response by organizing the table such that one character appears exactly $m \times n$ times in different column and different row throughout the table. By doing that, even when the attacker watches the input, he cannot decide which cell the user picked among the $m \times n$ candidates. We note that the response table might be shared among users within the same system, or a different response table maybe allocated to each individual user.

5.3 Registration Phase

The server constructs a password pool (denoted by a set P) whose size is p, and sets s, k, m and n based on the desirable security level for every new user. An element of the password pool is an image. The pool size p is configured such that $2 \times p/s = k \times m \times n$. That is, $p = s \times k \times m \times n/2$.

Then, the user selects s elements from the password pool as his secrets. The s secrets form a subset which is called the secret subset and is denoted by S. The p-s remaining elements in the password pool are called decoy elements. Decoy elements are also randomly allocated into subsets (referred to as decoy subsets) whose size is the same as the secret subset and is denoted by D_1, D_2, …, $D_{(p/s)-1}$. As a result, we obtain p/s subsets with size s.

5.4 Authentication Phase

5.4.1 Grid Construction

The authentication challenge for a user is an m by n grid of cells, each of which contains k elements from the password pool P. There are $m \times n$ grid cells in a challenge grid, and each grid cell contains k images. Therefore, to fill an entire challenge grid, $k \times m \times n$ images are required.

Because p was configured such that $2 \times p/s = k \times m \times n$ is satisfied, an entire challenge grid can be filled with pairs of images chosen randomly from every p/s subsets of the password pool P. Accordingly, the grid is prepared by the server as follows:

(1) The server randomly selects a response value (R) from the response table.

(2) The server locates R in the response table. Because every character in the response table appears $m \times n$ times, there are $m \times n$ cells that have R. Because the response table is diagonally symmetric, only $m \times n = 2$ cells among the m×n cells are used to construct the challenge grid. Now, let $\{(u_1, v_1), (u_2, v_2), \ldots, (u_{mn/2}, v_{mn/2})\}$ be the set of cell indexes' where R resides in the upper right triangle(that is, $u_i < v_i$).

(3) The server randomly selects p/s pairs of elements from the sets S, D_1, D_2, …, $D_{(p/s)-1}$, which include one secret subset S and all decoy subsets. Here, the number ofpairs are $p/s = kmn/2$, where p is the pool size and s is a subset size. We will denote them by $I = \{(a_1, b_1), (a_2, b_2), .. , (a_{kmn/2}, b_{kmn/2})\}$.

(4) The server repeats the following steps for each (u_i, v_i), where $i = 1, \ldots mn/2$:

(a) It chooses randomly k image pairs $\{(a_{i_1}, b_{i_1}), ..., (a_{i_k}, b_{i_k})\}$, from I without replacement.

(b) It allocates randomly $\{a_{i_1}, ..., a_{i_k}\}$ to the grid cell u_i.

(c) It allocates randomly $\{b_{i_1}, ..., b_{i_k}\}$ to the grid cell v_i.

As a result, two images from the same subset are located in two different grid cells.

5.4.2 Authentication Protocol

When a challenge query C is shown, a user should mentally locate two cells which contain his secret elements. According to the grid construction method in section 5.4.1, there are exactly two secrets displayed in each challenge grid, with the condition that these secret elements will not be within the same cell. Accordingly, and based on indexes of the two cells, the user looks up a response value from the response table. The authentication procedure is explained step by step as follows:

(1) The server sends an m by n query grid C after constructing C using the grid construction algorithm in section 5.4.1.

(2) The user mentally locates two cells including two secrets in the challenge grid (say, *u* and *v*).

(3) The user looks up a cell whose column is *v* and row is *u* (or of *u*-th column and *v*-th row) from the response table.

(4) The user inputs the response value *R* in the cell found in step (3).

(5) The server checks the correctness of *R*, and if correct, challenges the user with another grid by repeating steps (1) through (5).

(6) The user is authenticated after λ number of successful and successive rounds of authentication.

6 Security Analysis

6.1 Guessing Attack

Let *g* be the number of distinct user responses. For example, in the response table of Figure 3, g is 15. We define security level as the probability to successfully guess the input. In our proposal it is given as *1/g* in a single authentication round. After *r* authentication rounds, this probability becomes to $1/g^r$. Thus, the security depends on the number of authentication rounds. At the highest security level of our scheme, an attacker has chance of 1 in 2.5 billion (for *r*=8). The version used in comparison (section 8) has 1/750,000 (where *r*=5) security level against the guessing attack. In order to maintain such level of security, the number of wrong trials should be limited in order to defend against a guessing attack.

6.2 Shoulder-surfing Attack

The security analysis of the proposed scheme against intersection attack over multiple rounds of authentication is discussed in the following subsections. Due to lack of space, attack examples with illustrations have been omitted.

6.2.1 Challenge Set Only Intersection Attack

Security analysis of simple intersection attack (uses only challenge sets) defined in section 3.1 is discussed in this section. A countermeasure to this attack can be realized if the secret element appears no more than the decoy elements. Let *s* be the size of the secret element set *S*, and *p* be the size of the password pool *P*. Then, we have (*p*-*s*) number of decoy elements. If *k* elements are combined in one cell, the total number of elements to appear in one challenge is *kmn*, where the challenge grid has *m* rows and *n* columns. The number of secret elements among *kmn* elements is 2. Thus, the probability that the secret elements will appear in the challenge should be equal to the probability that the decoy elements will appear in that challenge. Therefore the following equation should hold:

$$\frac{kmn-2}{p-s}=\frac{2}{s} \quad \textbf{(1)}$$

Using (1), the numbers of secret and decoy elements are decided so that the distribution of the number of secret images is the same as that of the decoy images. That is, even though the attacker counts the number of every image for a long time, he will see the same number of occurrences of every image and thus the secret images are not distinguishable.

Validation: We implemented an intersection attack that calculates the intersection with the challenge sets until a certain number of images remains (we use that number as a range of 1, …, 20 images). To obtain the attacker's success probability on average, the experiment was conducted 100,000 times and the results were depicted in Table 1.

Table 1. Attacker's success probability (sp) and required number of rounds (r) to launch intersection attack that uses only challenge set for different intersection set size

Iss	1	2	3	4	5	10	15	20
Sp	0.031	0.031	0.031	0.030	0.031	0.031	0.031	0.031
	16	15	13	88	52	24	18	26
r	7.696	6.8	6.192	5.736	5.35	4.28	3.33	3.28

From this experiment, and for the case where the experiment is performed until only one image remains, we found that only 3,116 cases out of the 100,000 authentication attempts have 4 secret images where the others are decoy images. Thus, we find that (on average) the success probability of the attack is 0.03116. Similarly, in the same experiment and for intersection set of size 2, 3, 4, 5, 10, 15, and 20, we obtain success probabilities of 0.03115, 0.03113, 0.03088, 0.03152, 0.03124, 0.03118, and 0.03126, respectively.

We found that after 4.1 rounds only 12 images are left, the secret images disappear from the intersection set. Considering that the percent of the secrets in the password pool was 100×4/128=3.125 (corresponding to a probability of 0.03125) and the total probability of success in the above experiment of 0.03126, we confirm that an attacker gains no advantage by launching this attack.

6.2.2 Known Response Intersection Attack

The attacker collects authentication information (challenge set and user responses) over multiple sessions and uses them to recover user's secret images.

By logging user's input, an attacker can search for pairs of cells having the same response as the user's input. For example, there are 8(=*mn*/2) number of cell pairs for the user's input "K", which are (1, 12), (2, 11), (3, 10), (4, 9), (5, 8), (6, 7), (13, 15) and (14, 16) in Figure 3. We will denote this as a candidate cell list,

and note that the number of cells in the candidate cell list is $m \times n$. If the number of cells determined by the user's input is less than $m \times n$, the attacker is able to mount an intersection attack by excluding from a challenge grid the images in the cells that are not in the candidate cell list. Defense against the attack discussed above can be achieved by carefully arranging response values in the response table.

Response values should be arranged in a way that the number of cells determined by every response value in the response table must be $m \times n$, which is equal to the number of cells of a challenge grid. Thus, in the response table, there must be exactly *mn/2*instances of every response number to include all cells (=*mn*) of a challenge grid. Because there are $\binom{m}{2}$ number of pairs of cells in the response table and eachshould occur at least *mn*/2 times, the number of distinct user responses, g becomes:

$$g = \frac{\binom{m}{2}}{mn/2} = \frac{2}{mn} \times \binom{m}{2} = mn - 1 \qquad (2)$$

As shown in Figure 3, g=15 when m=n=4. A higher value of g promises higher security against a guessing attack. But the upper bound for g is already determined by the above equation, and to use large g requires large m and n, which decreases usability.

6.2.3 Intersection Attack with Known Response

It is possible to mount a different type of an intersection attack to collect s images that belong to one of the subsets S, D_1, D_2, ..., $D_{(p/s)-1}$. After successfully finding a subset, an attacker can pass the test without knowing the exact secret subset of a given user.

To understand such attack, we first use the term "subset pair" to refer to an image pair where both images are from the same subset, whereas a "non-subset pair" is one that has two images from different subsets. An attacker does not need to find the exact secret subset S, but it is enough to find any one correct subset among p/s subsets (S, D_1, D_2, ..., $D_{(p/s)-1}$) for a predetermined user at the time of registration. This is owing to the fact that any subset pair of images is not only from the secret subset S but also from the other decoy subsets that gives an attacker the correct response. Note that the non-subset pair does not generate a correct response value. Thus, in the following are going to analyze the security of our proposal in view of an attacker who tries to find out any subset among S, D_1, D_2, …, $D_{(p/s)-1}$.

For that, the attacker begins by randomly choosing one image from the first challenge grid and tries to find the images that are in the same subset with the chosen image. Let us first denote the chosen image by t_1. In every challenge grid that contains t_1, the attacker can get k candidate images, where one of them is in the same subset as t_1. The attacker's strategy is then to count the occurrences of all these candidate images across many sessions and get the s most occurring images.

A detailed description of this attack is shown in Figure 4.

The attacker who has collected r_{max} successful authentication sessions can use the algorithm to find a subset equivalent to the secret subset S. Occurrences of the candidate images are counted by the algorithm shown in Figure 5 and the array I_t holds the number of occurrences. As we mentioned before, the response table is public, so it is known to the adversary too.

To test whether s images construct subset or not in Figure 4, we can use the subset test algorithm shown in Figure 6.

Validation: We implemented the above intersection attack under a known response and conducted the experiment 100,000 times for different numbers of secrets (with the value of s taking the values 4, 5, 6, 7, and 8). The summary of this experiment is shown in Table 2. In case of the attack for s=4 the shortest attack took 6 rounds, but the longest attack took 133 rounds to successfully find the subset. In Figure 8(c), it takes 29, 58, 98, 150 and 216 rounds for the attacker to successfully find all the elements in a subset of size 4, 5, 6, 7 and 8, respectively. Thus, the number of shoulder-surfing attempts to launch this attack successfully depends on the number of rounds for one authentication session.

Table 2. Summary of intersection attacks under known responses

		Attack rounds			
S	p	Min	Max	Average	Median
4	128	6	133	29.2	69.5
5	160	9	268	58.04	138.5
6	192	17	420	98.2	218.5
7	224	20	615	150.95	317.5
8	256	39	800	216.95	419.5

6.2.4 Parallel Known Responses Intersection Attack

The number of required rounds can be reduced by running the attack shown in Figure 4 for all images in parallel. Although this attack requires more memory and takes more time, it can find the images in one subset in fewer sessions than the previous attack. The advantage of this attack over the previous attack is that here the attacker refers to intermediate results that are likely in the same subset as t_1 and are obtained by running the algorithm in Figure 4 in parallel for all images. In other words, the attacker sees more images that make up t_1, so he has more clues to find the subset having t_1. A pseudo-code of this parallel attack is described in Figure 7.

Known-Response-Intersection-Attack

Input: $C_1, \ldots, C_{r_{max}}$ are eavesdropped challenge grids, and $R_1, \ldots, R_{r_{max}}$ are the corresponding user responses.
Output: s images that are in the same subset

1. Choose randomly one image from the first challenge grid C_1, which is denoted as t_1.
2. For $r=1,\ldots,r_{max}$, do the following until s images from a subset are collected.
   (a) Call Candidate-Subset-Pair-Counter($t_1, C_1, \ldots, C_r, R_1, \ldots, R_r$.
   (b) $t_2, \ldots, t_s \leftarrow$ Images that have the highest values in I_{t_1} (most occurring images)
   (c) By calling Subset-Test($t_1, \ldots, t_s, C_1, \ldots, C_r, R_1, \ldots, R_r$, check if the $s-1$ images and t_1 compose one of subsets ($S, D_1, D_2, \ldots, D_{(p/s)-1}$). If so, return ($t_1, \ldots, t_s$). Otherwise, continue.
3. Return FAIL.

Figure 4. Algorithm of the known response intersection attack

Candidate-Image-Counter

Input: t is an image. $C_1, \ldots, C_{r_{max}}$ are eavesdropped grids, and $R_1, \ldots, R_{r_{max}}$ are the corresponding user responses.
Output: I_t holding the number of occurrences of candidate subset pairs for t.

1. Let I_t be an array that counts the occurrence of the images. It is initialized as 0.
2. For $r=1,\ldots,r_{max}$ such that C_r has t, do the following
   (a) Find the challenge grid cell that has t and denote its index as c_1.
   (b) Go to c_1-th row of the response table and find the cell that has R_r whose column index of the cell is denoted as c_2. Here, note every row has only one R_r owing to the construction method of the response table in section 5.2.
   (c) For k images in the c_2-th cell of challenge grid, let's denote names of the images as $i_{c_2,1}, i_{c_2,2}, \ldots, i_{c_2,k}$, which will be used as indices of the array I_t. Increment $I_t[i_{c_2,1}], \ldots, I_t[i_{c_2,k}]$ by one.
3. Return I_t.

Figure 5. Counting candidate images in the same subset

Subset-Test

Input: $t_1, \ldots, t_s$ are test images. $C_1, \ldots, C_r$ are challenge grids and $R_1, \ldots, R_r$ are response values.
Output: *SUCCESS* if test images are in the same subset, or *FAIL*.

1. For every $C_j (j=1,2,\ldots,r)$, repeat the following:
   (a) Search the images (t_1 *to* t_s) in C_j.
   (b) If the number of images found in step1 is not equal to 2, then return *FAIL*.
   This is because two images from a subset will be arranged in a challenge grid.
   (c) Let u and v are the cell indices of the two images found in step a. Let $R_{u,v}$ be the response value in u-th row and the v-th column of the response table. If $R_{u,v} = R_j$, then return *FAIL*.
2. Return *SUCCESS*

Figure 6. Algorithm to test whether the given images form a subset

Parallel-Known-Response-Intersection-Attack

Input: $C_1, \ldots, C_{r_{max}}$ are eavesdropped challenge grids, and $R_1, \ldots, R_{r_{max}}$ are the corresponding user responses.

Output: s images that are in the same subset

1. Choose randomly one image from the first challenge grid C_1, which is denoted as w_1.
2. For $r = 1, \ldots, r_{max}$ repeat the following:
   (a) For every image i_j where $j=1,\ldots,p$
      i. Obtain I_{i_j} by calling Candidate-Subset-Pair-Counter($i_j, C_1, \ldots, C_r, R_1, \ldots, R_r$).
   (b) For $j=1,\ldots,s-1$,repeat the following:
      i. w_{j+1}←the highest occurring image in I_{w_j}.
      ii. if w_{j+1} is equal to any of $w_1,\ldots,w_j$ then break.
   (c) $t_1, \ldots, t_s$←the most occurring s images from $I_{w_1}, I_{w_2}, \ldots, I_{w_j}$.
   (d) By calling Subset-Test($t_1, \ldots, t_s, C_1, \ldots, C_r, R_1, \ldots, R_r$), test if $t_1, \ldots, t_s$ are from same subset. If so return $(t_1, \ldots, t_s)$ otherwise go to step2.

Figure 7. Algorithm of parallel intersection attack under known responses

Validation: To validate this attack, we implemented the intersection attack under parallel known responses and conducted the experiment 100,000 times for different numbers of secrets. This parallel attack is 30% faster than a single known response attack. The line with squares in Figure 8(c) shows the number of required rounds to find a subset for various numbers of secrets.

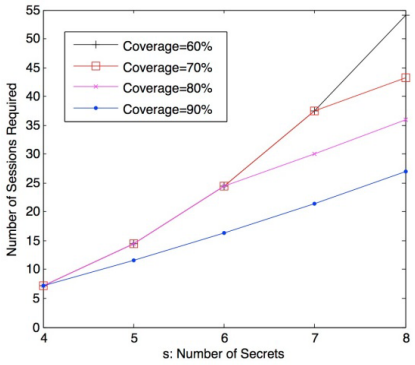
As a reference, the lower bound for the required number of rounds, l, to extract all the key bits in the information-theoretic sense is computed as follows and is shown as the line with circles in Figure 8(c):

$$l = \frac{\log\binom{p}{s}}{\log(g)} \tag{3}$$

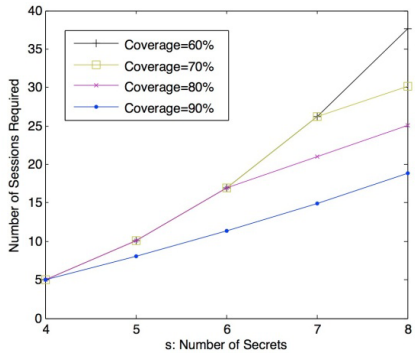
6.3 Recommended Parameter Sets

Round considerations. In our protocol, every round contains only two secret images and those two images are randomly chosen from the secret subset. Consequently, the number of rounds per session determines how many different secrets will be used to authenticate a user. We implemented the server's preparation of challenge grids and tested it 100,000 times to examine how many secrets are covered according to the number of rounds and secrets. Coverage is defined to be a fraction of the number of secret images appearing in r challenge grids of one session out of the number of secret images of a user.

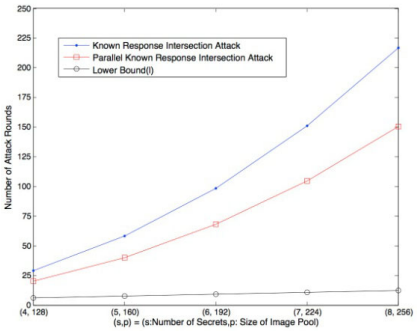
The number of rounds per authentication was chosen as the least value of r that covers more than 60, 70, 80 and 90 percent of secret images, respectively. For example, when s=4,…,8 and the coverage is 70%, the numbers of rounds required are 2, 3, 3, 4, 5, respectively. Therefore, with this setting, the system asks a user to prove the knowledge of at least 70 percent of the secrets for each authentication.



(a) Intersection Attack



(b) Parallel Intersection Attack



(c) Required Number of Rounds to Find a Subset by Two Intersection Attacks vs Lower Bound

Figure 8. Required Number of Sessions to Find a Subset

Validation. Even with the authentication with bounded coverage, it is sufficient to prevent an attacker who has a fraction of secret images from being authenticated. This is because the server may produce many different challenge grids with a secret image that the attacker does not have. Figure 9 shows the attacker's success probability with 60%, 70%, 80%, 90% secret images when the authentication coverageis60%, 70%, 80%, 90%. In the worst case, the success probability is less than 1.4%. Considering that after three successive login failures, the server blocks the account, and the attacker's success probability is negligible.

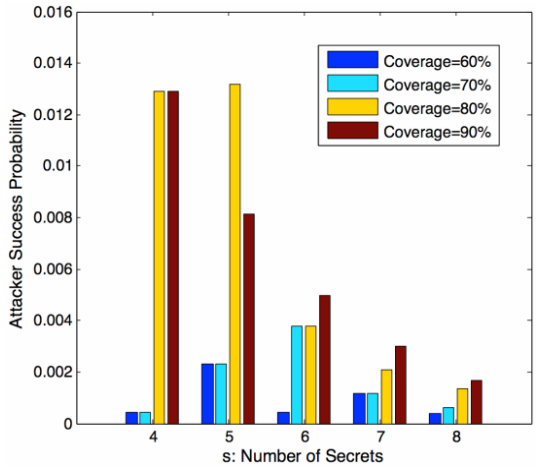


Figure 9. Attacker Success Probability for Different Coverage

Figure 8(a) and Figure 8(b) show the number of sessions required to find a subset for different number of secrets (with s=4, …, 8) and for different secret coverage (coverage is used as 90%, 80%, 70%, 60%) using the attack in Figure 4 and Figure 7, respectively.

Pool and grid size. The password pool size (p) is determined by (1). To increase the security against guessing attacks, the challenge grid size and the number of authentication rounds can be increased.

We recommend the parameter set in Table 3(a) for security against guessing attack. Our recommendations of the parameters for security against intersection attacks are presented in Table 3(b). Last two columns in Tables 3(a) and 3(b) represent the security against guessing attack and security against parallel intersection attack, measured by the number of logins. Highest security against parallel intersection attack is shown to be 150.7 rounds, which becomes 150.7/5=30.14 logins when r=5.

7 User Study

7.1 Experimental Design

On the first day, the participants chose the secret images and learned the secrets by logging into a system that uses our authentication scheme for 5 times. After learning the secrets, participants carried out their first retention trial. The second, third and fourth retention trials took place on week 2, week 3 and week 4, respectively. In the retention trials we measured the number of incorrect inputs and the amount of time for participants to enter a password. Additionally, we measure the time it takes users to recognize the secret images.

Table 3. Recommended Parameter Sets

(a) Set for Security against Guessing

s	P	r	k	m	n	Guessing	Intersection
4	128	4	4	4	4	2^{15}	5.05
5	160	5	4	4	4	2^{19}	8.03
6	192	6	4	4	4	2^{23}	11.31
7	224	7	4	4	4	2^{27}	15
8	256	8	4	4	4	2^{31}	18.83

(b) Set for Security against Intersection

s	P	r	k	m	n	Guessing	Intersection
4	128	2	4	4	4	2^{8}	10.1
5	160	3	4	4	4	2^{11}	13.39
6	192	3	4	4	4	2^{11}	22.62
7	224	4	4	4	4	2^{15}	26.2
8	256	5	4	4	4	2^{19}	30.14

7.2 Participants

Our user study uses 24 participants (9 females and 15 males). Most participants were university students except one high school student. The highest educational level achieved varied from secondary education to a Ph.D. The mean age of the participants was 29 years and the range was from 16 to 45 years old. All participants reported that they used PCs regularly, and used online accounts that required password-based authentication.

7.3 Parameters

The interface we used in our user study is shown in Figure 2. The challenge grid size was 4×4 (where m=n=4) and the number of images in one cell was k=4. The number of secrets was s=4, …, 8 and the corresponding password pool had a total of p=128 to 256 images. The participants are divided into 5 groups according to number of secrets. Number of subjects in groups for s=4, …, 7 and 8 is 9, 4, 4, 4 and 3, respectively.

7.4 Procedure

First, the participants were briefed about the purpose of the authentication protocol, authentication procedure and the experiment. Participants are then registered by choosing their secret images from a display of listed images (the password pool). This registration took 1 to 2 minutes depending on the number of secrets. Before giving the first retention trial, the participants practiced to login and learned the secrets. This introduction took less than 5 minutes in total. The participants then

performed 5 login attempts (over 10 authentication rounds) in every week. One authentication round is composed of three steps: image recognition, table lookup and response entering. The total number of rounds by all participants through all retention tests was 2,010.

7.5 Results

Image recognition time. We measured the amount of time spent in recognizing secret image on the first two weeks. The image recognition time in session 2 was significantly decreased from that in week 1. The average image recognition time in week 1 was 4.65sec and the average image recognition time in week 2 was 3.53 sec, constituting just over 24% reduction in time.

Image memorability. Of all participants, there were no participants who forgot their password images, but a small number of authentication attempts failed (26 out of 2010)—corresponding to 1.29% of the total authentication attempts. Table 4 shows the percentage of the correct rounds for all experiments with different number of secrets. There were 15 participants who successfully finished the total of 40 authentication rounds throughout 4 weeks without a single failure.

Table 4. Percentage of Correct Rounds

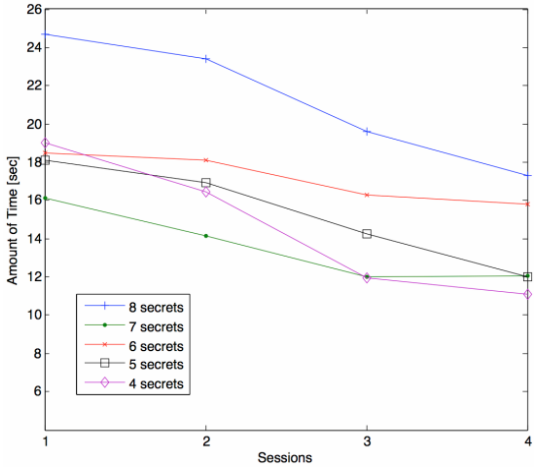
Number of Secrets	Week1	Week2	Week3	Week4
4	99.62	99.62	98.51	99.62
5	100	100	100	100
6	96	98	98	98
7	96	98	100	100
8	94.4	95.5	98.9	100
WeekAverage	97.2	98.22	99.08	99.52

Average login time. The average login time for different number of secrets across four weeks is shown in Figure 10(a). There were significant improvements in performance for each week. The average time per round for week 1, 2 and 3 was 19.1sec, 17sec and 14.5sec respectively. The fourth week's average response time was 13sec, which was 32.1% faster than in first week.

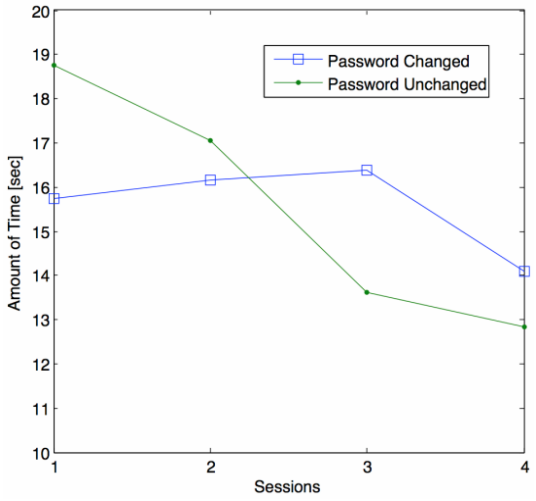
The time spent per round by all participants was15.9 sec, the average image recognition time was 4.09 sec, and the average correctness was 97.8%. With the add security, we believe this usability is within an acceptable range.

7.5.1 Password Change

We analyzed password change and updates effect on usability. A number of participants changed their password after week 2. The result showed that password changes do affect the login time. As shown in the Figure 10(b), the login time gradually decreases across weeks when there was no password change. However, the login time increases after the password change and then decreases for the subsequent week. We further notice that there was no significant change in correctness of the authentication, since the participants refreshed their memory quickly with new graphical passwords.



(a) Average Time spent in one Round



(b) Password change affect in login time (s=6)

Figure 10. User Study Results

7.5.2 Questionnaire

After week 4, participants were asked three questions about the easiness of the authentication protocol. Each question had Likert scale responses from 1 (very hard) to 5 (very easy). The questionnaire results are shown in Table 5. According to the results, participants had different opinion about the image recognition and table lookup. While 8 participants answered that image, recognition is easier than table lookup, 14 participants answered that table lookup is easier than the image recognition part. The majority of the participants responded that the proposed scheme was easy overall.

Table 5. Questionnaire result for easiness

Choice	ImageRecognition	TableLookUp	OverallScheme
5-VeryEasy	3	5	3
4-Easy	13	13	16
3-Normal	11	12	11
2-Hard	3	0	0
1-VeryHard	0	0	0
AverageEasiness	3.53	3.76	3.73

7.6 Other Usability Issues

According to the user study, image recognition time depends on the difficulty (complexity) of the image. The complexity of the image can be determined by the number of objects in that image and the number of colors of those objects. Most of the participants recognized images that have a single object with one color on a white or black background very well, while some individuals expressed that red and green-colored objects were more recognized than other colored objects.

Some participants reported that it would take them more time and would be less convenient to them when they login from notebooks that have small screens. We notice that this issue can be addressed by decreasing the size of the images in password pool (i.e., current 60×60 pixel images can be replaced by 20×20 pixel icons).

8 Comparison with other Schemes

A large number of schemes have been proposed in the literature on graphical password authentication, but not all of them are secure against the shoulder-surfing attack. Therefore, in this section we show a comparison with only graphical password schemes that resist shoulder-surfing adversaries. Four main schemes that pertain to this area were considered for comparison with our scheme; namely CAS [15], WIW [14], CHC [12] and PassMatrix [28]. We also attempt to extend the list of the schemes for comparison with our scheme to other schemes but realize that they are variants to the ones we already select. For example, the variants of WIW in [13] and CHC in [11] and [22], which exhibit the same levels of performance and security. To this end, and using the four major related works, we summarize the comparison result according to the usability and the security criteria in Table 6.

Table 6. Comparison with Existing Schemes (Security for Guessing Attack 2^{-19})

Criteria	SSR	WIW	CHC	CAS	PassMatrix
Number of Rounds Required(r)	5	3	3	10	6
Number of Secrets(s)	8	16	19	30	3
Number of Secrets in Challenge(sc)	2	16	3	30	1
Number of Challenge Images(c)	64	256	189	80	1
Number of Images in Pool(p)	256	256	256	80	3
Number of Distinct Responses(g)	15	256	189	4	77
Correctness of Inputs[percent]	97.7	100	90.351	95	93.33
Number of images per session(ds)	320	768	567	800	6
Communication Time[ms/session](ts)	412	850	653	881	106
Challenge set Only Intersection Attack	Secure	Secure	Secure	Secure	Secure
Key bit security[rounds](l)	12.4	10.3	12.4	37	18.8
Known Response Intersections(rounds/logins)	150.7/30.14	-/-	-/-	60/6	3/1

To compare the security of the schemes fairly and under similar conditions, we normalized the parameters of WIW and CHC as shown in Table 6. The number of images in the pool (p) was fixed to 256 for all schemes. The probability of successful guessing was normalized to $1/2^{19}$ and the security level against the intersection attack under the known response was also fixed by adjusting l, the number of rounds that is necessary to extract all the key bits (c.f. Eq. (3)).

For comparison, we fixed l to 12 and then computed the corresponding s. Thus, we chose s=16 for WIW because $l_{WIW} = \frac{\lg\binom{256}{16}}{\lg(256)} = 10.3$. For CHC, s=19 because it gives $l_{CHC} = \frac{\lg\binom{256}{19}}{\lg(189)} = 12.4..$ Note that g, the number of possible inputs, and c, the number of images in a challenge, are determined by p in both CHC and WIW. Especially, c of CHC is fixed to the same proportion as in the original scheme. That is, c=256×(83/112)=189.

The number of required rounds (r) to achieve $1/2^{19}$ security level against the guessing attack is calculated as $r = \frac{19}{\lg g}$. Communication overhead (ds) per session is calculated as $ds=r\times c\times$1KB, where r is the number of required rounds in one session, c is the number of images in one round and size of an image is assumed to be 1KB(assumed for a fair comparison). The total

communication time spent in one session (*ts*) is computed as $ts=(ds/ns)+nl$, where *ns* is the network speed (assumed to be 1MB/s for a fair comparison) and *nl* is the network latency (assumed to be 100ms for a fair comparison).

In terms of the usability, our scheme requires a user to keep in mind a lesser number of secret images (*s*), which is the main advantage for user convenience. In addition, the proposed scheme uses fewer images in a challenge (*c*) than in the other schemes. Also, our scheme uses fewer secrets (*sc*) in each challenge query, which enables a user to find secret images faster than that in the compared schemes.

From a security point of view, our scheme is strong in addressing intersection attacks under known responses (150 rounds). This is a considerable improvement over CAS, where an attacker can decide images in only 60 rounds [16]. While WIW, CHC and PassMatrix are known to be secure under a single round of authentication only.

In terms of the length of key bits in each scheme, WIW and CHC have the equivalent of 83 and 94 bits, respectively, whereas our scheme has only a 48 bit key length. However, a brute-force attack is not only impractical but also infeasible against most graphical password schemes, because the server usually blocks accounts after certain number of successive login failures. Accordingly, we conclude that the proposed scheme provides higher security against the shoulder-surfer compared to WIW, CHC, CAS and PassMatrix and has a better usability than WIW, CHC and CAS as well

9 Conclusion

We proposed a novel graphical password-based authentication protocol to provide security against the shoulder-surfing attack in which a camera or a spyware record login sessions to recover the graphical password. We tested the usability of the scheme and have shown various desirable features. The system does not reveal the authentication information of the legitimate users even when attackers have the ability to record the user screen, to log keyboard entries with the help of external recording devices and malicious software installed on compromised machines. Strong security against a powerful attacker and acceptable usability of the system suggest that the proposed authentication scheme can be employed as a practical shoulder-surfing resistant authentication system.

We will leave the research on enhancing usability for future work. More concretely, while keeping the same security level, designing compact version of the scheme for small screen devices using fewer images will be a challenging work. Also, one step authentication instead of the current two step design would be preferred in terms of usability if a new design can work without the response table.

Acknowledgements

This research was supported by the Global Research Lab. (GRL) Program of the National Research Foundation (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2016K1A1A2912757). D. Nyang is the corresponding author.

References

[1] B. Ur, J. Bees, S. Segreti, L. Bauer, N. Christin, L. F. Cranor, Do users' Perceptions of Password Security Match Reality?, *CHI'16: 34th Annual ACM Conference on Human Factors in Computing Systems*, San Jose, CA, 2016, pp. 3748-3760.

[2] F. Tari, A. A. Ozok, S. H. Holden, A Comparison of Perceived and Real Shoulder-Surfing Risks between Alphanumeric and Graphical Passwords, *SOUPS '06: Proceedings of the Second Symposium on Usable Privacy and Security*, Pittsburgh, PA, 2006, pp. 56-66.

[3] J.-N. Luo, M.-H. Yang, C.-L. Tsai, An Anti-shoulder-surfing Authentication Scheme of Mobile Device, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1263-1272, July, 2018.

[4] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, A. D. Rubin, The Design and Analysis of Graphical Passwords, *Proceedings of USENIX Security Symposium*, Washington, DC, 1999, pp. 1-1.

[5] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, N. D. Memon, Passpoints: Design and Longitudinal Evaluation of a Graphical Password System, *International Journal of Man-Machine Studies*, Vol. 63, No. 1-2, pp. 102-127, July, 2005.

[6] S. Chiasson, P. C. V. Oorschot, R. Biddle, Graphical Password Authentication Using Cued Click Points, *European Symposium on Research in Computer Security*, Dresden, Germany, 2007, pp. 359-374.

[7] N. J. Carter, Graphical Passwords for Older Computer Users, *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software &Technology*, Daegu, Kyungpook, Republic of Korea, 2015, pp. 29-32.

[8] M. Kameswara Rao, T. Usha Switha, S. Naveen, A Novel Graphical Password Authentication Mechanism for Cloud Services, in: S. Satapathy, J. Mandal, S. Udgata, V. Bhateja (Eds.), *Information Systems Design and Intelligent Applications, Proceedings of Third International Conference INDIA*, Vol. 1, Springer India, New Delhi, 2016, pp. 447-453.

[9] A. D. Angeli, L. M. Coventry, G. Johnson, K. Renaud, Is a Picture Really Worth a Thousand Words? Exploring the Feasibility of Graphical Authentication Systems, *International Journal of Man-Machine Studies*, Vol. 63, No. 1-2, pp. 128-152, July, 2005.

[10] R. Biddle, S. Chiasson, P. C. Van Oorschot, Graphical Passwords: Learning from the First Twelve Years, *ACM Computing Surveys*, Vol. 44, No. 4, Article No. 19, August, 2012.

[11] L. Sobrado, J.-C. Birget, Graphical Passwords, *An Electronic Bulletin for Undergraduate Research*, The Rutgers Scholar, 2002.

[12] S. Wiedenbeck, J. Waters, L. Sobrado, J.-C. Birget, *Design and Evaluation of a Shoulder-Surfing Resistant Graphical Password Scheme*, AVI, Venezia, Italy, 2006, pp. 177-184.

[13] S. Man, D. Hong, M. M. Matthews, A Shoulder-surfing Resistant Graphical Password Scheme- WIW, *Proceedings of the International Conference on Security and Management*, Las Vegas, NV, 2003, pp. 105-111.

[14] D. Hong, S. Man, B. Hawes, M. M. Matthews, A Graphical Password Scheme Strongly Resistant to Spyware, *Proceedings of the International Conference on Security and Management*, Las Vegas, NV, 2004, pp. 94-100.

[15] D. Weinshall, Cognitive Authentication Schemes Safe against Spyware (Short Paper), *IEEE Symposium on Security and Privacy*, Berkeley/Oakland, CA, 2006, pp. 295-300.

[16] P. Golle, D. Wagner, Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract), *IEEE Symposium on Security and Privacy*, Berkeley, CA, 2007, pp. 66-70.

[17] Q. Yan, J. Han, Y. Li, J. Zhou, R. H. Deng, Leakage-resilient Password Entry: Challenges, Design, and Evaluation, *Computers & Security*, Vol. 48, pp. 196-211, February, 2015.

[18] J. Bonneau, C. Herley, P. C. van Oorschot, F. Stajano, The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes, *IEEE Symposium on Security and Privacy*, San Francisco, CA, 2012, pp. 553-567.

[19] D. Nyang, A. Mohaisen, J. Kang, Keylogging-resistant Visual Authentication Protocols, *IEEE Transactions on Mobile Computing*, Vol. 13, No. 11, pp. 2566-2579, November, 2014.

[20] D. Florencio, C. Herley, How to Login from an Internet Cafe Without Worrying about Keyloggers, *Symposium On Usable Privacy and Security*, Pittsburgh, PA, 2006, pp. 1-2.

[21] K. V. Renaud, Guidelines for Designing Graphical Authentication Mechanism Interfaces, *International Journal of Information and Computer Security*, Vol. 3, No. 1, pp. 60-85, June, 2009.

[22] H. Zhao, X. Li, S3pas: A Scalable Shoulder-surfing Resistant Textual-graphical Password Authentication Scheme, *AINA Workshops*, Niagara Falls, Ont., Canada 2007, pp. 467-472.

[23] M. K. Rao, S. G. Santhi, M. A. Hussain, Spin Wheel Based Graphical Password Authentication Resistant to Peeping Attack, *International Journal of Engineering and Technology*, Vol. 7, No. 2.7, p. 984-987, 2018.

[24] M. Martinez-Diaz, J. Fierrez, J. Galbally, Graphical Password-Based User Authentication with Free- Form Doodles, *IEEE Transactions on Human-Machine Systems*, Vol. 46, No. 4, pp. 607-614, August, 2016.

[25] H.-M. Sun, S.-T. Chen, J.-H. Yeh, C.-Y. Cheng, A Shoulder Surfing Resistant Graphical Authentication System, *IEEE Transactions on Dependable and Secure Computing*, Vol. 15, No. 2, pp. 180-193, March-April, 2018.

Biographies

Zayabaatar Dagvatur received the B.Eng. degree in computer engineering in 2002, from Inha University, Incheon, Korea, where he is currently pursuing the Ph.D. degree. His research interests lie in computer security, human authentication, network security, biometrics, and mobile security.

Aziz Mohaisen obtained his Ph.D. degree in Computer Science from the University of Minnesota in 2012. He is currently an associate professor of Computer Science at the University of Central Florida. His research interests include systems security, data privacy, and measurements. He is a senior member of ACM and IEEE.

KyungHee Lee received her PhD degree in computer science from YONSEI University in 2004. From 1993 to 1996, she worked for LG Software Ltd. From 2000 to 2005, she was a senior research at ETRI, before moving to University of Suwon as a professor. Her research interests include computer security, vision computing, artificial intelligence, pattern recognition, and biometrics.

DaeHun Nyang received dual B.Eng. in EE and CS from KAIST, and M.S. and Ph.D. in computer science from Yonsei University, Korea in 1994, 1996, and 2000 respectively. Since 2003, he has been a professor at Computer Engineering Department of Inha University, Korea. His interests include network security, usable security, biometrics and counting theory.