PassGrid: Towards Graph-Supplemented Textual Shoulder Surfing Resistant Authentication

Teng Zhou[1], Liang Liu[1(✉)], Haifeng Wang[2], Wenjuan Li[3], and Chong Jiang[4]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
liangliu@nuaa.edu.cn

[2] NARI Technology Co., Ltd. State Key Laboratory of Smart Grid Protection and Control in China, Nanjing, China

[3] Department of Computer Science, City University of Hong Kong, Hong Kong S.A.R., China
wenjuan.li@my.cityu.edu.hk

[4] School of Computer Science, Guangzhou University, Guangzhou, China

Abstract. With the rapid development of intelligent mobile devices and network applications, user authentication plays an important role to help protect people's privacy and sensitive information. A large number of authentication textual and graphical schemes have been proposed in the literature, but the majority of them are vulnerable to shoulder surfing attacks, or have to sacrifice usability. Motivated by this challenge, we propose a graph-supplemented textual shoulder surfing resistant authentication system, called PassGrid. With a series of one-time login indicators and cyclic movable blocks with textual elements, PassGrid prevents attackers from guessing the passwords even with the help of a camera. To reduce users' workload, they only have to memorize one set of the password. Our user study shows that PassGrid can achieve good performance regarding security and usability, i.e., average login time consumption of 22s with a small password length.

Keywords: Graphical authentication · Textual password · Shoulder surfing resistant · User authentication · Security and usability

1 Introduction

With the rapid development of mobile devices, more applications are associated with users' private information like online transaction, which may allow attackers to gain unauthorized access to the device. Shoulder surfing is a kind of intrusion that attackers watch over a user's shoulder to gain the information. There are two ways to launch shoulder surfing attack [16–18,25]: (1) attacker users naked eyes, or (2) attacker uses camera-based device. In general, shoulder surfing attack tends to take place in the crowd, users will not easily figure out the attackers

W. Meng and S. Furnell (Eds.): SocialSec 2019, CCIS 1095, pp. 251–263, 2019.
https://doi.org/10.1007/978-981-15-0758-8_19

due to chaotic environment. Some attackers will capture videos that record the authentication scenes one or more times, and they are able to analyze users' behavior and then crack the password.

Texture password is the most widely used method during authentication [24]. People prefer to setting their passwords with ordinary words, phrases and symbols, making them easy to memorize. However texture password is too fragile to resist shoulder surfing attack. Users often input their password by pressing the buttons on the screen directly, which makes attackers to recover their passwords easily by shoulder surfing. In order to resist shoulder surfing attack, researchers proposed various graphic password schemes [3–7]. Graphic password consists of a series graphs and pictures, users need to select some graphs or pixels and upload them as passwords. A random indicator is always used with the graphic password to make passwords hard to be cracked. Though graphic password shows great advantages of resisting shoulder surfing attack, most applications authenticate users by texture passwords and most of the users still like to use texture passwords due to their habits. And only users in the crowd or insecure environments will be more likely to suffer from shoulder surfing attack, they can still log into the system by utilizing texture password in private occasions. Thus it is a trouble to memorize two completely different set of passwords for users. The ideal scheme is utilizing a new system based on texture password, learning from graphic password scheme to resist shoulder surfing attack.

In this paper, we focus on shoulder surfing attacks and propose a hybrid authentication system called *PassGrid*, which combines the features from both textual and graphical password authentication. PassGrid can provide one-time indicators and cyclic movable blocks with textual elements. Based on the random indicators, users have to move the blocks to the proper location with the correct inputting sequence. Our contributions can be summarized as below:

- We design a graph-supplemented textual shoulder surfing resistant authentication system named *PassGrid*, which can provide good usability and security with a small password length.
- We implemented a prototype of *PassGrid* and conducted a user study to evaluate the performance of *PassGrid* in the aspects of security and usability.

This paper is organized as follows: Sect. 2 provides the related studies regarding graphical password-based authentication system. Section 3 introduces our proposed system, and Sect. 4 conducts a user study to evaluate our scheme. Section 5 discuss some challenges and Sect. 6 concludes our work.

2 Background and Related Works

Textural password based authentication scheme was firstly introduced in the 1960s. It has become the most common authentication scheme nowadays. Textural passwords that comprised of numbers, upper- and lower-case letters and tokens are considered strong enough to resist brute attacks. However, complicate textural password is hard to memorize and still unable to restrict shoulder surfing attack.

From long ago, due to the limited graphic support of handheld devices, it is very difficult to design a good graphic authentication system. Jermyn et al. proposed an authentication system called Draw-A-Secret (DAS) [1]. This system makes users be able to draw a secret shape on the grids as a password. It will record the coordinates that the shape occupied. Users should draw the shape almost the same as stored one to pass authentication. Blonder [2] designed a graphic password scheme, the scheme generates password by clicking on the several locations on an image. During authentication, the user clicks the tolerant location where the user set up. It claims that the image can assist users to recall the password. Thus this schema may be more convenient than texture passwords. PassPoint system is proposed by Wiedenbecket et al. [3] Such password consists of a sequence of PassPoints on one image. The users should select some pixels and touch the screen to create their password. To log into the system, they have to select the pixels they picked before and click them in a tolerant area and in a correct sequence.

Most existing textural and graphic password schemes above are still vulnerable to shoulder surfing attacks. To address this issue, Sobrado and Birget developed a graphic password technique [4]. The system displays a number of 3 pass-objects that pre-selected by users among several other objects. The user has to click inside the triangle formed by the 3 pass-objects when they are authenticated. The Spy-Resistant Keyboard, a novel interface that allows users to enter private text without revealing it to an observer. The keyboard looks like an on-screen keyboard. A user study has been conducted, based on the study, users require more time to enter the password but they are prevented from observation attack [20].

PassBYOP is a new graphic authentication system, in which user presents the image to a system camera and then enter their password as a sequence of selections on live video of the token. Highly distinctive optical features are extracted from these selections and used as the password. They present three feasibility studies of PassBYOP examining its reliability, usability, and security about shoulder surfing attack [19]. Hung-Min Sun et al. proposed a shoulder surfing resistant authentication system named PassMatrix by using special graphic password [13]. In their system, there are random indicators and two circulative bars. It divides a picture into 7×11 parts and users choose one part as a digit of the password. During authentication, users will get an indicator, and then they need to align the corresponding letters on the horizontal bar and the vertical bar with the selected part of the picture. Some other relevant graphical passwords can refer to [5–12, 22].

3 Our Proposed Scheme

Textual password is the most widely used authentication method, but is too fragile to resist shoulder surfing attack. Many graphical password schemes were also proposed in the literature [1–4, 13]. However, the existing authentication systems often have the following problems: (1) most textual and graphical password

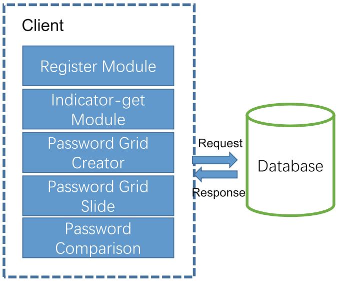
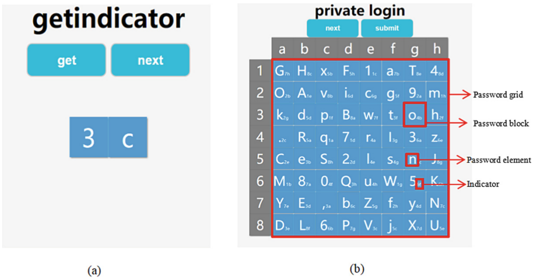


Fig. 1. The high-level architecture of PassGrid.



3 c

	a	b	c	d	e	f	g	h
1	G_{7h}	H_{8c}	x_{5b}	F_{5h}	1_{1c}	a_{7b}	T_{8e}	4_{8d}
2	O_{2b}	A_{1e}	v_{8b}	i_{6d}	c_{6g}	g_{5f}	9_{2a}	m_{1h}
3	k_{2g}	d_{5d}	p_{1f}	B_{8a}	w_{7f}	t_{3f}	o_{4b}	h_{2f}
4	$._{2c}$	R_{5a}	q_{1a}	7_{1d}	r_{4a}	l_{3g}	3_{6a}	z_{6e}
5	C_{2e}	e_{3b}	S_{8h}	2_{2d}	l_{4e}	s_{4g}	n	J_{8g}
6	M_{1b}	8_{7a}	0_{4f}	Q_{3h}	u_{4h}	W_{1g}	5_{6f}	K
7	Y_{7e}	E_{3d}	$,_{3a}$	b_{6c}	Z_{5g}	f_{2h}	y_{4d}	N_{7c}
8	D_{3e}	L_{8f}	6_{6b}	P_{7g}	V_{3c}	j_{5c}	X_{7d}	U_{5e}

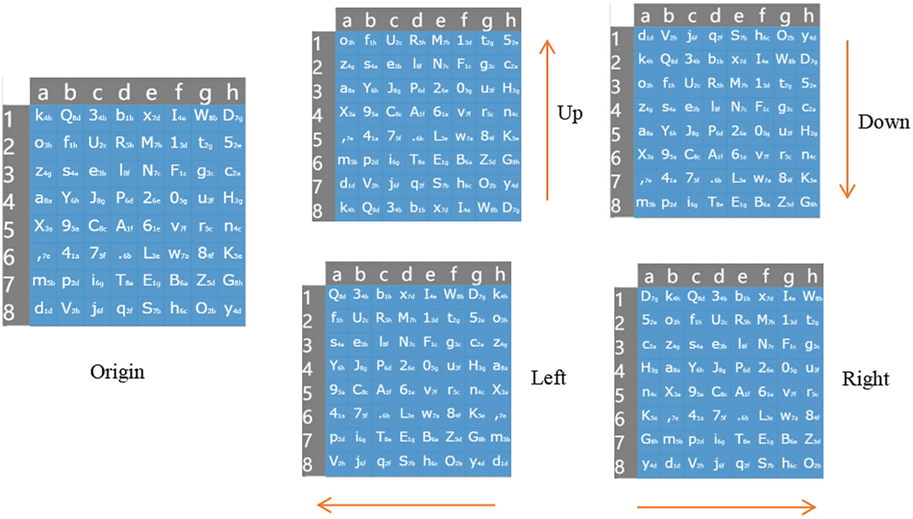
Fig. 2. (a) Indicator-get Module (b) Password grid creator Module.

schemes are vulnerable to shoulder surfing attacks; (2) some password schemes are too complicated to reduce the usability; and (3) some schemes require users to memorize too much information or steps. In this work, our goal is to design a graph-supplemented textual shoulder surfing resistant authentication system without reducing the usability.

3.1 Architecture

The architecture of our system is shown in Fig. 1, which consists of the following components:

- *Register module.* Users should have an account with user name and password in text strings.
- *Indicator-get module.* The system creates the coordinate of the first character of the password (As shown in Fig. 2(a)). In particular, it uses the number of 1–8 as a horizontal coordinate, and the letter of *a*–*h* as a vertical coordinate. The horizontal and vertical coordinates will be generated randomly. To get an indicator, we adopt the methods proposed in [13]. There are two ways: the indicator could be shown on the display directly, or can be delivered by an audio signal through the ear buds or Bluetooth.
- *Password grid creator module.* As shown in Fig. 2(b), we create a $M \times N$ matrix and select all printable characters as password elements. In addition, we assume the password grid is divided into a 8×8 matrix including numbers from 0 to 9, lowercase letters from *a* to *z*, uppercase letters from *A* to *Z*, and two token , and . as valid password elements. There is a random textual indicator in the lower right of every block, representing the next password character location.
- *Password grid slide module.* As shown in Fig. 3, users can drag each block by their finger in four directions and the whole password grid can move cyclic in rows and columns.
- *Password comparison module.* In this module, our system collects all textual elements that users input and constructs a password sequence. Then the system (or a server) can make a comparison to verify the password.



Origin

	a	b	c	d	e	f	g	h
1	k$_{4h}$	Q$_{8d}$	3$_{4b}$	b$_{1b}$	x$_{7d}$	I$_{4e}$	W$_{8b}$	D$_{7g}$
2	o$_{3h}$	f$_{1h}$	U$_{2c}$	R$_{5h}$	M$_{7h}$	1$_{3d}$	t$_{2g}$	5$_{2e}$
3	z$_{4g}$	s$_{4a}$	e$_{3b}$	l$_{8f}$	N$_{7c}$	F$_{1c}$	g$_{3c}$	c$_{2a}$
4	a$_{8a}$	Y$_{6h}$	J$_{8g}$	P$_{6d}$	2$_{6e}$	0$_{5g}$	u$_{3f}$	H$_{3g}$
5	X$_{3a}$	9$_{5a}$	C$_{8c}$	A$_{1f}$	6$_{1e}$	v$_{7f}$	r$_{5c}$	n$_{4c}$
6	,$_{7e}$	4$_{1a}$	7$_{5f}$	.$_{6b}$	L$_{3e}$	w$_{7a}$	8$_{4f}$	K$_{5e}$
7	m$_{5b}$	p$_{2d}$	i$_{6g}$	T$_{8e}$	E$_{1g}$	B$_{6a}$	Z$_{5d}$	G$_{8h}$
8	d$_{1d}$	V$_{2h}$	j$_{6f}$	q$_{2f}$	S$_{7b}$	h$_{6c}$	O$_{2b}$	y$_{4d}$

Up

	a	b	c	d	e	f	g	h
1	o$_{3h}$	f$_{1h}$	U$_{2c}$	R$_{5h}$	M$_{7h}$	1$_{3d}$	t$_{2g}$	5$_{2e}$
2	z$_{4g}$	s$_{4a}$	e$_{3b}$	l$_{8f}$	N$_{7c}$	F$_{1c}$	g$_{3c}$	c$_{2a}$
3	a$_{8a}$	Y$_{6h}$	J$_{8g}$	P$_{6d}$	2$_{6e}$	0$_{5g}$	u$_{3f}$	H$_{3g}$
4	X$_{3a}$	9$_{5a}$	C$_{8c}$	A$_{1f}$	6$_{1e}$	v$_{7f}$	r$_{5c}$	n$_{4c}$
5	,$_{7e}$	4$_{1a}$	7$_{5f}$	.$_{6b}$	L$_{3e}$	w$_{7a}$	8$_{4f}$	K$_{5e}$
6	m$_{5b}$	p$_{2d}$	i$_{6g}$	T$_{8e}$	E$_{1g}$	B$_{6a}$	Z$_{5d}$	G$_{8h}$
7	d$_{1d}$	V$_{2h}$	j$_{6f}$	q$_{2f}$	S$_{7b}$	h$_{6c}$	O$_{2b}$	y$_{4d}$
8	k$_{4h}$	Q$_{8d}$	3$_{4b}$	b$_{1b}$	x$_{7d}$	I$_{4e}$	W$_{8b}$	D$_{7g}$

Down

	a	b	c	d	e	f	g	h
1	d$_{1d}$	V$_{2h}$	j$_{6f}$	q$_{2f}$	S$_{7b}$	h$_{6c}$	O$_{2b}$	y$_{4d}$
2	k$_{4h}$	Q$_{8d}$	3$_{4b}$	b$_{1b}$	x$_{7d}$	I$_{4e}$	W$_{8b}$	D$_{7g}$
3	o$_{3h}$	f$_{1h}$	U$_{2c}$	R$_{5h}$	M$_{7h}$	1$_{3d}$	t$_{2g}$	5$_{2e}$
4	z$_{4g}$	s$_{4a}$	e$_{3b}$	l$_{8f}$	N$_{7c}$	F$_{1c}$	g$_{3c}$	c$_{2a}$
5	a$_{8a}$	Y$_{6h}$	J$_{8g}$	P$_{6d}$	2$_{6e}$	0$_{5g}$	u$_{3f}$	H$_{3g}$
6	X$_{3a}$	9$_{5a}$	C$_{8c}$	A$_{1f}$	6$_{1e}$	v$_{7f}$	r$_{5c}$	n$_{4c}$
7	,$_{7e}$	4$_{1a}$	7$_{5f}$	.$_{6b}$	L$_{3e}$	w$_{7a}$	8$_{4f}$	K$_{5e}$
8	m$_{5b}$	p$_{2d}$	i$_{6g}$	T$_{8e}$	E$_{1g}$	B$_{6a}$	Z$_{5d}$	G$_{8h}$

Left

	a	b	c	d	e	f	g	h
1	Q$_{8d}$	3$_{4b}$	b$_{1b}$	x$_{7d}$	I$_{4e}$	W$_{8b}$	D$_{7g}$	k$_{4h}$
2	f$_{1h}$	U$_{2c}$	R$_{5h}$	M$_{7h}$	1$_{3d}$	t$_{2g}$	5$_{2e}$	o$_{3h}$
3	s$_{4a}$	e$_{3b}$	l$_{8f}$	N$_{7c}$	F$_{1c}$	g$_{3c}$	c$_{2a}$	z$_{4g}$
4	Y$_{6h}$	J$_{8g}$	P$_{6d}$	2$_{6e}$	0$_{5g}$	u$_{3f}$	H$_{3g}$	a$_{8a}$
5	9$_{5a}$	C$_{8c}$	A$_{1f}$	6$_{1e}$	v$_{7f}$	r$_{5c}$	n$_{4c}$	X$_{3a}$
6	4$_{1a}$	7$_{5f}$	.$_{6b}$	L$_{3e}$	w$_{7a}$	8$_{4f}$	K$_{5e}$	,$_{7e}$
7	p$_{2d}$	i$_{6g}$	T$_{8e}$	E$_{1g}$	B$_{6a}$	Z$_{5d}$	G$_{8h}$	m$_{5b}$
8	V$_{2h}$	j$_{6f}$	q$_{2f}$	S$_{7b}$	h$_{6c}$	O$_{2b}$	y$_{4d}$	d$_{1d}$

Right

	a	b	c	d	e	f	g	h
1	D$_{7g}$	k$_{4h}$	Q$_{8d}$	3$_{4b}$	b$_{1b}$	x$_{7d}$	I$_{4e}$	W$_{8b}$
2	5$_{2e}$	o$_{3h}$	f$_{1h}$	U$_{2c}$	R$_{5h}$	M$_{7h}$	1$_{3d}$	t$_{2g}$
3	c$_{2a}$	z$_{4g}$	s$_{4a}$	e$_{3b}$	l$_{8f}$	N$_{7c}$	F$_{1c}$	g$_{3c}$
4	H$_{3g}$	a$_{8a}$	Y$_{6h}$	J$_{8g}$	P$_{6d}$	2$_{6e}$	0$_{5g}$	u$_{3f}$
5	n$_{4c}$	X$_{3a}$	9$_{5a}$	C$_{8c}$	A$_{1f}$	6$_{1e}$	v$_{7f}$	r$_{5c}$
6	K$_{5e}$	,$_{7e}$	4$_{1a}$	7$_{5f}$	.$_{6b}$	L$_{3e}$	w$_{7a}$	8$_{4f}$
7	G$_{8h}$	m$_{5b}$	p$_{2d}$	i$_{6g}$	T$_{8e}$	E$_{1g}$	B$_{6a}$	Z$_{5d}$
8	y$_{4d}$	d$_{1d}$	V$_{2h}$	j$_{6f}$	q$_{2f}$	S$_{7b}$	h$_{6c}$	O$_{2b}$

Fig. 3. Password grid slide module.

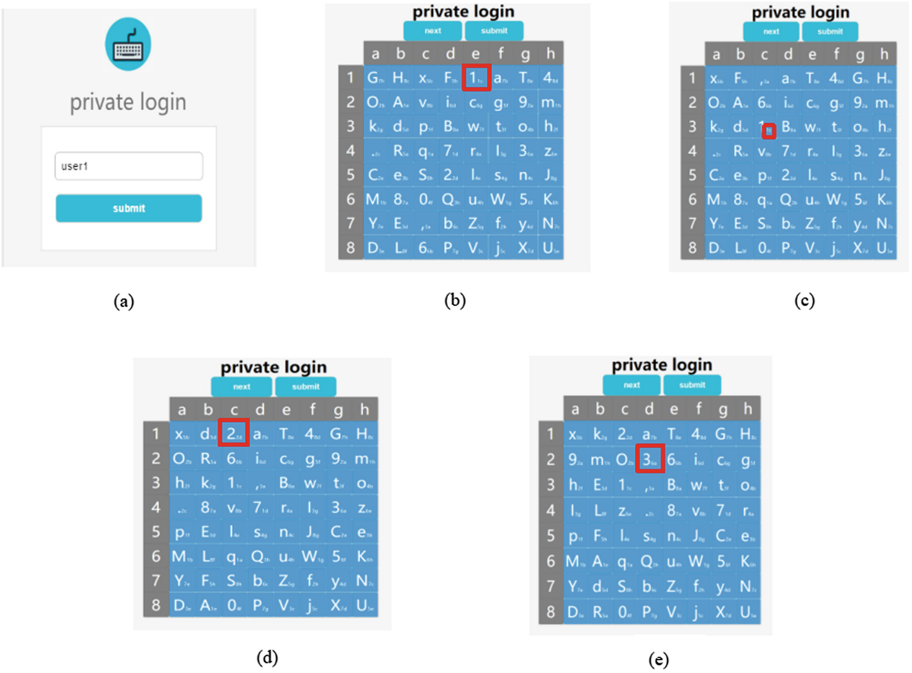
- *Database.* It mainly contains a *user table* that records usernames and passwords, and a *LoginData table* that records the login data for all users.

In this work, we mainly consider two conditions: private and public environment. For the former, users can log into the system by inputting texture password directly. For the latter, as shown in Fig. 2(a), users have to receive an indicator that represents the location of the first character of the password before entering texture password. The indicator is a 2D coordinate like $(3, c)$. When a user performs password verification, as shown in Fig. 2(b), a password grid sequence will be generated according to the password length. Each password character corresponds to a password grid, which is composed of several password blocks. A password element is displayed in the middle and the indicator is displayed in the lower right corner.

3.2 PassGrid Design

As mentioned earlier, based on different environments, our scheme considers two conditions: a Normal Login phase and a Private Login phase.

- Normal Login phase. If the user is authenticated in a private environment, he or she just needs to input username and password directly.
- Private Login phase. If the user is authenticated in a public place, then the detailed procedures are shown below:
  - Step 1: as shown in Fig. 4(a), the user has to first input the username to verify whether there is valid user.
  - Step 2: the index of a password character in a password is denoted by i, which is initialized as 1.



private login

user1

(a)

private login

	a	b	c	d	e	f	g	h
1	G_{7h}	H_{8c}	x_{5b}	F_{5h}	1_{1c}	a_{7b}	T_{8e}	4_{8d}
2	O_{2b}	A_{1e}	v_{8b}	i_{6d}	c_{6g}	g_{5f}	9_{2a}	m_{1h}
3	k_{2g}	d_{5d}	p_{1f}	B_{8a}	w_{7f}	t_{3f}	o_{4b}	h_{2f}
4	$._{2c}$	R_{5a}	q_{1a}	7_{1d}	r_{4a}	l_{3g}	3_{6a}	z_{6e}
5	C_{2e}	e_{3b}	S_{8h}	2_{2d}	I_{4e}	s_{4g}	n_{4c}	J_{8g}
6	M_{1b}	8_{7a}	0_{4f}	Q_{3h}	u_{4h}	W_{1g}	5_{6f}	K_{6h}
7	Y_{7e}	E_{3d}	$,_{3a}$	b_{6c}	Z_{5g}	f_{2h}	y_{4d}	N_{7c}
8	D_{3e}	L_{8f}	6_{6b}	P_{7g}	V_{3c}	j_{5c}	X_{7d}	U_{5e}

(b)

private login

	a	b	c	d	e	f	g	h
1	x_{5b}	F_{5h}	$,_{3a}$	a_{7b}	T_{8e}	4_{8d}	G_{7h}	H_{8c}
2	O_{2b}	A_{1e}	6_{6b}	i_{6d}	c_{6g}	g_{5f}	9_{2a}	m_{1h}
3	k_{2g}	d_{5d}	1_{1c}	B_{8a}	w_{7f}	t_{3f}	o_{4b}	h_{2f}
4	$._{2c}$	R_{5a}	v_{8b}	7_{1d}	r_{4a}	l_{3g}	3_{6a}	z_{6e}
5	C_{2e}	e_{3b}	p_{1f}	2_{2d}	I_{4e}	s_{4g}	n_{4c}	J_{8g}
6	M_{1b}	8_{7a}	q_{1a}	Q_{3h}	u_{4h}	W_{1g}	5_{6f}	K_{6h}
7	Y_{7e}	E_{3d}	S_{8h}	b_{6c}	Z_{5g}	f_{2h}	y_{4d}	N_{7c}
8	D_{3e}	L_{8f}	0_{4f}	P_{7g}	V_{3c}	j_{5c}	X_{7d}	U_{5e}

(c)

private login

	a	b	c	d	e	f	g	h
1	x_{5b}	d_{5d}	2_{2d}	a_{7b}	T_{8e}	4_{8d}	G_{7h}	H_{8c}
2	O_{2b}	R_{5a}	6_{6b}	i_{6d}	c_{6g}	g_{5f}	9_{2a}	m_{1h}
3	h_{2f}	k_{2g}	1_{1c}	$,_{3a}$	B_{8a}	w_{7f}	t_{3f}	o_{4b}
4	$._{2c}$	8_{7a}	v_{8b}	7_{1d}	r_{4a}	l_{3g}	3_{6a}	z_{6e}
5	p_{1f}	E_{3d}	I_{4e}	s_{4g}	n_{4c}	J_{8g}	C_{2e}	e_{3b}
6	M_{1b}	L_{8f}	q_{1a}	Q_{3h}	u_{4h}	W_{1g}	5_{6f}	K_{6h}
7	Y_{7e}	F_{5h}	S_{8h}	b_{6c}	Z_{5g}	f_{2h}	y_{4d}	N_{7c}
8	D_{3e}	A_{1e}	0_{4f}	P_{7g}	V_{3c}	j_{5c}	X_{7d}	U_{5e}

(d)

private login

	a	b	c	d	e	f	g	h
1	x_{5b}	k_{2g}	2_{2d}	a_{7b}	T_{8e}	4_{8d}	G_{7h}	H_{8c}
2	9_{2a}	m_{1h}	O_{2b}	3_{6a}	6_{6b}	i_{6d}	c_{6g}	g_{5f}
3	h_{2f}	E_{3d}	1_{1c}	$,_{3a}$	B_{8a}	w_{7f}	t_{3f}	o_{4b}
4	l_{3g}	L_{8f}	z_{6e}	$._{2c}$	8_{7a}	v_{8b}	7_{1d}	r_{4a}
5	p_{1f}	F_{5h}	I_{4e}	s_{4g}	n_{4c}	J_{8g}	C_{2e}	e_{3b}
6	M_{1b}	A_{1e}	q_{1a}	Q_{3h}	u_{4h}	W_{1g}	5_{6f}	K_{6h}
7	Y_{7e}	d_{5d}	S_{8h}	b_{6c}	Z_{5g}	f_{2h}	y_{4d}	N_{7c}
8	D_{3e}	R_{5a}	0_{4f}	P_{7g}	V_{3c}	j_{5c}	X_{7d}	U_{5e}

(e)

Fig. 4. Private login phase

- Step 3: as shown in Fig. 2(a), the user then presses the *'get' button* to obtain a random indicator *ind*. The user should memorize the indicator and then move to the next step.
- Step 4: the system creates a password grid and the user moves the block containing the *ith* password character to the location, where the indicator *ind* points to. The user needs to memorize the indicator *nextInd* in the lower right corner of the block containing the *ith* password character, and then presses the *'next' button*. After that, *i:=i+1*, *ind := nextInd*.
- Step 5: the user repeats step 4 until he/she acknowledges he/she has finished inputting the password. Then the user presses the *'submit' button*.
- Step 6: password sequence *pwd* will be generated by password comparison module, and then the system or server will compare *pwd* with the correct password pwd'. If successful, the user passes the authentication; otherwise, the authentication process fails.

The password grid-based authentication is summarized in Algorithm 1. As an example, we assume that a user sets 'user1' as the account name and '123' as the password. In a public environment, the user has to firstly input the username and obtain the indicator (see Fig. 2(a)) in the Indicator-get module, which is '3c'. The next step is to move the textual password block with the number of '1' to the location of $(3, c)$ (see Fig. 4(b)). Then the user should press the 'next' button

Algorithm 1. Password grid-based authentication

```
Initialization:
  1. Password sequence users input is pwdlist and password sequence users
pre-defined is pwd. Initial pwdlist as NULL.
  2. Generate $L$ password grid matrixs $\{MA_1, MA_2, \ldots, MA_L\}$, where $L$ is the
length of pwd. In each matrix, there are $MN$ password elements $MA_i[j][k]$ and
indicators $ind_i[j][k]$. $i \in \{1,2,\ldots,L\}$, $j \in \{1,2,\ldots,M\}$ , $k \in \{1,2,\ldots,N\}$
  3. The current indicator is denoted by ind, which is initialized as NULL.
get the first indicator ind1, and ind = ind1
while Ture do
    generate a new password grid
    look for the location ind points to and move the block with element pwd[i]
    to the location. Get the password element $MA_i[j][k]$ from the block in $MA_i$
    and append $MA_i[j][k]$ into pwdlist, and $ind = ind_i[j][k]$, $i = i+1$
    if the user acknowledges he/she has finished inputting password and chooses
    to submit then
        break;
generate pwdlist as a string
if pwdlist is equal to pwd then
    the user passes the authentication
else
    the user fails to pass the authentication
```

to continue. As shown in Fig. 4(c), the indicator of the next password character is displayed in the lower right of the previous texture block. It is worth noting that as indicators will be recreated in every round. Then the user should drag the right password block to the proper location and repeat previous step for the next password (as shown in Fig. 4(d) and Fig. 4(e)). When the user completes the whole password, he or she can press the submit button and wait for verification.

4 User Study

We designed a user study that requires participants to register three different accounts with three different passwords, and they need to login about five times in each account. We set the length of the passwords: a short one (1–3 digits), a medium one (4–6 digits) and a long one (above 6 digits). We adopted two common metrics to help evaluate the performance of PassGrid in the aspects of both accuracy and usability.

- Accuracy. In the study, we investigate whether participates can log into the system successfully with different length of passwords. To analyze the accuracy, each participant was required to register three different accounts.
- Usability. In this study, we measured the system usability by recording time consumption by all participants on different length of passwords.

Participants. We recruited 15 participants (M = 23 years; SD = 3; 6 females), who are volunteers and have an interest in our study. All participants are graduate students from different disciplines such as Mathematics, Computer Science and so on. After an informal interview, we found all participants are common smartphone users, but were unfamiliar with graphical authentication system. This work received an approval from the Department and all participants have to make an agreement before they started.

Study phases. In the study, we have three main phases:

- *Introduction phase.* We explained the basic idea and our goals to all participants. We also provided a presentation on how to use the prototype system and answered any questions from participants.
- *Practice phase.* Participants were required to register three different accounts with different password lengths. Before the study, they could have several trials until they believe that they have been familiar with the system.
- *Login phase.* After practice, participants were requested to log into the system five times for each account. The login phase would be repeated once, that is, there are two rounds of login trials.

Accuracy. We recorded the login information after each of the Private Login phase. We defined the *Total Accuracy* as below:

$$Total\ Accuracy = \frac{Successful\ Times}{Total\ Times} \tag{1}$$

Figure 5 shows the total accuracy in the login session with three different password lengths. It is found that when users utilized a short password, they could achieve a higher successful rate like 96% and 98% in two rounds. Intuitively, it is easily understand that all participants were more proficient in the second round. In addition, the successful rate was found to decrease with the increase of password length. While participants could improve the rate in the second round, i.e., from 76% to 86%, and from 72% to 80%. We particularly surveyed participants to investigate why the successful rate would drop with a longer password length. In addition to memory limitation, we also found that participants might conduct many errors, i.e., accidentally moving the block to the wrong place, or pressing the next button carelessly. These errors could be avoided when they are more careful during the authentication process.

Usability. As mentioned above, we evaluate the usability of PassGrid by computing the time consumption for all participants in each round, those who could successfully pass the private authentication phase.

- It is found that participants spent less time handling the passwords with a smaller length. With the increase of length, participants have to spend more time in authentication. For example, the average time consumption is around 24.4 s for the password length with 3 letters or less. The time consumption could increase to over 82 s for the middle-length and long-length passwords.

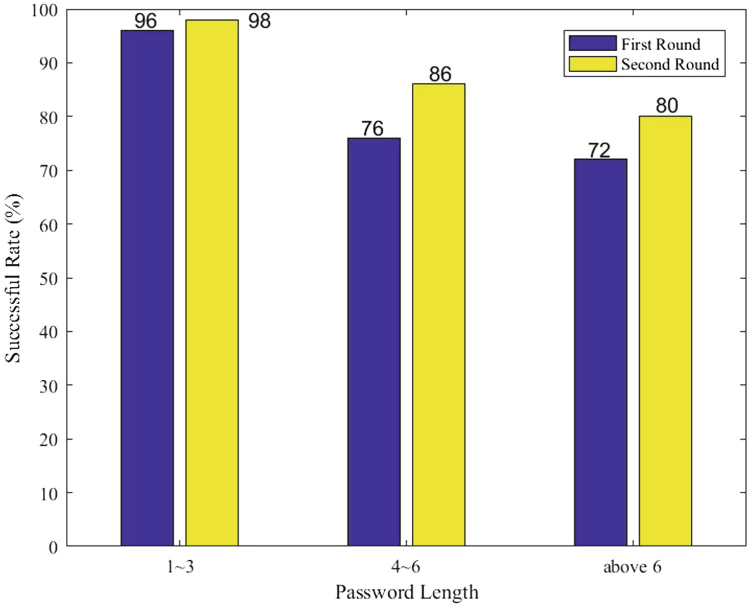


Fig. 5. Successful rate of different passwords' length in two rounds

- In addition, participants can perform better in the second round, where the average time consumption is around 22 s for the short password length, and over 74.2 s for the middle-length and long-length passwords. This observation validates that PassGrid can provide good usability when participants get familiar with the system, i.e., after several practice trials.

Limitation Discussion. In our informal interview with all participants, we found that most of them positively rated our system to be acceptable and usable in practice, by considering the enhanced security provided by PassGrid, i.e., against shoulder surfing attacks. However, as our work is at an early stage, we acknowledge there existing many issues and challenges.

- Based on the data, it is observed that PassGrid could reach good performance in the aspects of accuracy and usability, when the password length is small, i.e., 22 s for the short password length. As a reference, PassMatrix [13] provided an average time consumption of 30 s. However, with a larger password length, the time consumption of PassGrid could keep increasing. This is actually an open issue for most shoulder surfing resistant schemes including PassMatrix. We acknowledge this is the major limitation of PassGrid, and it is one of our future work to investigate how to reduce the time consumption.
- As an ongoing research study, we only involved 15 participants in the evaluation part, and all of them are graduate students. We acknowledge that more participants with diverse background are desirable in practice. We plan to

involve more than 50 participants in our future work and validate the results obtained in this work.
- In this work, we only analyze the security of PassGrid against several typical attacks. We acknowledge that more efforts should be provided to explore the practical performance. One of our future directions is to perform a shoulder surfing attack or guessing attack with the attempt to compromise our system. Each participant is expected to have many trials, like 10 times [13], to crack our system. We believe that this study can provide statistical data and validate the performance.

5 Security Discussion

In this section, we mainly discuss the security of PassGrid against three typical attacks: brute force attack, shoulder surfing attack, and smudge attack.

5.1 Brute Force Attack

To perform the brute force attack, intruders have to enumerate all possible passwords. We assume that users set the password in n characters. Considering one character from a password, we divided one password grid into a $M \times N$ matrix, which represents the password space is *(*$M \times N$*)*. On the other hand, the possibility that attackers can crack one character is *1/(*$M \times N$*)*. We used location indicators to increase the randomness of PassGrid. This means only when attackers can input the right former password sequence, they can get the next right character of the password. Thus, the probability of cracking the whole authentication scheme is *1/(*$M \times N)^{2n}$. In our prototype, we assume $M = 8$ and $N = 8$, then our system can reach the above possibility of $(0.024\%)^n$. We believe it is should be sufficient to defend against brute force attacks.

5.2 Shoulder Surfing Attack

Most of the textural or graphical password authentication systems cannot succeed in resisting shoulder surfing attacks, which have been a real threat and challenge. In our scheme, there is no need for users to click the button or input the password on the screen directly. They only need to drag the textural blocks and ensure the password character being moved to the correct location.

The password spaces of other schemes such as those in CAPTCHA-based method [21], Pass-icons [23] and Color-rings [15] can be narrowed down by camera-based shoulder surfing attacks. While in our system, the order of password elements on each password grid is totally distinct, i.e., the first location indicator is given randomly to ensure that users can obtain it in a private way. The indicator in the lower right of each block is also randomly recreated in every round. That is, even if an attacker can capture users' authentication phase by using camera or other recording devices, it is still hard to determine the correct blocks which the user moves in each round.

5.3 Smudge Attack

When users interact with phone screens, they will leave oily residues, or smudges. This attack allows intruders to examine *smudge* on the phone screen after users inputting their sensitive information [14]. In our scheme, the smudge can be any shape, e.g., irregular, and not easy to identify, since the password grid is cyclic in rows and columns, and users can drag any block many times. In this case, it is very difficult for attackers to extract useful information based on the complicated and vague smudges.

6 Conclusion

For most current authentication schemes, shoulder surfing attack is one of the major threats, where attackers can compromise the authentication process by watching over people's shoulder or recording videos through some camera-based devices. The use of graphical passwords can alleviate shoulder surfing attack to some degree, whereas people are accustomed to textual password and cannot memorize completely distinct password sequences.

To overcome this challenge, we propose a graph-supplemented textual shoulder surfing resistant authentication system, called PassGrid, which only requires users to set textual password, based on randomly generated indicators and cyclic movable texture blocks. Users have to move password blocks to the right location according to the location indicator instead of touching the screen or inputting directly, which can effectively prevent shoulder surfing attack. In the evaluation, we conducted a users study to investigate the performance of PassGrid. Our results indicate that PassGrid can provide good security and usability when the password length is small, while the usability would be degraded with the increase of password length. In future work, we plan to investigate how to simplify the authentication process and decrease the login time. In addition, we plan to compare our system with other related schemes.

References

1. Jermyn, I., Mayer, A., Monrose, F., Reiter, M., Rubin, A.: The design and analysis of graphical passwords. In: Proceedings of the 8th Conference USENIX Security Symposium, vol. 8 (1999)
2. Blonder, G.E.: Graphical passwords. United States Patent 5559961 (1996)
3. Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., Memon, N.: Passpoints: design and longitudinal evaluation of a graphical password system. Int. J. Hum.-Comput. Stud. **63**(1–2), 102–127 (2005)
4. Sobrado, L., Birget, J.C.: Graphical passwords. The Rutgers Scholar, An Electronic Bulletin for Undergraduate Research, vol. 4 (2002)
5. Meng, Y., Li, W., Kwok, L.-F.: Enhancing click-draw based graphical passwords using multi-touch on mobile phones. In: Janczewski, L.J., Wolfe, H.B., Shenoi, S. (eds.) SEC 2013. IAICT, vol. 405, pp. 55–68. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39218-4_5

6. Meng, W.: RouteMap: a route and map based graphical password scheme for better multiple password memory. Network and System Security. LNCS, vol. 9408, pp. 147–161. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25645-0_10
7. Meng, W.: Evaluating the effect of multi-touch behaviours on android unlock patterns. Inf. Comput. Secur. **24**(3), 277–287 (2016)
8. Meng, W., Li, W., Wong, D.S., Zhou, J.: TMGuard: a touch movement-based security mechanism for screen unlock patterns on smartphones. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 629–647. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_34
9. Meng, W., Lee, W.H., Au, M.H., Liu, Z.: Exploring effect of location number on map-based graphical password authentication. In: Pieprzyk, J., Suriadi, S. (eds.) ACISP 2017. LNCS, vol. 10343, pp. 301–313. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59870-3_17
10. Meng, W., Li, W., Kwok, L.F., Choo, K.K.R.: Towards enhancing click-draw based graphical passwords using multi-touch behaviours on smartphones. Comput. Secur. **65**, 213–229 (2017)
11. Meng, W., Fei, F., Jiang, L., Liu, Z., Su, C., Han, J.: CPMap: design of click-points map-based graphical password authentication. SEC **2018**, 18–32 (2018)
12. Meng, W., Liu, Z.: TMGMap: designing touch movement-based geographical password authentication on smartphones. In: Su, C., Kikuchi, H. (eds.) ISPEC 2018. LNCS, vol. 11125, pp. 373–390. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99807-7_23
13. Sun, H.-M., Chen, S.-T., Yeh, J.-H., Cheng, C.-Y.: Shoulder surfing resistant graphical authentication system. IEEE Trans. Dependable Secure Comput. **15**(2), 180–193 (2018)
14. Aviv, A., Gibson, K., Mossop, E., Blaze, M., Smith, J.: Smudge attacks on smartphone touch screens. In: Proceedings of USENIX 4th Workshop on Offensive Technologies (2010)
15. Zhao, H., Li, X.: S3pas: a scalable shoulder-surfing resistant textual-graphical password authentication scheme. In: Proceeding of the 21st International Conference Advances Information Network Applications Workshops, vol. 2, pp. 467–472 (2007)
16. Long, J., Mitnick, K.: No tech hacking: a guide to social engineering, dumpster diving, and shoulder surfing (2011). https://www.hackersforcharity.org/files/NTH_SAMPLE.pdf
17. Kwon, T., Shin, S., Na, S.: Covert attentional shoulder surfing: human adversaries are more powerful than expected. IEEE Trans. Syst. Man Cybern. Syst. **44**(6), 716–727 (2014)
18. Google glass snoopers can steal your passcode with a glance. http://www.wired.com/2014/06/google-glass-snoopers-can-steal-your-passcode-with-a-glance/
19. Bianchi, A., Oakley, I., Kim, H.S.: PassBYOP: bring your own picture for securing graphical passwords. IEEE Trans. Hum.-Mach. Syst. **46**(3), 2168–2291 (2015)
20. Tan, D., Keyani, P., Czerwinski, M.: Spy-resistant keyboard: towards more secure password entry on publicly observable touch screens. In: Proceedings of OZCHIComputer- Human Interaction Special Interest Group (CHISIG) of Australia, Canberra, Australia. ACM Press. Citeseer (2005)
21. Wang, L., Chang, X., Ren, Z., Gao, H., Liu, X., Aickelin, U.: Against spyware using captcha in graphical password scheme. In: Proceeding of the 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 760–767. IEEE (2010)
22. Yu, X., Wang, Z., Li, Y., Li, L., Zhu, W.T., Song, L.: EvoPass: evolvable graphical password against shoulder-surfing attacks. Comput. Secur. **70**, 179–198 (2017)

23. Wiedenbeck, S., Waters, J., Sobrado, L., Birget, J.-C.: Design and evaluation of a shoulder-surfing resistant graphical password scheme. In: Proceedings of the working conference on Advanced Visual Interfaces, ser. AVI 2006, pp. 177–184. ACM, New York (2006)
24. Suo, X., Zhu, Y., Owen, G.S.: Graphical passwords: a survey. In: Proceedings of the 21st Annual Computer Security Applications Conference, ACSAC, pp. 463–472. IEEE Computer Society, USA (2005)
25. Takada, T.: Fakepointer: an authentication scheme for improving security against peeping attacks using video cameras. In: Proceedings of the 2nd International Conference Mobile Ubiquitous Computer, System, Service Technology, pp. 395–400 (2008)