A Graphical Password Scheme against Snapshot, Remote Monitoring, And Shoulder-surfing with Its Application in One-Time Password

Chengxue Qian[1], Xingwei Song[1], Yun Huang[2], Xuejia Lai[2]

[1]Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{09310112152006 & 09310112152001}@sjtu.edu.cn, hycnsjtu@hotmail.com, lai-xj@cs.sjtu.edu.cn

Abstract - In this paper, we present a novel user-friendly graphical password scheme resistant against "watching" attacks. Snapshot, remote monitoring, and shoulder-surfing have in common that all these attacks act as if one could directly watch the users' behavior on the screen, resulting in an insecure use of alphanumeric passwords ("watching" attacks). New technology based on graphical passwords uses graphs as authentication media where the user identifies, reproduces, or interacts with graphs to prove his identity, which partly blocks the danger. However, current graphical passwords such as D-A-S, PassPoints, Passfaces TM, and the algorithms D. Hong and Sobrado, etc. proposed are either too complicated or ineffective against "watching" attacks. In our proposal, the authentication process uses familiar images that only true users can recognize. It is hard to fabricate even many previous authentication processes are totally exposed. Furthermore a detailed application in OTP, which basically establishes an extra OTP input encryption, is discussed and its security analysis is presented.

Index Terms – Graphical password, snapshot, shoulder-surfing, OTP.

I. Introduction

In this work we present a novel graphical password scheme against "watching" attacks (snapshot, remote monitoring, and shoulder-surfing). It takes advantage of a background sub-image pool and of a graphic or digit disc to realize the user authentication process. A detailed application in one-time password (OTP) is demonstrated and its security analysis is presented.

Trustworthy authentication has become a crucial component for mobile devices and by extension for any company or institution providing online services because the proliferation of mobile devices and wireless networks has fostered a stronger dependence on online social activities. There are constantly more and more people who expect to manage all their accounts, including some sensitive information, anywhere at any time. Traditional alphanumeric password user authentication has started to be replaced by graphical password alternatives, like textual passwords with graphical assistance or purely graphical password schemes.

E. G. Blonder first proposed a graphical password model in mid 1990s: the user is provided with a picture onto which he is required to click different regions in order to pass theauthentication [1]. In 1999, Jermyn introduced the DAS system whereas a year later, Dhamija and Perrig presented their graphical password model which contained three stages, namely, designation, training, and identification [2, 3].

Up to now, many graphical schemes have emerged and this research area has come to a flourishing age. However, current graphical password schemes are either too complicated for users, or too simple to escape the whole scope of "watching" attacks.

The remainder of this article is organized as follows: in Section 2, a survey of related work on graphical password schemes is presented together with a threat model of three kinds of "watching" attacks. In Section 3, the general idea and design of this graphical scheme are illustrated while Section 4 contains a specific application in OTP including a detailed algorithm and its security analysis. Finally, Section 5 discusses ideas for future work.

II. Background and Related Work

A. Threats and Attacks

One of the major drawbacks with current passwords, both alphanumeric and graphical, is the threat of "watching" attacks: snapshot, remote monitoring, channel interception and shoulder-surfing. All of them allow stealing the secrets and credentials by watching both virtually (screen capture, online monitoring over IP, information stream monitoring via channel intercepted, etc...) and in the real world as shoulder-surfing attacks.

1) "Watching" attacks: Snapshot, screen/mouse/keyboard capture, instantly freezes sensitive data input and causes information exposure. In the remote monitoring case the attacker maliciously implants some spying software in the PC and together with the server; he can gain access to the credentials by monitoring, comparison and analysis. Although several security precautions, such as installing and maintaining a firewall; keeping the antivirus database up-to-date; regularly applying security fixes; and configuring security software appropriately, can prevent such attacks, it is not an easy task for many users that prefer not to bother with those security concern and even at time simply ignore warnings given by the operation system or the web browser. Moreover it is difficult to prevent shoulder-surfing even though the password itself are hard to guess. A person who gets to observe a few login sessions could, depending on the scheme, eventually Fig. out

* This work is supported by National Natural Science Foundation of China (61073149 and 61272440); State Key Laboratory of ASIC & System (11KF0020); Key Lab of Information Network Security; Ministry of Public Security (C11603).

the password. Besides, the internal staff of a company or institution should be taken into consideration when securing a system. Although security policies and regulations edict rules for the employees, there is still a big risk that credentials and secrets get compromised at some stage. For instance one can imagine that during an online bank transfer data might be abused by internal bank staff. Another scenario could be external attackers that utilize a flaw in the security strategy from the internal staff to get access to private accounts. This is indeed what happened to the RSA SecurID (an OTP product) data leak accident in 2011 [4]. Long known for its good reputation among some big companies and government agencies this dual-factor authentication user token system lost part of its credibility when the system got breached due to a human error. The truth is that a low-level employee opened a phishing e-mail and thinking it was legitimate messages. Therefore internal users and staff should be seen as an integral part of the whole authentication system and as such must be taken into serious consideration.

B. *Graphical Password Defenses*

1) Draw a secret: "Draw a Secret'' (DAS) is a purely graphical password selection and input scheme as showed in Fig. 1 [2]. It is an authentication system widely used on PDA. In this scheme the password is a simple picture drawn on a grid through the initialization stage when a user first logs in. The system will record each stroke toward (across) the grid and in order to pass the authentication, a user must draw a picture similar to the initial one. However, study shows that the user's habits largely jeopardize its security. Jermyn pointed out that users are more inclined to choose simple patterns easy to guess and at a more central location. Surveys show that 86% of the users draw around the center; 45% of the patterns are symmetry; 29% of the patterns are invalid; and more than 80% of the patterns have no more than three strokes [2]. Therefore, many researchers have proposed improvements. Chalkias, for instance, put together grids of different size refocusing the attention of the user to area besides the central part whereas Dunphy proposes the presence of a background image that enables the user to draw a password of higher security [5, 6]. Nevertheless, the security of DAS largely depends on user behavior and is in the end quite vulnerable to "watching" attacks as the password image can be gained through mere observation.

2) PassPoint: PassPoints is an authentication system model proposed by Wiedenbeck where a user clicks on sequential pixels set beforehand to authenticate himself [7]. For example, N times' pixel-clicking in an 800x600 picture gives a password space of size 1200N. However, most users tend to choose easily identifiable points such as edges or centres of objects, which favours attackers using behaviour analysis to crack the password. For instance, Dirik set up models for "points of interest" analysis to simulate points that are more likely to be selected by the user [8]. Also, this scheme cannot survive "watching" attacks.

3) Passfaces TM: Developed by RealUser, PassfacesTM is another authentication system mainly used on PDA [9]. Using human face picture as authentication media, basically takes advantage of the fact that human tend to recognize faces much easier. While authenticating a user needs to click a pre-specified face pictures and such process repeats itself several times for security enhancement purpose. Although users may find the process quite straightforward, the system gives a very limited password space which is pretty unsatisfying from security perspective. Again the system doesn't escape "watching" attacks.

4) Other graphical defenses: D. Hong and Sobrado designed different models to prevent shoulder-surfing [10]. They proposed a model system with a combination of graphs and alphanumeric password. The system provides many different pictures each with slight differences that can easily be identified by human but not by computers. When first using this system, a user needs to select four graphs and specify a certain string corresponding to each change of these four graphs. Note that different users can specify different strings for the same graph. In an authentication process, the system will randomly generate 11x11 graphs, four of which are specified in advance by the user. The slight changes with those four graphs are randomly generated and are to be identified by the user using the previously defined string to enter into the authentication window. The scheme is designed to prevent shoulder-surfing. However, despite its large password space, the authentication process is complex and time-consuming.

III. Graphical Password Scheme Design

The basic idea in this scheme is to use a disc and a background where the user is only required to turn the disc until the pattern matches. Moreover, the characteristic pattern or sub-images won't get compromised since any clicks or special operations on them will be excluded in the process.

The design of the human brains makes it automatically discard random information such as a seemingly meaningless alphanumeric sequence, which is commonly used to generate secure password. Although some of them do stand for specific meanings, it just does not impress our brains enough to be easily remembered. In order to remember all the passwords linked to his various accounts the user has to note them either on a book or in a computer file. This creates huge security breaches as neither the book nor the file is usually secured.

Fig. 1 From left to right: DAS; Passpoints; Passfaces.

A much easier alternative would be to take advantage of images or patterns, which are much easier to memorize for the human brain and by then meet the need for password and authenticated access. (There are profound theories behind it from psychological aspect which lies beyond the scope of this paper and will not be further discussed here.)

However, as discussed above, current graphical passwords can hardly guarantee its security when faced with numerous close observations. So how to input password in an open environment without the input criteria and content exposed is our main goal when designing the scheme. We offered a good solution to this question in this paper.

A. *Disc-Arranged Graphical Password Scheme*

For clarity, we will only consider a simple case where for every validation process, there is only one characteristic piece of image that needs to be recognized by the user. We assume that Fig. 2 (a) represents the characteristic figure and that only one characteristic region on the graphic disc pre-chosen by the client needs to be identified during the authentication process as showed in Fig. 2 (b). Note that the user does not need to write down any information in order to easily remember the "secret". He could even upload his own image of favour. As the process only ask the user to recognize the characteristic image among all other images presented on the screen, it is much easier to achieve than to reproduce a password out of nothing as it is always the case for static password. In fact, there is a solid and sophisticated psychological groundwork for it which lies beyond our scope of research.

1) Input process: The input process can be split into two steps as shown in Fig. 3. **Step1:** Among all the rings of consecutive figures, find the characteristic image. (Only one ring is showed below in Fig. 3 whereas in an actual case, there would be at least 10 rings arranged on the same centre but with increasing radius. Moreover, the number of the identical-sized sub-images covering each ring is increasing as the circumferences increases. Therefore, there will be abundant sub-images for the selection pool, thus promoting security against a random guess.) **Step2:** Identify your characteristic region -- the blue feather end on Lena's hat in this case; position the graphic disc on the characteristic image, a little bit like when you turn a password disc on a safe. (There is no need to place it precisely at the middle of the characteristic image as it has some tolerance which can be set beforehand according to different security demand.

Fig. 2 (a) one characteristic image—a red and black transformer
(b) one characteristic region—blue feather end on Lena's hat

Fig. 3 From left to right and from top to bottom: spot the characteristic image, which is already highlighted in ginger; identify your pre-chosen characteristic region on graphic disc as framed already; press a "rotate" button on the screen which is omitted here to turn the region at the characteristic figure spotted previously; eventually, the region" blue feather end" pointed at target figure as framed above and the graphical

B. *Discussion*

1) Selection pool: For each user authentication process, the sub-images are randomly generated from the selection pool to form a unique background. Usually the client is required to choose or upload more than one characteristic image during the initialization step and then later identify only one of them at a time during the authentication process. This is due to the size of the password space that renders the probability of guessing the correct characteristic image on the background very low. Note that: *SelMembers* denotes the number of selection pool members; whereas *SubImages* denotes the number of sub-images for each background. Hence: SelMembers > SubImages.

2) Background and disc image arrangement: The background sub-images can be arranged in different random ways and the graphic disc can be placed at an arbitrary position each time. The position can be either chosen by the user or randomly generated by the computer to achieve higher security. For example, Fig. 4 shows a different arrangement where the "blue feather end" is aiming at the twin popsicles. Note that the area covered by the graphic disc can be either left blank or filled with images as mere decoration. To further enlarge the password space, a graph with more potential candidate points that can be chosen as characteristic regions should be the best choice for a perfect graphic disc. For example, Fig. 5 shows a better graphic disc than the "Lena" version showed in Fig. 2 (b).

Fig. 4 Seemly random background arrangement and an arbitrary position for graphic disc

Fig. 5 "Fairy Tales" – a better graphic disc with a characteristic region randomly chosen as framed in ginger: little man on the globe

4. A Detailed Application in One-Time Password (OTP)

A. Background Information about OTP

A better and more secure way of authentication is the so called "one time passwords"(OTP) strategy where instead of authenticating with a simple static password, each user carries a device ("token") or a software that generates passwords valid only for one use, one login session or one transaction for instance [11, 12].

1) Advantages of OTP over static password: Using traditional static passwords for authentication is no longer reliable for its security drawbacks: passwords can be guessed, forgotten, written down and stolen; thus this password mechanism fails to meet the prerequisites of high-security and high-reliability in today's online banking system.

2) Attacks on OTP: On a more general scale, algorithms for OTP mechanism can be classified into three main categories: time-base, HMAC-based (event-based), and challenge-response algorithms [13, 14, 15]. For these three OTP mechanisms, apparently the offline attacks won't work since the password for each authenticated access is limited to a short life span and the credentials will never be pre-exposed, which is enough to thwart both malicious software and phishing. However, online channel-breaking attacks still cast great doubt on the reliability of all these mechanisms. Once the communication channel is intercepted, the invader can steal vital information transmitted and fake to be the other side for both the user and the server. To be more specific, among the three OTP mechanisms, challenged-response based one-time password somehow wins over the two others as having a short life, being nondeterministic and ideally bounded to a previous communicated account. Nevertheless, it still cannot escape the whole scope of online attack. Channel interception somehow resembles remote monitoring in that the information stream is monitored, thus causing credential compromise. Therefore, it can be included in the range of "watching" attacks. In addition, shoulder-surfing is hard to prevent in all these three mechanisms.

B. Graphical Password Scheme Countermeasure: Disc-Arranged Graphical Input

A detailed application of the graphical password scheme proposed in the previous section is presented here. It mainly focuses on "watching" attacks: snapshot; remote monitoring; online channel-breaking; and shoulder-surfing. More precisely, snapshot aims at obtaining the client's OTP value by printing the screen; shoulder-surfing attacks jeopardize the security of an account when surrounding people spy on the user; online channel-breaking attacks intercept sensitive information between user and server; remote monitoring attacks are done by third party daemons or internal staff members who eavesdrop the whole authentication process. This scheme is actually an extra encryption process with the goal of enhancing the security of the input process.

For clarity, we consider a simple case where for every authentication process; there is only one characteristic figure that needs to be recognized by the user. Recall Fig. 2 (a) as the characteristic figure.

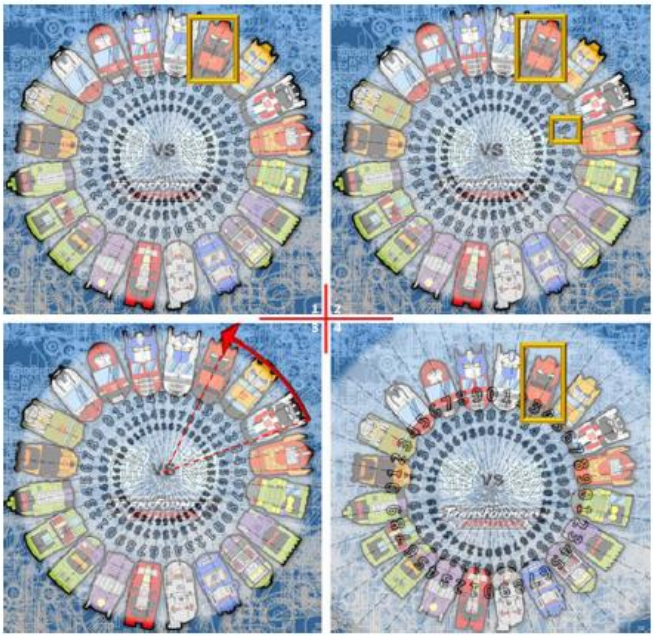

Fig.6 From left to right and from top to bottom: spot the characteristic image, which is already highlighted in ginger; to input the target digit "3", pick one of the "3"s on the digit disc as framed already; press a "rotate" button on the screen which is omitted here to turn "3" at the characteristic figure spotted previously; eventually, we have a "3" in the middle of target figure as framed above and the graphical input of one digit "3" is done.

1) Single-digit input process: The input process can be split into two steps similar to the previous design illustrates in Fig. 6. **Step1:** Among all the rings of consecutive figures, find the characteristic image. (Only one ring is showed below in Fig. 6 whereas in an actual case, there would be at least 10 rings arranged on the same center but with increasing radius.) **Step2:** Suppose the digit to enter in the OTP value is "3" (Fig. 6); turn the digit disc until "3" positions on the characteristic image. (There is no need to point the digit precisely, it is only required that no neighboring digit is closer than the target digit.)

2) Complete authentication process: **Step1:** Obtain OTP value.**Step2:** Determine all the digits to enter according to the instruction displayed in computer interface. **Step3:** Wait for a background and the digit disc to appear. **Step4:** Input one digit following the single-digit input process in section 4-B-1. **Step5:** Wait for another background and the digit disc to appear. **Step6:** Input the next digit following the single-digit input process. **Step7:** Repeat Step5 and Step 6 until all digits are input. **Step8:** Wait for the outcome of the authentication process. The whole process is showed in Fig. 7.

3) Rotating coordinate system and angle-determine method: "Angle-determine" method in a rotating frame is used to test the idea where we first appoint one point for the target sub-image (*TarImage*) and when one digit input is done, the angle between the point-to-centre line and the left-nearest zero (*LeftNearZero*) line is calculated. In this setup the zeros are fixed points with a stationary centre (0, 0). Within the calculation, *TarImage* is considered as mobile and is rotated by the user. Fig. 8 illustrates the idea of such a setting. Once we get the angle between (*Angle*), we can then calculate the digit closest to *TarImage* and regard it as the target digit input (*TarDigit*). We illustrate this method in the previous case: First note that there are a 0~9 within every 90 degree out of 360 degree with one digit change per 9 degree. Therefore, the software shall limit the rotation to 90 degree, excluding extra labor for user. Mark some coordinates as showed in Table 1, and the *Angle* can then be obtained. Finally the *TarDigit* can be determined as follows. The whole idea is illustrated in Fig. 9. Alternatively, we can calculate the angle between *TarImage* and *FlagZero*, as showed in Fig. 10 and then get the *TarDigit* with similar process but showing slight differences. The complete calculation process is showed as follows.

$D1 = \text{flag}$

$D2 = \text{distance between TarImage \& center}$

$$D2 = \sqrt{XtarImage^2 + YtarImage^2}$$

$D3 = \text{distance between leftNearZero \& TarImage}$

$$D3 = \sqrt{(XleftNearZero - XtarImage)^2 + (YleftNearZero - YtarImage)^2}$$

$$Angle = \cos^{-1}(D1^2 + D2^2 + D3^2 - 2 \times D1 \times D2)$$

$$\text{LeftNearZero} = \begin{cases} FlagZero, & TarImage \in I \\ SecondZero, & TarImage \in II \\ ThirdZero, & TarImage \in III \\ FourthZero, & TarImage \in IV \end{cases}$$

We can then get: $\text{TarDigit} = \text{round off}(\frac{Angle}{9})$

TABLE I Coordinates setting

	FlagZero	SecondZero	ThirdZero	FourthZero	Center	TarImage
X	flag	0	-flag	0	0	XtarImage
Y	0	flag	0	-flag	0	YtarImage
"flag" is a pre-determined positive integer.						

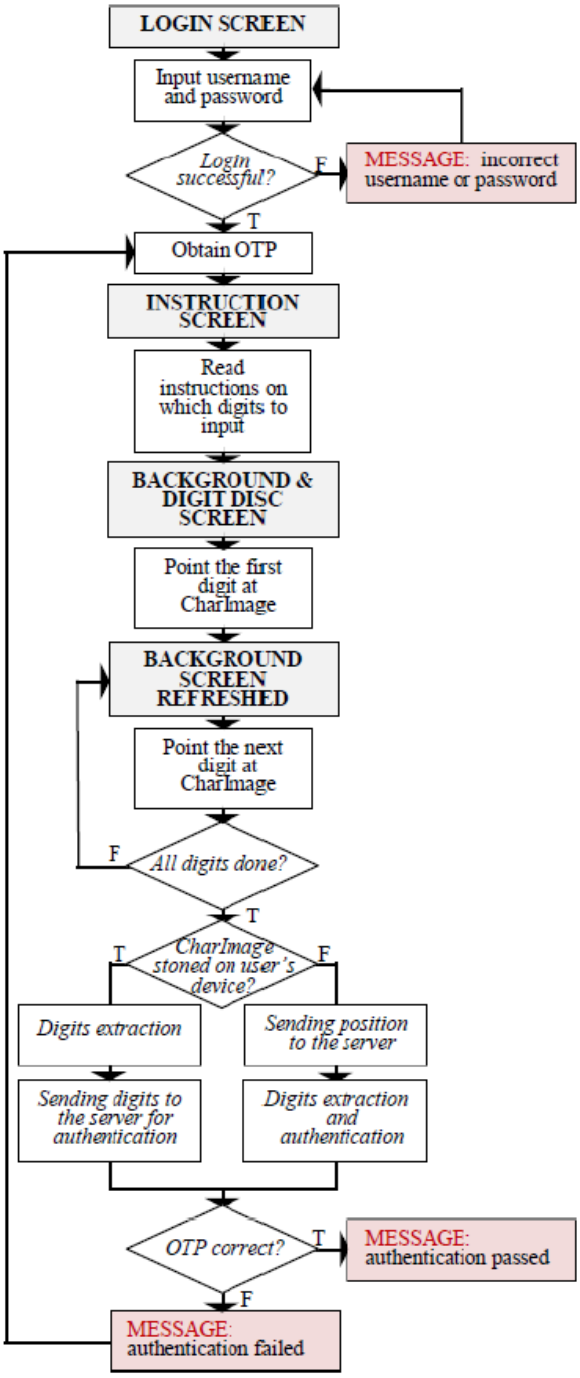

Fig.7 Flow chart for complete authentication process

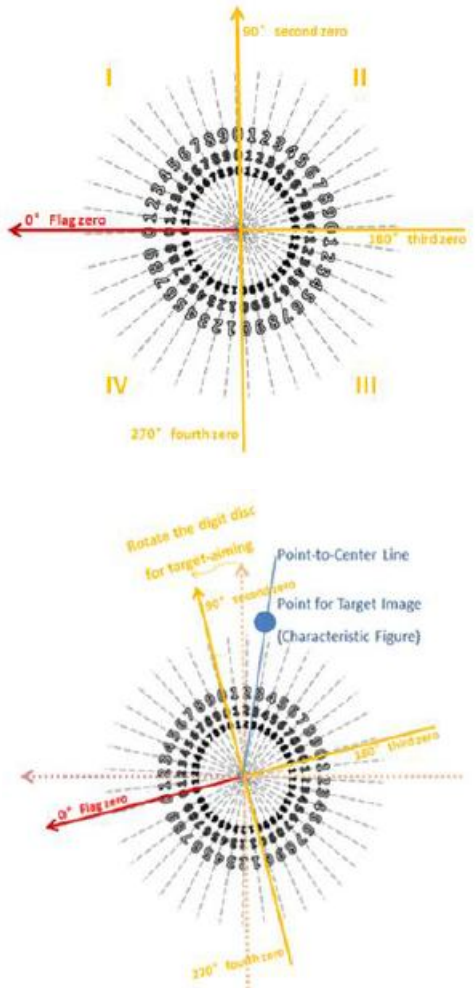

Fig. 8 Rotating coordinate frame with *TarImage*

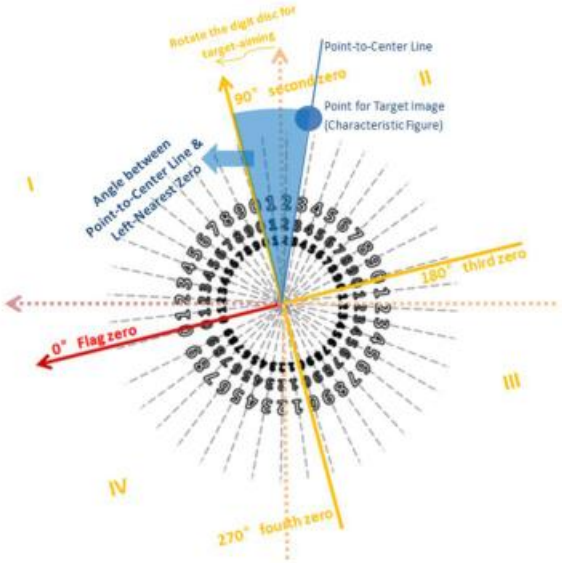

Fig. 9 *Angle* between *TarImage* and *LeftNearZero*

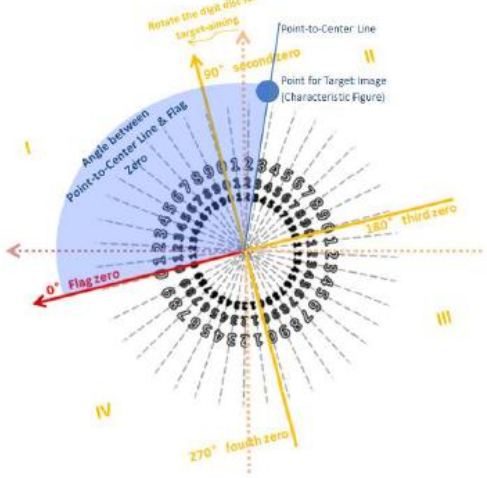

Fig. 10 *Angle* between *TarImage* and *FlagZero*

4) Security analysis: If an attacker wants to get the OTP through a snapshot, and assuming he is able to capture every move at undetermined time interval, he still does not know which digit is the target within one digit input process. After a large number of observations the attacker has still no clue about the image of favor. Same applied to remote monitoring and channel breaking as long as at least part of the decryption process is performed on the server side. As for shoulder-surfing the attacker, especially if acquainted with the client and used of spying on him, seems to have all the advantage to steal the token. In fact he is still blocked by the input encryption process that can only be performed by the true user. In practice, the number of sub-image-loop in the disc-arranged background image must be increased to obtain higher security. We present the security analysis for an n-loop case. The number of sub-images changes for each loop equals the number of sub-images within the innermost loop. Numerical security analysis is showed in Table II~V along with Fig. 11~14. **Case1:** OTP value and characteristic image are both unknown; the probability of the correct guess under random operation. **Case2:** Under the threat model of shoulder-surfing; OTP value known and one digit input process is observed; the probability of the correct guess of the characteristic image. N (…) denotes the number of.

TABLE II Case 1for single digit input

a	b	c	n-loop
One 0~9	100	50	1/10n
		100	1/20n
	200	100	1/10n
		200	1/20n
Two 0~99	100	50	1/100n
		100	1/200n
	200	100	1/100n
		200	1/200n
a: N (digits input each time) b: N (digits on the digit-disc) c: N (innermost-loop sub-images)			

TABLE III Case 1 for m times' repeat

a	b	c	n-loop
One 0~9	100	50	$(1/10n)^m$
		100	$(1/20n)^m$
	200	100	$(1/10n)^m$
		200	$(1/20n)^m$
Two 0~99	100	50	$(1/100n)^m$
		100	$(1/200n)^m$
	200	100	$(1/100n)^m$
		200	$(1/200n)^m$
a: N (digits input each time) b: N (digits on the digit-disc) c: N (innermost-loop sub-images)			

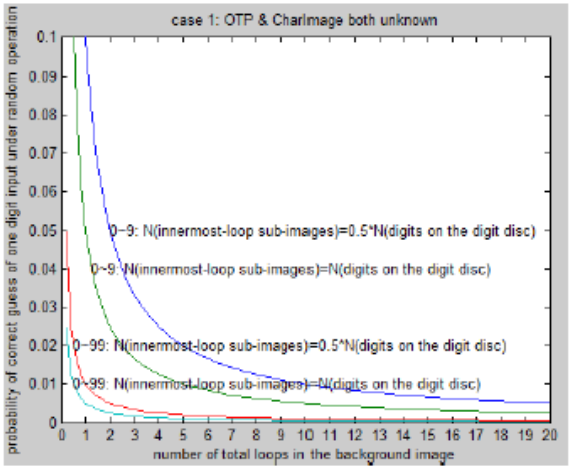

Fig. 11 Case 1 for single digit input

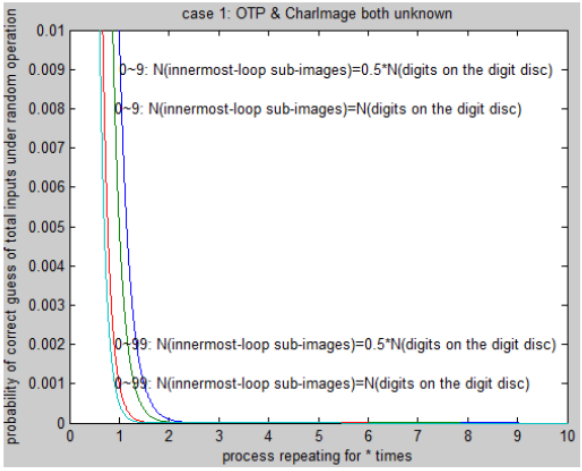

Fig.12 Case 1 for m times' repeat (m=10 in this

There is a balance between case 1 and case 2: the probability of a correct guess decreases when the OTP value and the characteristic image are both unknown, on the contrary it increases when the OTP value is leaked such as in the shoulder-surfing model case. In other words increasing the number of digits improves the user experience, he has less repeated processes to complete when the total number of input digits is fixed, and also promotes the security performance when the OTP value and the characteristic image are unknown. On the other hand, it can cause the characteristic image to be compromised under the assumption that OTP value is exposed. The security performance within one digit input increases ***linearly***; the overall security performance within the whole input process increases ***exponentially***, which makes the difficulty of an attack to also increase ***exponentially***.

TABLE IV Case 2 for single digit input

a	b	n-loop
One 0~9	40	1/4n
	100	1/10n
	200	1/20n

TABLE V Case 2 for m times' repeat

a	b	n-loop
One 0~9	100	$(1/10n)^m$
	200	$(1/20n)^m$
Two 0~99	100	$(1/n)^m$
	200	$(1/2n)^m$

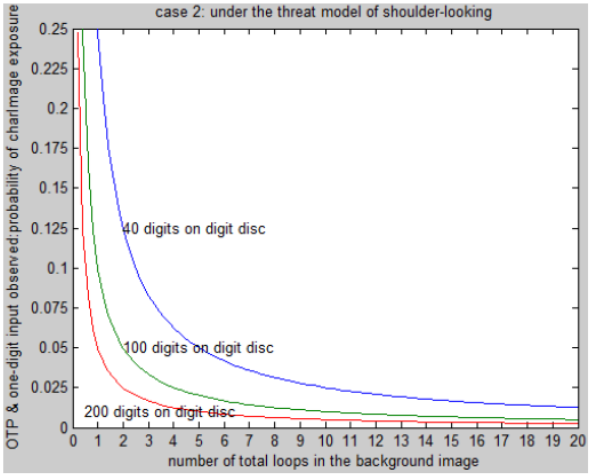

Fig. 13 Case 2 for single digit input

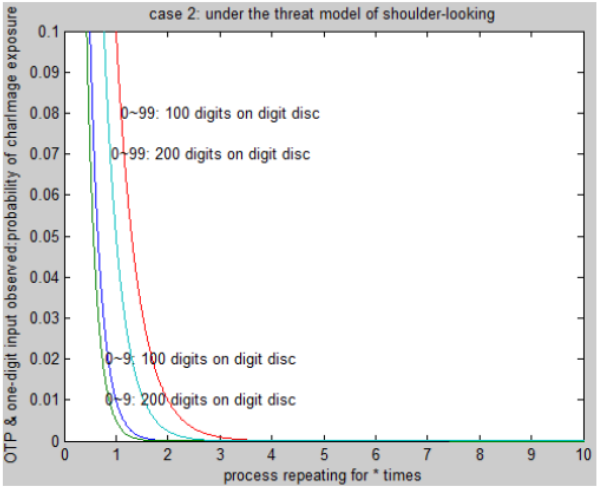

Fig. 14 Case 2 for m times' repeat (m=10 in this example)

C. *Other-Arranged Graphical Input*

Aside from the previous arrangement, the background can be other shaped—random; matrix; star; etc. depending on how big the image bank is and what security level is to be achieved. Here we briefly present another random-arranged background and illustrate it in Fig. 16. The characteristic image is showed in Fig. 15. In this case, the very spot to initialize the digit disc can also be determined by two or more characteristic images. For example, one can first find a region circumscribed by multiple characteristic images, and with one mouse-click locate the digit disc's center for that region. The user can then look for the target digit to enter at the fourth characteristic image. The process largely improves the overall security. Note that the angle-determine method can also be applied together with a rotating coordinate system to realize the whole process in a similar fashion as in subsection 4-B-3.

Fig. 15 One characteristic image—twin

Fig. 16 From left to right and from top to bottom: part of original background pool; TarImage framed in ginger; one random mouse-click to initialize the digit disc; turn a "2" which is assumed to be the current digit to input at the TarImage.

V. Future Work

Graphical password scheme requires further work mainly on the software aspect to promote algorithms and operating efficiency. Research for better compatibility with the running environment is also primordial in order for it to be embedded into user authentication processes such as the whole OTP validation system. Also, the initial process of building the bank for characteristic images requires a deeper exploration as to how long the period should last before the images of favor expires or how the images of favor should be chosen or built, especially in conditions with high security demands. Moreover, the problem of designing both the background and the disc is of major importance in order to generate a well-functioned graphical password authentication. Appropriate mathematical aspects would greatly improve the algorithm efficiency and the user experience.

Acknowledgment

Special thanks would like to be delivered to Manuel Charlemagne for write-checking and other students in my research group for their helpful comments and inspiring guidance during The SJTU project PRP21. This work was supported by the National Natural Science Foundation of China (61073149 and 61272440), State Key Laboratory of ASIC & System (11KF0020), Key Lab of Information Network Security, Ministry of Public Security (C11603) .

References

[1] G. E. Blonder, "Graphical Passwords," United States patent 5559961, Lucent Technologies, Inc. (Murray Hill, NJ), 1996.

[2] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The Design and Analysis of Graphical Passwords," *In Proceedings of 8th USENIX Security Symposium*, 1999.

[3] R. Dhamija, A. Perrig, "Déjà Vu: User Study Using Images for Authentication," *9th USENIX Security Symposium*, 2000.

[4] Rivner, Uri, "Anatomy of an Attack," RSA Fraud Action Research Labs, April 2011.

[5] Konstantinos Chalkias, Anastasios Alexiadis, and George Stephanides, "A Multi-Grid Graphical Password Scheme," http://inf.ucv.ro/~aidc/proceedings/2006/10%20kchalkias.pdf.

[6] Di Lin, Paul Dunphy, Patrick Olivier, and Jeff Yan, "Graphical Passwords & Qualitative Spatial Relations," *Symposium on Usable Privacy and Security (SOUPS)*, July 18-20, 2007, Pittsburgh, PA, USA.

[7] S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskiy, and N. Memon, "Authentication Using Graphical Passwords: Effects of Tolerance and Image Choice," *Symposium on Usable Privacy and Security (SOUPS)*, 2005, Carnegie-Mellon University, Pittsburgh.

[8] Dirik A. E., N. Memon, and J. C. Birget, "Modeling User Choice in the PassPoints Graphical Password Scheme," Int. J. of Human-Compu. Stud., 63: 102-127, 2007.

[9] RealUser www.realuser.com

[10] D. Hong, S. Man, B. Hawes, and M. Mathews, "A Password Scheme Strongly Resistant to Spyware," *In Proceedings of International Conference on Security and Management*, 2004, Las Vegas, NV, USA.

[11] Krawczyk H., Bellare M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, February 1997.

[12] N. Haller, C. Metz, P.Nesser, and M.Straw, "A One-Time Password System," RFC 2289, February 1998.