A MIXED LOGIN SCHEME PERFORMED ON MOBILE DEVICE TO RESIST MULTIPLE ATTACK

Jie Wan[1], Liang Liu[1], Dai Hua[2] and Wei Liu[3]

[1]College of Computer Science and Technology, NanJing University of Aeronautics and Astronautics, NanJing, China
[2]School of Computer Science, Nanjing University of Post &Telecommunications, NanJing, China
[3]Nanjing Institute of technicians, NanJing, China

ABSTRACT

Nowadays text-based password has been widely used in our daily life. However, rather than choose a complex text password people prefer to use a brief password so that they can remember it easily. Moreover, with the rapid increasing use of mobile applications, people often input passwords in public. Attacker can perform shoulder surfing attack to observe the password directly with naked eyes or some video record devices. In order to resist the shoulder surfing attack, a number of authentication schemes based on graph have been proposed. However, graph-based password is totally different from text-based password. It's difficult for users to memorize two different kinds of passwords. In this paper, we propose a mixed login scheme called MixedKey which mixes graphic and traditional textual password. The login indicator in the scheme is randomly and safely generated for each login. MixedKey divides each password into characters, which connects graphic and text-based password. Users could login our system in both public and private situations with just one password. We also implemented MixedKey and conducted experiments to measure the memorability and usability. The results show MixedKey outperforms the existing schemes.

KEYWORDS

Shoulder Surfing Attack, Graph-based Password, Text-based Password, Login Scheme

1. INTRODUCTION

Text based password has been widely used since personal intelligence information service becomes popular. Various ways of text-based password such as PIN password and length- limited text password are used to examine whether the user is certified. Usually, text-based password is robust enough to defend the Random Guess Attack and Violent Crack Password Software.

However, text-based password has its backwards as well. Text based authentication schemes ask users to insert password with keyboard or touch screen. However, attackers could use cameras or video recorders to record the process of authentication to perform Shoulder Surfing Attack [1] to recover password, PIN or some other sensitive information. For example, there was a shopkeeper happened to found a mobile phone in his shop, then he got the video and cracked the password with the help of surveillance camera installed in the store [2]. Text-based password is insecure in public, users may leak out their passwords when they input passwords within the sight

Natarajan Meghanathan et al. (Eds) : ICCSEA, WiMoA, SCAI, SPPR, InWeS, NECO - 2019
pp. 135-150, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91811

of other people. For convenience, many users tend to use similar password to login different accounts, so how to defend shoulder surfing attack has become a serious problem.

As a result, a lot of graph-based authentication schemes [3], [4], [27], [28] were proposed to overcome the backwards of text-based password. Most of them produce a unique indicator on the screen directly for users to login every time (For example, a verification code is an indicator during one authentication). However, the indicators are usually required to be generated and transmitted privately. It is difficult to guarantee privacy of indicators. Take Figure 1 for an example, many researches directly provide indicators on the screen, which is easy for attackers to capture[5]. What's more, most graph-based schemes are independent of text-based schemes. It means users need to remember two kinds of passwords to login, which is a challenge for their memories.

Figure 1. Obtain the login indicator (E,11) directly.

In this paper, we proposed a hybrid authentication scheme named MixedKey that protects users from being victims of shoulder surfing attacks when inputting accounts in public.

Different from traditional graph-based schemes, our scheme does not generate indicator directly on the screen. Instead, we propose a novel way to create indicators. Indicator is hidden in the ElementArray (As showed in Figure 2) of password elements to confuse attackers, and only the user who knows the first character of password could find the indicator (the location of the first character of password in the first ElementArray.), which protects the indicators in public. MixedKey generates one ElementArray for each password character, user needs to move password element to the indicator. Moreover, our scheme connects text and graph-based password with the help of randomly generated ElementArray.

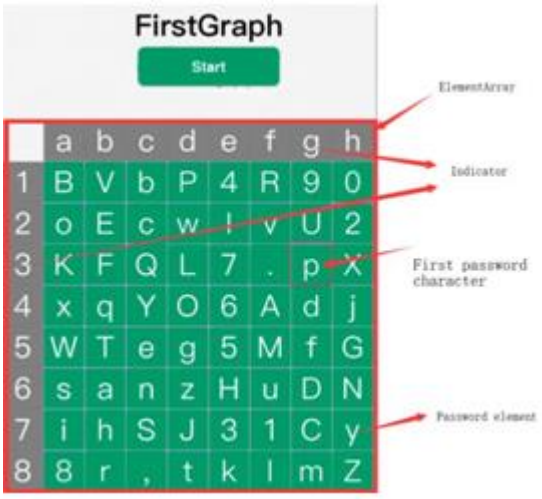

Figure 2. The components of ElementArray.

The structure of this paper is showed as below: Section 2 provides the related backgrounds about graph-based authentication schemes. Section 3 discusses the attack models. Section 4 is mainly about the MixedKey proposed in this paper. Section 5 and 6 describes how we perform our experiment and shows the results. Section 7 is about the security analysis and Section 8 is summarize of this paper.

2. RELATED WORK AND BACKGROUND

In the early days, the graphical capability of handheld devices was weak; the colour and pixel it could show was limited. Under this limitation, the Draw-a-Secret (DAS) technique was proposed by Jermyn et al. in 1999, where the user is required to re-draw a pre-defined picture on a 2D grid [6]. In 2005, Susan Wiedenbeck et al. introduced a graphical authentication scheme PassPoints [3], and at that time, handheld devices could already show high resolution colour pictures. Using the PassPoint scheme, the user has to click on a set of pre-defined pixels on the predestined photo. With a correct sequence and within their tolerant squares during the login stage. Moreover, Marcos et al. also extended the DAS based on finger-drawn doodles and pseudo signatures in recent mobile device [7], [8]. This authentication system is based on features which are extracted from the dynamics of the gesture drawing process (e.g., speed or acceleration). These features contain behavioral biometric characteristic. In other words, the attacker would have to imitate not only what the user draws, but also how the user draws it.

However, the above authentication schemes are still vulnerable to shoulder surfing attacks [19,20] as they may reveal the graphical passwords directly to some unknown observers in public.Hung-Min Sun et al proposed PassMartix [5], a graphic based scheme which divides a picture into 77 parts. Users select one part as password and move horizon and vertical bars. However, if observers are able to capture the indicator, the passwords can be cracked easily. Song, Yunpeng et al proposed a Multi-touch authentication which rely on hand geometry and behavioural information, as showed in Figure 3, TFST gestures (here touch with fingers straight and together), both hand geometry information and behavioural characteristics (such as the touch pattern, area, time, pressure, etc.) are recorded and used for user authentication. With stretching fingers moving on the screen, a unique multi-touch gesture generates as the password to login [9]. Zhen Linget al. introduced a novel secure back of device input system [10] with the help of accelerometer and orientation sensor in the smartphone to defend side channel attack. With tilting their devices users could move the on-screen cursor to the intended key and tapping on the back of their phones to

finish the authentication, which hides the moving trace of finger to defend shoulder surfing attack and smudge attack. Figure 4. shows the main idea of the scheme.

Figure 3. TFST gesture performance

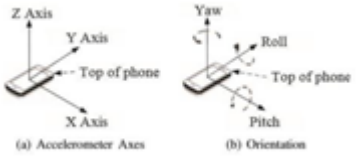

Figure 4. SecTap input method

In addition to graphical authentication schemes, there were some researches on the extension of conventional personal identification number (PIN) entry authentication systems. In 2004, Roth et al. [11] presented an approach for PIN entry against shoulder surfing attacks by increasing the noise to observers. In their approach, the PIN digits are displayed in either black or white randomly in each round. The user must respond to the system by identifying the color for each password element. After the user has made a series of binary choices (black or white), the system can figure out the PIN number the user intended to enter by intersecting the users' choices. This approach could confuse the observers if they just watch the screen without any help of video capturing devices. However, if observers are able to capture the whole authentication process, the passwords can be cracked easily. In order to defend the shoulder surfing attacks with video capturing, FakePointer [4] was introduced in 2008 by T. Takada. In addition to the PIN number, the user will get a new answer indicator each time for the authentication at a bank ATM. In other words, the user has two secrets for authentication: a PIN as a fixed secret and an answer indicator as a disposable secret. The answer indicator is a sequence of n shapes if the PIN has n digits. At each login session, the FakePointer interface will present the user an image of a numeric keypad with 10 numbers (similar to the numeric keypad for phones), with each key (number) on top of a randomly picked shape. The numeric keys, but not the shapes, can be moved circularly using the left or right arrow keys.

There exists many kinds of side-channel attacks against patterns as well, including smudge attacks [15, 16, 17], sensor-based attacks [18] and random guessing attacks [21, 22, 23]. Some researches on side-channel attacks were also studied during last decade as well. Aviv et al. demonstrated that it is possible to reconstruct a locking pattern by analysing the oily residues left on the screen [12]. This method is highly restricted as oily residues can be messed up by any on-screen activities after pattern drawing. Wu et al. [13]study such a camera-based attack on mobile phones and Ye et al. [14] report to crack the Android pattern lock in five attempts. David Kim et al. [24] proposed a visual authentication scheme for table top interfaces called “Color Rings”, These existing works are essentially orthogonal to this paper, which do not threat the performance of MixedKey obviously.

3. PROBLEM STATEMENT AND ATTACK MODEL

3.1. Problem Statement

When the users log in their accounts in public, they would leak out the passwords to someone else whether consciously or not. Those who intends to crack your password will use cameras or monitors everywhere to spy on your log in phase, even analyses it repeatedly. Attackers will access the account once they get your password, which does harm to the security of users' assets. Here follows the problems addressed in this paper:

a) How to perform effective authentication in various situations? We concentrate on defending shoulder surfing attack in public and ensuring usability in the private environment.
b) How to integrate text based authentication and graph based authentication?
c) How to reduce extra information that users need to remember in the log-in phase?

3.2. Attacker Model

3.2.1.Shoulder Surfing Attack

The attacker models in this paper mainly are based on shoulder surfing attack, including three kinds of attacks as follows:

a) Attack with naked eyes, which is the most wildly way of leaking out sensitive messages.
b) By means of video cameras, attackers can capture the whole process of the authentication.
c) With the help of video cameras, attackers can capture the whole process of the authentication several times.

The last type of attack requires more effort and techniques from attackers, so if one authentication scheme is able to resist the last kind of attack, it is also secure against the other types of attack. Traditional PIN and text-based password, even some graph based authentication schemes [4] are vulnerable to the first type of attack and thus are vulnerable to the second and third kind of attack.

3.2.2.Violence Attack

Attacker guesses users' passwords randomly or use violent attack software to crack password.

3.2.3.Smudge Attack

Previous research [12] shows that when users touch the screen of smart device to perform the pattern lock, with fingers move in the screen, smudge left on the screen. The attackers could perform the attack easily after recording the smudge.

4. MIXEDKEY

The proposed graph based schemes are faced with following problems: (1) the security backwards of graph based schemes, (2) the independence of two kinds of authentication schemes, (3) the problem of extra information users need to remember during authentication. In order to solve these problems, we proposed a hybrid authentication scheme called MixedKey. In MixedKey, users login the system with just one kind of username and password in both text and graph based ways.

4.1. Overview

Our system includes three sub systems: register, private authentication and public authentication. In the register sub systems, users need to input strings as username and password. The password will be stored in the form of n characters sequence and saved in the cloud. Then our authentication system provide users with two ways of authentication: private authentication and public authentication. When users perform log-in phase, they need to insert username first. The system will provide two kinds of authentication if the username is legitimate. Users could insert the password directly in private and secure environments like their home, which is secure and fast. They could also use public authentication to improve the security level in public and insecure environments without leaking out sensitive information.

In the public authentication sub system, MixedKey provides an initializing ElementArray(As shown in Figure 5(a)) which is composed of randomly placed password elements. Users need to find and memorize the location of the first character of the password, which is the indicator(As shown in Figure 2). If user presses the 'start' button, the system will rearrange elements to generate a new ElementArray as shown in Figure 5(b), users need to move the password element to the indicator showed in the initializing ElementArray and press 'ok' to finish authentication of this password character. The system will repeat this process until the last password character is authenticated. After that, MixedKey will confirm whether the verification process is successful or not.

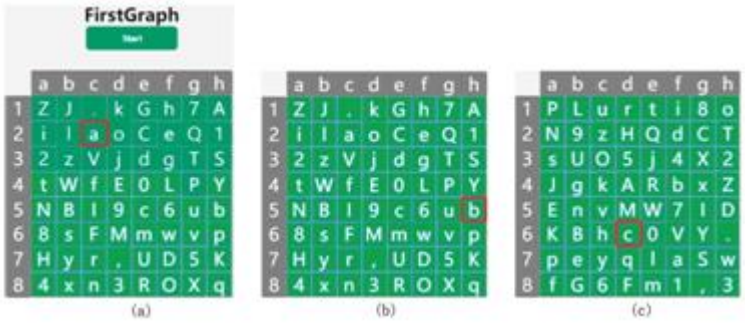

Figure 5. The authentication progress of ‘abc’

For example, we assume that the username is 'admin' and the password is 'abc'. User needs to input 'admin' first. MixedKey generates an initializing ElementArray as Figure 5(a) shows. User 'admin' needs to find the location of the first password character 'a'. As shown in Figure 5(a), the location of 'a' is (c,2). User 'admin' needs to memorize (c,2) as indicator.Then MixedKey generates the second ElementArray. User needs to move the second password character 'b' to the location of the indicator namely (c,2). Finally, MixedKey generate the last ElementArray. User needs to move the last password character 'c' to the location of the indicator. Each password element can be moved circularly so that the attacker would be confused. For example, if the password character is 'a', the indicator is (3,g), the user can not only move the password element 'a' from (1,h) to (3,g) (as shown in Figure 6(a) and 6(b)), but also move 'Á' from (3,e) to (5,d) (as shown in Figure 6(c) and 6(d))to authenticate 'a' successfully. As each password element can be moved, attackers can hardly figure out which is the real password character even they watch the whole process of authentication.

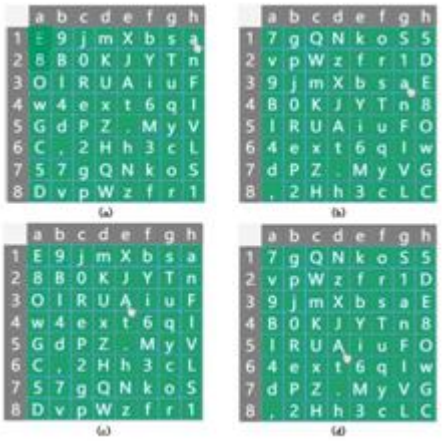

Figure 6. Each password element can be moved

4.2. Detailed Steps

4.2.1.Registration

We describe the process of registration in this section as shown in Figure 7. The user needs to create an account with a username and a password during the registration phase. After the user submits, the information will be stored in the cloud for verification in the following authentication process. The registration page is shown in Figure 8, users need to fill username and password in the two text boxes to finish the registration.

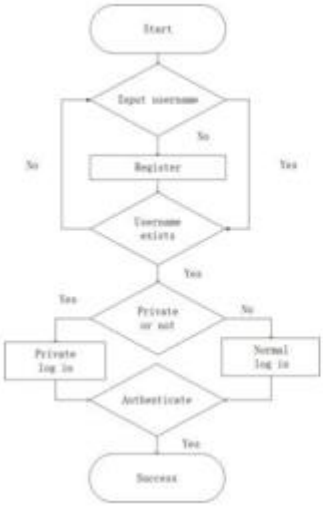

Figure 7. Flow chart of implementation of our system.

Figure 8. Registration phase

4.2.2.Authentication in Private Situation

The private authentication is designed for authentication in the secure situations. Users need to input the username and password in the text boxes in Figure 9(a). Then the system will verify whether the account is valid. The flow chart of private authentication is showed in Figure 9(b).

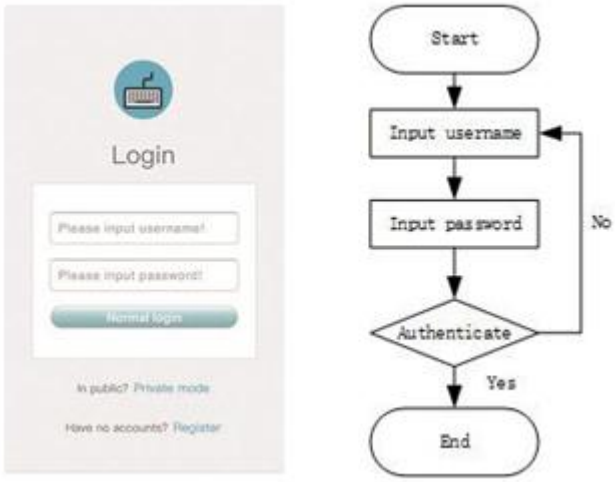

Figure 9. private authentication and it's flow chart

The following describes the steps in detail:

a) The user enters the username and the password which he/she filled in during the registration phase.
b) The system verifies the username and password.

4.2.3.Authentication in Public

Public authentication sub system is generally used in insecure scenes such as in a restaurant that are vulnerable to shoulder surfing attacks. It is composed of the following modules:

a) Username input module

This module is used to verify whether the username is legal. As shown in Figure 10, users need to input the username. The system would compare the username entered by the user with the accounts stored in the cloud. If the username exists, MixedKey would turn to authentication module, else it would return to the login page again.

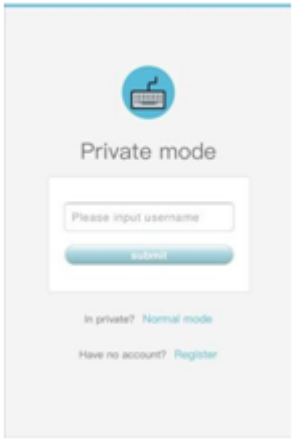

Figure 10. username input module

b) Password ElementArray generating module

As shown in Figure 5(a), we divide a box in the screen as ElementArray to store password elements. The ElementArray is M*N matrix which contains several password elements. In our experiment, we generated 64 password elements in the form of 8*8. Although more elements may increase the password space, it increases the authentication time as well. Moreover, more elements leads smaller size, which brings more possibility of mistake. A character or a punctuation or a number is stored in a password element. The password elements are random arranged. We use a horizontal bar with a sequence of letters and a vertical bar with a sequence of numbers. These two bars are provided as coordinate axis for users to find the location of each password element. As shown in the Figure 11, Vertical axis is indicated from '1' to '8' and horizontal from 'a' to 'h'.

Figure 11. Vertical and horizontal coordinate axis

c) Password element location control module

This module aims at getting all password elements tightly aligned when users' fingers move on the touch screen. We record the whole touch process, from the event "touching start" to "touching move" and then "touching end". In the "touching start" event, we record the coordinate of the finger in the screen and the password element which the finger is touching on. The selected password element will move with finger and stop when the finger puts up from the screen. Then the ElementArray will rearrange with the location of the moved password element. For example, the location of the password element 'a' is (1,h) in Figure 12(a). If user moves 'a' to (3,g), each password element in the ElementArray will move once to the left and move twice down(see Figure.12(b)).

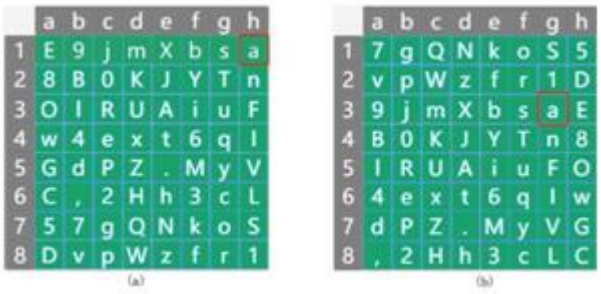

Figure 12. Password element location control module

c) Password verification module

This module performs authenticating of password. The authentication is successful if the coordinate of each password character is coincide with the indicator.
Here are the specific steps for private login phase:

1. The user inputs the username.

2. If the username is legitimate, MixedKey gives an initializing ElementArray. The user needs to find and memorize the location of the first character of password as the indicator. Then the authentication starts.
3. Let i equals 2.
4. MixedKey generates ith ElementArray. The user needs to move the the ith password element to the indicator showed in the initializing ElementArray and press 'ok' to finish authentication of this password character.
5. Repeat 4. until the last character is authenticated.
6. The Password verification module compares the position of each password element with indicator. Only the nth character of the password matches the password element whose location is same as the indicator in the nth ElementArray ($1 \leq n \leq$the length of password), the authentication will succeed.

The flow chart of private authentication is showed is Figure 13.

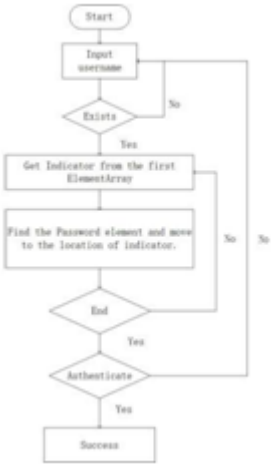

Figure 13. The flow chart of private authentication.

5. EXPERIMENT

5.1. Experimental Design

In the experiment, two indicators are used to evaluate the performance of MixedKey: accuracy and usability.

We intend to evaluate the performance of MixedKey by recording data when users use it. Mainly including several kinds of data as showed below:

a) The success rate of public and private authentication.
b) The time consumed by authenticating each password element.

In order to analyse the performance of the system, we require participants to register as legitimate users at first. Then we require users to log in both in public and private authentication ten times. We record the data mentioned above, to evaluate the performance.

We installed the application on the participant's device. The operating system version and the size of the screen vary with the users' devices.

We recruit 20 students who are not familiar with our system as our participants and all of them are university students.

5.2. Procedure of Experiment

1st phase: Participants need to attend training before conducting the experiment. The training mainly includes explaining the basic ideas and purposes of experiment on this system, teaching participants operate our system and creating an account. In order to guarantee the security in registration phase and the randomness of password space, we inform participants some advice during the training phase:

a) They should ensure privacy in registration phase.
b) They need to verify the accounts after setup is complete to avoid mistakes.
c) Because the entire ElementArray can be moved, it is not necessary to move the password character during the verification process, it is recommended that users try not to move those password elements which represent password.

2nd phase: We will give the registered users a period of practice to repeat two authentication schemes until they are familiar with the operation of the system. The data at this stage will not be collected in the analysis of the login phase.

3rd phase: After the user finishes practicing, the login phase is performed, participants are required to login in two modes respectively.10 times in private mode are required.

4th phase: After all login sessions are completed, we will let participants perform shoulder surfing attack and smudge attacks. Each participant randomly picked two login sessions of three ElementArray to be cracked, six videos of three password character in total. Each attacker has ten chances to crack it.

6. COLLECTED RESULTS

The main kinds of data evaluated in our experiment are accuracy and usability. We describe the experimental results in detail.

First of all, we calculate two kinds of accuracy rate: First Accuracy Rate and Total Accuracy Rate. The definitions are showed as below:

$$First\ Accuracy\ Rate = \frac{first\ successful\ attempts}{first\ attempts} \quad (1)$$

$$Total\ Accuracy\ Rate = \frac{successful\ attempts}{total\ attempts} \quad (2)$$

Figure 14. Definitions of First Accuracy Rate and Total Accuracy Rate

Usability is measured by calculating the time it takes to verify each password element.

6.1. Accuracy

Table 1 shows the First Accuracy Rate and the Total Accuracy Rate of the in login sessions with our 20 participants.

Table 1. First Accuracy and Total Accuracy in experiment.

	First Accuracy Rate	Total Accuracy Rate
Short	85.0%	82.5%
Medium	85.0%	81.6%
Long	80.0%	82.9%

On average, all but two participants prefer to use short passwords in the login phase. It shows that the First Accuracy Rate are higher than Total Accuracy Rate both in the short length password and the medium length password authentication. In the first session, 17 of 20 participants were able to log into the system successfully in the authentication of short password (in the length of 1-3) and medium password (in the length of 4-6) with just one try and the first success rate of long password (the length more than 6) authentication is 80 percent. After more than five tries, the accuracy of short password and medium password authentication decreased a little. However the accuracy of the long password authentication increased. We surveyed the participants for the possible reasons of the changes in accuracy rate. First of all, the random arrangement of password elements make it difficult for users to perform the public authentication several times continuously. Moreover, forgetting the indicator and submitting without checking also lead to failure of authentication.

6.2. Usability

To measure the usability of our system, we counted the time of authenticating each password element. Time consumed in the authentication is showed in Table 2. The average time each user spent on short, medium and long password authentication are 17.23 seconds,37.10 seconds and 52.01 seconds. The time consumed in each element during different length of password are 8.43 seconds, 8.23 seconds and 8.29 seconds. After practicing for several times, all the required time decreased, Figure 15 shows average time of each password element in the last five authentications. The result is 7.75 seconds, 7.84 seconds and 7.92 seconds. The change in the average time is subtle with the length increasing. This is because participants need to recall their passwords and elements. Based on the response of our participants, spending a little bit extra time is worthwhile if the authentication system can protect users' passwords from being pried by others who stand by them.

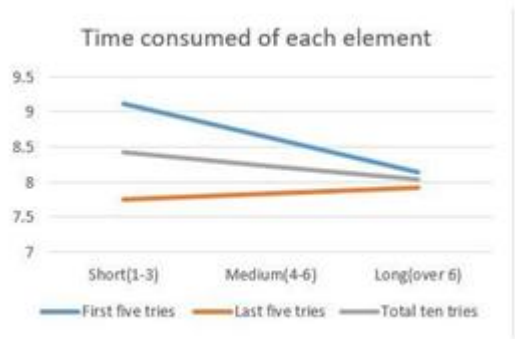

Figure 15. Average time consumed on first five, last five and all ten tries

Table 2. Average time consumed on each password element and whole authentication.

Average time (second)	Short Password (length 1-3)	Medium Password (length 4-6)	Long Password (length over 6)
Each Element	8.43	8.23	8.03
Whole Authentication	17.23	37.10	52.01

6.3. Cracking MixedKey

In order to examine the security level of our scheme against shoulder surfing attacks, we recorded the videos of two login sessions for participants to crack. Each participant was provided with six videos of authenticating three password characters randomly.

From the answer sheet returned, we found that no one was able to get the password successfully. For each password element, they couldn't figure out the location. The two login records didn't help them crack easily because the two login records are independent and the indicators are different from each other. Everyone has five chances to guess the password from two videos. As the result showed in Table 3, none of these 100 guessing tries cracked the account successfully. One of them could crack two password elements of three. 32 cases guessed correct on one element, and others failed to guess anyone correctly.

Table 3. Results of Cracking HybridPass.

		Number of cases
Success	3 Correct	0
Fail	2 Correct	1
Fail	1 Correct	32
Fail	0 Correct	67

7. SECURITY ANALYSIS

In this section we discuss about the performance when facing some mainstream attacks such as Random Guess Attack, Smudge Attack and Shoulder Surfing Attack.

7.1. Random Guess Attack

Because the size of ElementArray is M*N, in order to launch a random guess attack, the attacker needs to guess password element for the password character randomly, then move the password element to the M*N positions. As a result, guess one character of password has $(M*N)^2$ cases, so that the possibility of successful guess is $1/(M*N)^2$. If the user set a password with n characters. The possibility would decrease to $1/(M*N)^{2n}$. For example, in our experiment, we generated ElementArray in the form of 8*8 to arrange 64 password elements. If the password is 'admin', the length is 5, the possibility of being attack is $1/(8*8)^{10}$.

7.2. Shoulder Surfing Attack

In order to launch shoulder surfing attack, videos of three login phases of one participant is provided to another. Because the video of the verification process includes both the process of the user's finger moving on the screen and their password information. So the attack we provide is a mixture attack includes shoulder surfing attack and smudge attack. If our system could defend this kind of attack, it also could defend the two attacks well respectively.

The MixedKey takes advantage of adding extra information to protect the authentication, by using a special way to hide the locations of indicators instead of showing on the screen directly. Since the ElementArray is filled with password elements and is circulative, the scheme would protect the authentication even the process is recorded by the attackers. What's more, indicator for every authentication varies so that everyone is independent. As a result, no password element could be extracted in an authentication. As a result, MixedKey has a good performance of

defending shoulder surfing attacks as well as smudge attacks, even the attackers are equipped with video recorders.

8. CONCLUSION

8.1. Discussion

Although the private authentication mode provides different length of passwords for user to login, which outstands the existing schemes, we still think it is not easy for users to perform. According to the result of questionnaires in our experiments, users expressed that they prefer to use short passwords which only consist less than three letters after considering the tradeoff between usability and security. We think in the future, how to make the authentication easier to perform is the key improvement.

Similar to other graphical password authentication systems, MixedKey is vulnerable to random guess attack based on hot-spot analysing [25], [26]. This can be mitigate by users move innocent password elements from the initial location to the end.

Compared with other graph based authentication schemes , MixedKey is strong enough to resist shoulder surfing attacks, even if the attacks are camera-equipped. And the PIN-entry method [11], FakePointer [4], Wiedenbeck's scheme [3]do not have enough computational complexity while facing to random guess attack, which MixedKey can withstand.

8.2. Conclusion

In order to resist the shoulder surfing attack, a number of authentication schemes based on graph have been proposed. However, graph based password is totally different from text based password. It's difficult for users to memorize two different kinds of passwords. In this paper, we propose a hybrid authentication scheme called MixedKey which mixes graphic and traditional textual password. The login indicator in the scheme is randomly and safely generated for each login sessions, which prevents attackers from getting sensitive information when they perform shoulder surfing attacks. MixedKey connects graph and text based password. Users could login our system in both public and private situations with just one password. We also implemented MixedKey and conducted experiments to measure the memorability and usability. Our theoretical analysis and empirical experiments show that our scheme can resist the shoulder surfing attack and smudge attack as well.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant No. (61402225, 61373015, 41301407), the National Natural Science Foundation of Jiangsu Province under Grant No.BK20140832, the China Postdoctoral Science Foundation under Grant No.2013M540447,the Jiangsu Postdoctoral Science Foundation under Grant No.1301020C, State Key Laboratory for smart grid protection and operation control Foundation, Science and Technology Funds from National Electric Net Ltd.(The Research on Key Technologies of Distributed Parallel Database Storage and Processing based on Big Data), the Foundation of Graduate Innovation Center in NUAA under Grant No.kfjj20181608.

REFERENCES

[1] Wikipedia, "Shoulder surfing (computer security)." https://en.wikipedia.org/wiki/ Shoulder_surfing_(computer_security). Accessed Dec 19, 2018.

[2] R.N. police, “Shopkeeper uses surveillance camera to unlock lost mobile phone and steal customers’ money..” https://baijiahao.baidu.com/s?id=1612024658890034305&wfr= spider&for=pc. Accessed Sep 19, 2018.

[3] S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskiy, and N. Memon, “Passpoints: Design and longitudinal evaluation of a graphical password system,” International journal of human computer studies, vol.63, no.1-2, pp.102–127, 2005.

[4] T. Takada, “fakepointer: An authentication scheme for improving security against peeping attacks using video cameras,” 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp.395–400, IEEE, 2008

[5] H.M. Sun, S.T. Chen, J.H. Yeh, and C.Y. Cheng, “A shoulder surfing resistant graphical authentication system,” IEEE Transactions on Dependable and Secure Computing, vol.15, no.2, pp.180–193, 2018.

[6] I. Jermyn, A. Mayer, F. Monrose, M.K. Reiter, and A.D. Rubin, “The design and analysis of graphical passwords.,” USENIX Association, 1999.

[7] M. Martinez-Diaz, J. Fierrez, and J. Galbally, “The doodb graphical password database: Data analysis and benchmark results,” IEEE Access, vol.1, pp.596–605, 2013.

[8] M. Martinez-Diaz, J. Fierrez, and J. Galbally, “Graphical password-based user authentication with free-form doodles,” IEEE Transactions on Human-Machine Systems, vol.46, no.4, pp.607– 614, 2016.

[9] Y. Song, Z. Cai, and Z.L. Zhang, “Multi-touch authentication using hand geometry and behavioral information,” 2017 IEEE Symposium on Security and Privacy (SP), pp.357–372, IEEE, 2017.

[10] Z. Ling, J. Luo, Y. Liu, M. Yang, K. Wu, and X. Fu, “Sectap: Secure back of device input system for mobile devices,” IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp.1520–1528, IEEE, 2018.

[11] V. Roth, K. Richter, and R. Freidinger, “A pin-entry method resilient against shoulder surfing,” Proceedings of the 11th ACM conference on Computer and communications security, pp.236– 245, ACM, 2004.

[12] A.J. Aviv, K.L. Gibson, E. Mossop, M. Blaze, and J.M. Smith, “Smudge attacks on smartphone touch screens.,” Woot, vol.10, pp.1–7, 2010.

[13] L. Wu, X. Du, and X. Fu, “Security threats to mobile multimedia applications: Camera-based attacks on mobile phones,” IEEE Communications Magazine, vol.52, no.3, pp.80–87, 2014.

[14] G. Ye, Z. Tang, D. Fang, X. Chen, K.I. Kim, B. Taylor, and Z. Wang, “Cracking android pattern lock in five attempts,” 2017.

[15] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz,“A pilot study on the security of pattern screen-lock methods and soft side channel attacks,” in Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2013.

[16] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, “Smudge Attacks on Smartphone Touch Screens,” in Proceedings of the 4th USENIX Workshop on Offensive Technologies, 2010.

[17] S. Cha, S. Kwag, H. Kim, and J. H. Huh, “Boosting the Guessing Attack Performance on Android Lock Patterns with Smudge Attacks,” in Proceedings of the 12nd ACM Asia Conference on Computer and Communications Security, 2017.

[18] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, “Practicality of accelerometer side channels on smartphones,” in Proceedings of the 28th ACM Annual Computer Security Applications Conference, 2012.

[19] E. von Zezschwitz, A. De Luca, P. Janssen, and H. Hussmann, “Easy to Draw, but Hard to Trace? On the Observability of Grid-based (Un)Lock Patterns,” in Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems, 2015.

[20] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan,“Shoulder Surfing Defence for Recall-based Graphical Passwords,” in Proceedings of the 7th ACM Symposium on Usable Privacy and Security, 2011.

[21] A. J. Aviv, D. Budzitowski, and R. Kuber, “Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android’s Pattern Unlock,” in Proceedings of the 31st ACM Annual Computer Security Applications Conference, 2015.

[22] Y. Song, G. Cho, S. Oh, H. Kim, and J. H. Huh,“On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks,” in Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems, 2015.

[23] S. Uellenbeck, M. Durmuth, C. Wolf, and T.Holz, "Quantifying the security of graphical passwords: the case of android unlock patterns," in Proceedings of the 20th ACM Conference on Computer and Communications Security, 2013.
[24] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. Nicholson, J. Nicholson, and P. Olivier, "Multi-touch authentication on tabletops," in, Proceedings of the 28th international conference on Human factors in computing systems. ACM, 2010, pp. 1093–1102.
[25] J. Thorpe and P. van Oorschot, "Human-seeded attacks and exploiting hot-spots in graphical passwords," in Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium. USENIX Association, 2007, p. 8.
[26] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon,"Authentication using graphical passwords: effects of tolerance and image choice," in Proceedings of the 2005 symposium on Usable privacy and security. ACM, 2005, pp. 1–12.
[27] Z. Zheng, X. Liu, L. Yin, and Z. Liu, "A stroke-based textual password authentication scheme," in Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on, vol. 3. IEEE, 2009, pp. 90–95.
[28] X. Bai, W. Gu, S. Chellappan, X. Wang, D. Xuan, and B. Ma, "Pas: predicate-based authentication services against powerful passive adversaries," in 2008 Annual Computer Security Applications Conference. IEEE, 2008, pp. 433–442.

AUTHORS

Jie Wan received his Bachelor's degree in 2018, from the Nanhang Jincheng College, Nanjing, China. He is currently a master student in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include, System Security and UAV Security.

Liang Liu is currently a Lecturer in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China. His research interests include distributed computing, big data and system security. He received the B.S. degree in computer science from North western Polytechnical University, Xi'an, Shanxi Province, China in 2005, and the Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, China in2012.

Dai Hua is currently an associate professor in the School of Computer Science, Nanjing University of Post &Telecommunications, Nanjing, Jiangsu Province, China. His research interests include data security, database and system security.

Wei Liu is currently working in Nanjing Institute of technicians,