Noisy Password Scheme: A New One Time Password System

HANAN A. MAHMOUD
hmahmoud@ksu.edu.sa

King Saud University, Riyadh,
Kingdom of Saudi Arabia

Abstract. The purpose of a one-time password (OTP) is to make it more difficult to gain unauthorized access to restricted resources, like a computer account. Traditionally static passwords can more easily be accessed by an unauthorized intruder given enough attempts and time. By constantly altering the password, as is done with a one-time password, this risk can be greatly reduced. In this paper, we propose the new noisy password technique. The proposed system attempts to alleviate the problem of shoulder surfing or eves dropping by making the replay of a password useless. Every time a user is authenticated by totally different password. The noisy password constitute of several parts, the actual password and additional noisy parts that are well studied to generate different passwords almost every time a user wants to authenticate himself. The noisy parts are proven to be robust against any hacking attacks. Experimental results give good indication of the ease of utilization of the new system with low error rates that can be enhanced by time.

Key-Words: - One-time password, Static passwords, Security, Shoulder surfing.

I. Introduction

Most current commercial websites will ask their users to input their user identifications (IDs) and the corresponding Passwords for authentication. Once a user's ID and the corresponding password are stolen by an adversary, the Adversary can do anything with the victim's account, leading to a disaster for the victim. The secure protocol SSL/TLS [1] for transmitting private data over the web is well-known in academic research, but most current commercial websites still rely on the relatively weak protection mechanism of user authentications via plaintext password and user ID. Meanwhile, even though a password can be transferred via a secure channel, this authentication approach is still vulnerable to attacks. Phishers attempt to fraudulently acquire sensitive information, such as passwords and credit card details, by masquerading as a trustworthy person or business in an electronic communication. Password Stealing Trojan is a program that contains or installs malicious code. There are many such Trojan codes that have been found online today [10]. Key loggers capture keystrokes and store them somewhere in the machine, or send them back to the adversary. Shoulder surfing is a well-known method of stealing other's passwords and other sensitive personal information by looking over victims' shoulders while they are sitting in front of terminals or in front of an ATM machine [2]. This attack is most likely to occur in insecure and crowded public environments, such as an Internet Café, shopping mall, airport, etc. [3]. It is possible for an attacker to use a hidden camera to record all keyboard actions of a user. Video of the user's actions on a keyboard can be studied later to figure out a user's password and ID. "Shoulder surfing" or "peeping attacks" refers to stealing information (especially authentication information) by looking over the shoulder of an unsuspecting user. Shoulder surfing is defined as the unauthorized observing of an authorized user's session on an electronic device in order to gain access to information. Historically, shoulder surfing concerns moved from telephone calling card fraud to automated teller machine (ATM) fraud, and more recently to mobile computer users. Mobile computing users in a public place cannot be aware of all the activity in their surroundings, and are vulnerable to persons visually observing or even recording their authentication session, with the intent of extracting login information. Similar considerations apply to access points (security doors) where the user is expected to authenticate by entering a PIN code. Most authentication methods involve pressing keys on a keyboard or selecting objects on a screen, and both the screen and the keyboard are visible to the authorized user as well as to the shoulder surfer [4], [5].

This paper presents a new password choice technique. The new technique will be proven secure against shoulder surfing and other form of attacks. This paper is divided into sections. In section 2, an overview of authentication methods are presented along with literature survey. The new technique is presented in section 3.

2. Overview of the Authentication Methods

Current authentication methods can be divided into three main areas:

- Token based authentication
- Biometric based authentication
- Knowledge based authentication

Token based techniques, such as smart cards are widely used. Token-based authentication systems use knowledge based techniques to enhance security. For example, ATM cards are used with a PIN number. Biometric based authentication techniques, such as fingerprints, iris scan, or facial recognition, are not yet widely adopted. The major drawback of this approach is that such systems can be expensive, and the identification process can be slow and often unreliable. However, this type of technique provides the highest level of security. Knowledge based techniques are the most widely used authentication techniques and include both text-based and picture-based passwords. The picture-based techniques can be further divided into two categories namely recognition-based and recall-based graphical techniques. In those techniques, a user is presented with a set of images and the authentication is performed by recognizing and identifying the images he selected previously. Using recall-based techniques, a user is asked to reproduce a figure that was selected earlier during the registration stage.

A graphical authentication scheme based on the Hash Visualization technique is proposed [4], [11] where, the user is asked to select a certain number of images from a set of random pictures generated by a program. The authentication phase is carried by identifying the preselected images. The most disadvantageous feature of such a system is that the average log-in time is longer than the traditional approach. A weakness of this system is that the server needs to store the seeds of the portfolio images of each user. Also, the process of the user selecting a set of pictures from the picture database can be tedious and time consuming. Akula and Devisetty's algorithm [5] is similar to the technique proposed by Dhamija and Perrig [4]. The difference is that they used hash function SHA-1, which produces a 20 byte output, which means less memory is required. Other password schemes sketched several authentication schemes, such as picture recognition, object recognition, and pseudo word recognition [6]. In the picture recognition study, a user is trained to recognize a large set of images (100 – 200 images) selected from a database of 20,000 images. After one to three months, users in their study were able to recognize over 90% of the images in the training set. This study showed that pictures are the most effective among the three schemes tested. Pseudo codes can also be used, but require proper setting and training [6]. The authors in [7] adopt a game-like graphical method of authentication to combat shoulder-surfing; it requires the user to pick out the passwords from hundreds of pictures, and then complete rounds of mouse clicking in the Convex Hull. However, the whole process needs the help of a mouse and it takes a long time. In [8], the authors propose a scheme to ask a user to answer multiple questions for each digit. In this way, it is resistant to shoulder-surfing only to a limited degree, because if an adversary catches all the questions, then they will know what the password is. The author in [9] allows a user to make some calculations based on a system generated function and random number for the user to prevent password leaking. However, the scheme in [9] is not anti-Phishing and the password can possibly be stolen if an adversary uses a camera to record all the screens of the system and motions of the victim. Biometric identification techniques such as face scanners and particularly fingerprint readers are gaining in popularity, but some of these methods are still prone to false positive and false negative identification [1]. Some conditions, such as a cut finger, may result in further problems with biometric authentication.

3. One-Time Passwords

The purpose of a one-time password (OTP) is to make it more difficult to gain unauthorized access to restricted resources, like a computer account. Traditionally static passwords can more easily be accessed by an unauthorized intruder given enough attempts and time. By constantly altering the password, as is done with a one-time password, this risk can be greatly reduced. There are basically three types of one-time passwords: the first type uses a mathematical algorithm to generate a new password based on the previous, a second type that is based on time-synchronization between the authentication server and the client providing the password, and a third type that is again using a mathematical algorithm, but the new password is based on a challenge (e.g. a random number chosen by the authentication server or transaction details) and a counter instead of being based on the previous password. One approach, credited to Leslie Lamport, uses a one-way function (call it f). The one-time password system works by starting with an initial seed s, then generating passwords f(s), f(f(s)), f(f(f(s))), ... as many times as necessary. If an indefinite series of passwords is wanted, a new seed value can be chosen after the set for s is exhausted. Each password is then dispensed in reverse, with f(f(...f(s))...) first, to f(s). If a shoulder surfer sees the password, he will have access for one time period for login, but it becomes useless once that period expires. The time-synchronized one-time passwords are usually related to physical hardware tokens. Inside the token is an accurate clock that has been synchronized with the clock on the authentication server. On these OTP systems, time is an important part of the password algorithm since the generation of new passwords is based on the current time rather than the previous password or a secret key. For a large installation, time-synchronized OTPs is more expensive choice, as any additional cost to the non-time-synchronized server should be outweighed by the more expensive and less durable individual tokens. One-time passwords that are not time-synchronized are also vulnerable to phishing. In late 2005 customers of a Swedish bank were tricked into giving up their one-time passwords. However, even time-synchronized one-time passwords are vulnerable to phishing, if the password is used quickly enough by the attacker.

4. Noisy Password Technique as a Onetime Password Technique

In this paper, we are introducing a new technique for one time password by adopting the noisy password technique. The noisy password technique is proposed in subsection 4.1. Experimental results are presented in subsection 4.2.

4.1 First Algorithm Noisy Passwords with Terminators

To authenticate a user, a system (S) needs to verify a user (U) via the user's password (P) which the user provides. It is very reasonable that a password should be constant for the purpose of easily remembering it. However, the price of easily remembrance is a password theft. At the same time, we cannot put P in a randomly variant form, which will make it impossible for a user to remember the password. To confront such a challenge, we propose a scheme using a new concept of password we named noisy password. A noisy password is a password that contains the actual password embedded in it. It cannot be applied directly but instead a software extracts the actual password from it and generates the password which is submitted to the server for authentication, or compared to the one stored on the smart card. A noisy password P is defined as a quadruplets. It is defined with four parts, a fixed alphanumeric F and a variable alphanumeric V, a terminator X and a safeguard S. Therefore, a password will be defined as an ordered quadruplets as follows,

$$P = < T_S, V, X, F > \qquad (1)$$

F will be defined by the user and it is be a variable length alphanumeric text from 4 to 8 alphanumeric characters. This part is fixed and it represents the actual password of the user. It is defined as an ordered set as illustrated in equation 2.

$$F = \{ f_1, f_2, f_3, ..., f_i \quad 4 \le i \le 8\} \quad (2)$$

V is the variable part and it is text of alphanumeric characters that users enter between each character of F until they hit a terminator. Elements of X is a chosen one character and it acts as a terminator. X is an ordered set as illustrated in equation 3. V is characterized by being of variable length text and is defined in equation 4. Elements of V are faced with some restrictions as shown in equation 5. S is a safeguard number, where a user will be requested to embed an alphanumeric text T_s with length S after each $f_i \in F$. T_s is defined in equation 6. A password p may be defined from equations 1 to 6 as presented in equation 7.

$$X = \{x_1, x_2, ..., x_i \quad 4 \le i \le 8 \ \} \quad (3)$$

$$V_j = \{v_1, v_2, v_3, ..., v_i \quad 0 \le i \le n_j\} \quad (4)$$

$$(v_k \in V_i) \ne (x_i \in X) \qquad \forall i \qquad (5)$$

$$T_S^g = \{t_1^g, t_2^g \ldots\ldots t_S^g \quad 6 \le g \le 10\} \qquad (6)$$

$$p \in P = \{(T_S^1, V_1, x_1, f_1), \ldots\ldots (T_S^i, V_i, x_i, f_i), V_{i+1} \quad 4 \le i \le 8\} \quad (7)$$

There is nothing will be prompted in the login screen by the server for extra security. The user input includes (ID, P), where ID is user ID. On the server side, the server will calculate F to compare it with the stored password in the smart card or in the database. The user should be free to pick the all parts of the noisy password. We propose a differentiated security mechanism in the next section to restrict the system to choose the noisy part without interference from the user.

The concept of the noisy password is to add the variable part as a noise that the user can change it every time he/she makes a password entry so no need to memorize this part. The user needs only to memorize the terminators. The terminator is a set of characters that act as terminator of subset of the variable part and start of a subset of the fixed part of the password. A user has to remember both the fixed part and the terminator part, and as a result will require a little bit more effort to remember. However, the noisy password will be resistant to a dictionary attack, which is mostly caused by the fact that users like to create a password which is either related to their own name, date of birth, other simple words, etc. In a traditional password scheme, users can change their password, and this is also true in our noisy password scheme. Different from the traditional scheme, users can change the fixed part of the fixed part password or the terminator, or even both. Also, the user will enter a different string (of different length) every time a password is required, which makes the password more robust to shoulder surfing or eavesdropping because it is very difficult to deduce the fixed part from the noisy part.

Example 1:
Password parts are: F, T_S, V, X
The following is a numeric example:
F =1,4,5,2
S = 5
X = 3,6,7,6
V = Any subset of the alphanumeric set, not specified at time of user registration, some conditions apply.

The user will input a variable length password, a valid one would be:
234341244209080206077799931234344377889145642
343423242323231212133121313131317523434123578
904323232323232323232323232323545554544334554322
34562123456789171661617188888888888888888861132
312457879797987987965426514253652363276767676
768887878989898989123 Enter
The system will look for the terminators in the string as follows (the terminators are colored in red)

2343412442090802060777999***3***12343443778891456***6***42
34342324232323121213312131313131***7***523434123578
904323232323232323232323232323545554544334554322
345***6***223434123456789171661617188888888888888888
611323124578797979879879654265142536523632767
676767688878789898989123 Enter

The system then extracts the password which are located after the terminator characters as follows (the fixed part of the password is colored in blue)

12442090802060777999***31***4377889145**64**23242323231
21213312131313131***75***12357890432323232323232323
23232323545554544334554322345***62***123456789171661
617188888888888888888861132312457879797987987 96
5426514253652363276767676767688878789898989123
Enter
For password changing, it is similar to traditional password changing. The user can choose a new password, which is the fixed part of the noisy password and/or a new terminator and/or a new safeguard number. After such changes, the user needs to remember the new password.

Two algorithms are presented, the first algorithm is for choosing the password and storing it in a database. the second algorithm is for extracting the password and recognize it. For more security issues the user will be allowed limited number of incorrect password entry. Also for more security, history of correct password entries will be stored in the database. If a shoulder surfer recorded a noisy password and attempts to use it, the system will compare it to the database and in case of a match the system will ask for reentry where the perpetrator has to enter another noisy password, which he will have no knowledge about it. This is an advantage of the noisy part, where users are advised to randomly choose it every time.

Algorithm Noisy-Password-Choice(ID, F, X, S)

```
{
The system prompts the user for entry of ID, F, X and S;
The user enters ID, F, X, S;
The system stores this information along with user ID in the database and a counter COUNT of number of times the user used the system;
}

Algorithm Noisy-Password-Extract(ID, p)
{
Number-of-trails=0;
START:
If (number-of-trails< 4)
{
The system prompts the user for entry of ID, p;
The user enters ID, p;
Read the user information F, X ,S, COUNT  from the database using ID as a key;
k=1;
Repeat Until end-of-string (p)
{
        For (i= 1 to S)
                Read-character(p);
        Z=’’
        Repeat Until (Z=X(k))
                Z=Read-character(p);
        f1(k)=Read-character(p);
        k++;
}
If (COUNT =1)
Then
 {
        {If (F=F1)
        Then
                { Extract all $T_S$ from p;
                 Store $T_S$ in the database;
                Accept the password;
                COUNT++};
        Else
                {
                Do not accept the password;
                Number-of-trials++;
                Go to START;
                }

}
Else
{
        Extract $T_S$ from p;
        Compare the $T_S$ with all $T_S$s  in the database;
        If (The comparison yields a positive comparison)
        Then
        {
        Do not accept password;
        Number-of-trials++;
        Go to START;
        }
}
}
```

5. Experimental Results

Twelve subjects, ranging from 35 years old to 40 years old, participated in the experiment. Eight were female and four were male. The mean age of participants was 37.7 (SD=1.33). Most of the participants graduated from college.

They all used PCs frequently. The noisy password system, used in this study, has a simple interface where four fields for ID and password choice, first field is for user ID, the other three fields are for F, X and S values. The interface for entering id and database is another simple interface with two fields one for Id and one for the password. All instructions for the participants were described to them. Feedback on correctness of a password input was given on screen after the user clicked the Submit button. The study was carried out in two sessions. Each participant was sat at a laptop. In the experiment, which lasted about 30 minutes, the participants first chose an ID and a password. When the participant had chosen a password, a valid noisy password was displayed as an example to the participant. The display showed the noisy password with a heavy outline of each part of the password. When the participant had created a valid password, the learning phase began. The participant entered the noisy password repeatedly until he or she achieved ten correct password inputs. Participants received binary feedback on the correctness of each password input and could see an on-screen count of how many correct and incorrect entries they had made. In the retention phase password retention was measured at the end of the first session (S1) and three days later (S2). The participant had to enter the password correctly one time. The trial was over as soon as the participant entered a correct password. If the participant entered an incorrect password, the system gave feedback that the password was wrong, and the participant was instructed to reenter the password. If the user failed to input the password correctly after five attempts, the participant was allowed to view a correct password, and then make another attempt to input it.

5.1 Results

Results about the two phases of the study were recorded: password learning, and retention, as discussed below. In the learning session participants entered their password repeatedly until they had accomplished 10 correct inputs. We measured the number of attempts to meet the criterion and the time. The means and standard deviations are show in Table 1. Further experiments about the number of individuals who entered their password 10 times without any errors in the learning phase are done – 3 subjects were able to do that. However, 4 out of 12 individuals succeeded with only one or two extra attempts. The rest of the subjects took many more trials (Table 2). Tables 3 and 4 show the means and standard deviations of the retention phase. S1 is the password input at the end of the first session. S2 is the session three days later. Participants had only to enter their password correctly one time in the retention trials. The number of incorrect submissions showed that the effect of retention trial was significant with a higher number of incorrect attempts in S2. In terms of individual participants, our data show that in S1 there were two participants that made errors submitting their password. By contrast, in S2 6 participants made at least one error. Furthermore, we examined how many participants failed to log in by the criterion of making a correct log in within 4 or less attempts. This criterion was chosen because existing password systems often block users if they make repeated errors on input. Only 2 of 12 failed. Group in S2 took significantly more time on incorrect attempts.

Table 1. Means (SD) in learning phase

Number of incorrect submissions	1.56 (3.65)
Time for incorrect submissions (sec)	116.08 (78.52)
Time for correct submissions (sec)	33.52 (4.35)

Table 2. Number of participants making incorrect submissions in the learning phase

Number of incorrect submission	0	1	2	3	4	5	6	7	> 7
Number of subjects	3	2	2	1	0	0	2	1	1

Table 3. Means (SD) in S1 retention trial

Number of incorrect submissions	0.27 (0.65)
Time for incorrect submissions (sec)	26.08 (7.52)
Time for correct submissions (sec)	22.52 (8.35)

Table 4. Means (SD) in S2 retention trial

Number of incorrect submissions	0.56 (1.65)
Time for incorrect submissions (sec)	31.08 (9.5)
Time for correct submissions (sec)	24.52 (6. 5)

5.2 Discussion

Participants had little difficulty learning their password via repeated password inputs, while it posed challenges to some. Another indicator of this trend is the long trail of participants who took many practice trials (Table 2). Using noisy password schemes was new to the participants and we expected errors in the learning phase. In the S1 retention trial there were very few incorrect password submissions. It should be noted that the mean time for these incorrect submissions was very low. This likely indicates that the errors were slips, in which the participants noticed a slip immediately and submitted it so they could start over. In the correct submissions the time was approximately, 22 to 35 seconds. In other studies, the time for inputting alphanumeric passwords was between 10 and 11 seconds. We found it encouraging that after a little practice the difference was few seconds. Generally, we expect slower input times in noisy password systems. On the other hand, slower noisy password input in our studies may also be related to the participants' lack of experience. We expect users to input noisy passwords faster with continual use and automation of the process. We are currently carrying out a study of repetitive use of our noisy passwords to evaluate how fast users can enter a correct noisy password when they have become very well practiced with their password. Even if the input time for noisy passwords is substantially higher than alphanumeric passwords in normal use, then these passwords will be suitable only for authentication needs where security is more important than time. The results of the S2 retention trial were strikingly different from S1. The main issue was the participants' ability to remember their noisy password. First, it should be noted that there were relatively few errors, only 12 in total for a mean of slightly less than 1 per person. However, 4 participants had no errors at all, and most of the errors were from two people, one who needed 4 attempts to be successful and the other who needed 8 attempts. Only one of the participants took more than 4 attempts, our criterion for failure. The time for incorrect submissions was not highly elevated over S1, suggesting that participants knew what to do for their password entry quickly, even though they made errors. The longer mean time for correct password submissions in S2 than in S1 may mean that individuals who made an error in their first attempt(s) were slower and more cautious in their subsequent correct attempt.

6. Conclusion

Most network applications authenticate users with an account-name/password system. The most common computer authentication method is to use alphanumerical usernames and passwords. Recently graphical password systems raise more attention to them. Both methods have been shown to have significant drawbacks. For example, users tend to pick alphanumeric passwords that can be easily guessed. On the other hand, graphical password is susceptible to shoulder surfing and is hard to remember. To address this problem, some researchers have developed other authentication methods that use eye gaze as a password entry technique. Such technique suffers from high error rates. In this paper, we proposed the new noisy password technique. The proposed system alleviated the problem of shoulder surfing or eves dropping by making the replay of a password is of no use. Every time a user is authenticated by totally different password. The noisy password is constituted of several parts, the actual password and additional noisy parts that are well studied to generate different passwords almost every time a user wants to authenticate himself. The noisy parts are proven to be robust against any hacking attacks. More experimentation is to take place in future studies of our system.

References

[1] L. Vasiu and I. Vasiu, "Dissecting Computer Fraud: From Definitional Issues to a Taxonomy," presented at 37th Hawaii International Confernces on System Sciences, Hawaii, 2004.

[2] J. Goldberg, J. Hagman, and V. Sazawal, "Doodling Our Way to Better Authentication," presented at Proceedings of Human Factors in Computing Systems (CHI), Minneapolis, Minnesota, USA, 2002.

[3] L. D. Paulson, "Taking a Graphical Approach to the Password," *Computer*, vol. 35, pp. 19, 2002.

[4] R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication," in *Proceedings of 9th USENIX Security Symposium*, 2000.

[5] S. Akula, V. Devisetty, "Image based registration and authentication system," *Midwest Instruction and Computing Symposium* (2004).

[6] D. Weinshall and S. Kirkpatrick, "Passwords You'll Never Forget, but Can't Recall," in *Proceedings of Conference on Human Factors in Computing Systems (CHI)*. Vienna, Austria: ACM, 2004, pp. 1399-1402.

[7] W. A. Jansen, "Authenticating Users on Handheld Devices," in *Proceedings of Canadian Information Technology Security Symposium*, 2003.

[8] T. Takada and H. Koike, "Awase-E: Image-based Authentication for Mobile Phones using User's Favorite Images," in *Human-Computer Interaction with Mobile Devices and Services*, vol. 2795 / 2003: Springer-Verlag GmbH, 2003, pp. pp. 347 - 351.

[9] J. Thorpe and P. C. v. Oorschot, "Towards Secure Design Choices for Implementing Graphical Passwords," in Proceedings of the 20th Annual Computer Security Applications Conference. Tucson, Arizona, 2004.

[10] B. Paulson and T. Hammond, "Paleo Sketch: Accurate Primitive Sketch Recognition and Beautification, " In Proc. of IUI, 2008.

[11] Y. Qiao, J. Liu, and X. Tang, "Offline signature verification using online handwriting registration, " In Proc. Of CVPR '07, pages 1–8, 2007.