Elastic Password

JEEVITHA C[1], Mrs.R. VIJAYALAKSHMI[2]
[2] Assoc.Prof.
[1, 2] Krishnasamy College of Engineering and Technology,
Cuddalore.

Abstract- *A New Mechanism for Strengthening Passwords using Time Delays between KeystrokesText passwords was and still is an easy, common, practical and reliable authentication method. To ensure the desired goal of access protection, passwords should be implemented carefully. This study proposes a new mechanism to strengthen passwords through introducing time gaps (delays) between some password characters during the sign-up process. During sign-in, users should not only enter their correct passwords, but also they must leave time delays in the correct position(s) as it was designated during the sign-up process. Any attempt to login using a real password without adding time delays in the correct position(s) will be rejected. Laboratory experiments were conducted to test the proposed mechanism.These experiments are based on known password but hidden delay position(s). The obtained results show that the FalseAcceptance Rate (FAR) was 11.33% and the False Rejection Rate (FRR) was 17.3%. These ratios could be improved when users became more familiar with the system.*

Keywords- user authentication; Information Security; password-based Systems

I. INTRODUCTION

Individuals use passwords as a defense technique to protect what they think is valuable, like jewelry and money. A simple example from human ancient cultures denotes the existence of password is “Open Sesame”; the secret of a thieves' cave that was mentioned in a well-known popular story: Ali Baba and the Forty Thieves, a story of the "Arabian Nights" tales. Even though passwords are the most common mechanism to authenticate users, it suffers from a set of vulnerabilities that make it easy to break . For example, brute force and dictionary attacks, in which intruders attempt every possible combination of letters and any available information about the user in order to guess the password . Moreover, passwords can be stolen using shoulder surfing, key-loggers or by using social engineering methods which is an attempt by an intruder to elicit password and account information from a user. On the other side, designers have responded with several methods to counter these types of attacks such as: enforcing rules about the length of the password, the diversity of the characters that comprise it, automatic user lockout after failed attempts, and forcing password change periodically. Furthermore, user training and awareness can reduce many bad actions that users may do, such as choosing poor passwords, writing them down, sharing them with others, and giving information to strangers that have no need to know .Researchers proposed different methods to enhance the security of conventional password-based authentication systems. A lot of research has been going on in making passwords stronger. Some of these methods verifies not just the knowledge of the password, but also verifies existing another credentials such as the possession of a specific token (e.g. smart card), the current GPS location of the user, fingerprints or any other biometrics , or something user can do such as signature and keystroke dynamics . In this research, a new mechanism to strengthen passwords is proposed. The proposed method is based on adding time gaps between the password's characters. The selected positions are chosen by the user. In order to let the users access the system, the password along with the time gaps must be provided in a correct manner.

It should be clear that this approach is not the same as keystroke dynamics. Keystroke dynamics are timing information that measure individual manner and rhythm during typing on a keyboard. The system extracts the natural user behavior during typing using long training sessions. To authenticate the login attempts, users should write the correct password using his way of writing (i.e. rhythm) . In this proposed system, the user predefined his keystroke pattern rather than using his human patterns (rhythm). The rest of this paper is organized as follows. Section 2 a technical description of the proposed mechanism. In section 3, the conducted experiments and its results are presented. Security analysis of the proposed mechanism is given in section 4. Section 5 concludes the paper and introduces a potential future work.

II. PROPOSED MECHANISM TECHNICAL DESCRIPTION

In this proposed mechanism, a new form of a strong password is formulated. Basically, this password is based on augmenting time gaps (delay) between certain password characters during typing (i.e. elastic). The user locates these gap positions during the sign-up process. Thus, these positions form what we call "adopted keystroke pattern". In order to let a user access a certain system, the user should type the correct

password and leave relatively larger time gaps in the specified position than the time delays between the rest of the password characters. As an illustration, if the user specifies that he/she wants to add a time gap after the third character, then the system will not let users - or perhaps the adversary - access the system unless he/she type in the correct password and leave larger time delay after the third character in comparison with time delays after the rest of the characters. For example, Khaled decided to add a time gap after the second, fifth and the seventh character. The sign-up process is shown in the right hand side of Fig. 2. Generally speaking, large delay is considered large when compared with other normal delays after the remaining characters. The exact delay time value is not critical, the user should leave a delay time at the selected position that exceeds his/her normal typing rhythm. Any accidental waiting time during typing password will be considered as a large delay and will lead to reject login if this delay is in the wrong position. To bring this proposed mechanism to reality, a visual basic program was developed. This program consists of sign-up and sign-in processes. All user data are stored in a database that was built using Microsoft Access 2013. Thirty participants volunteered to do the experiments. They are considered as legitimate users in part of the experiments and are considered the imposter users in the other part. Certain metrics are used for evaluation such as false acceptance rate (FAR) and false rejection rate (FRR).

A. Sign-Up Process

A normal sign-up interface is provided, except for the extra feature that allows users to specify the time gap positions, see Fig. 1 below. The sign-up process goes as follows: the user initially types in his/her information: first name, last name, username and password. After that, the user selects the characters he/she wants to make an unusual delay after. For simplicity, we assume that the password length is 8 characters. A seven checkboxes are added to specify the position(s) of the time gap(s). Table I shows the password file structure that was used to store users' information. Note that, the designated positions of the specified delays are concatenated and stored in the last column.

Fig. 1. Sign-up interface.

For example, Khaled decided to add a time gap after the second, fifth and the seventh character. The sign-up process is shown in the right hand side of Fig. 2.

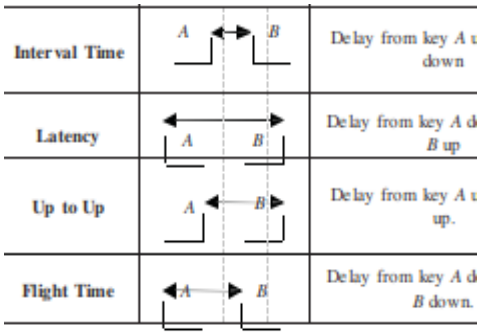
TABLE I. PASSWORD FILE STRUCTURE

	First Name	Last Name	User Name	Password	Gap Position
1	Khaled	Mahmoud	Khaled1969	abc#025x	257
2	Basel	Ali	Basel1970	Sunshine	37

B. Sign-In Process

During the sign-in process, a registered user enters his/her username and password with a large time delay(s) in the correct position(s) that was selected during the sign-up process. In order to check the authenticity of the login attempt, the system captures the delay time between each two successive characters. In the literature, different methods are used to calculate this delay [2]. Table II shows these different methods. Note that, pressing a key generates two basic timing events: key down (when the key is pressed) and key up (when the key is released). In this system, we use Interval Time in order to calculate the time delay. In other words, the time delay is the time elapsed from releasing the current character and pressing the next character. This time is measured in milliseconds. Username, password and a list of time delays are passed to a specific authentication module that is used to check the validity of the input parameters. More about this module is given next.

TABLE II. *FEATURES THAT CAN BE EXTRACTED FROM ANY TWO SUCCESSIVE KEYSTROKES [9].*



C. Authentication Module

First of all, the authentication module checks whether the given password is related to the given username or not; if not, then the login fails and the user is prompted to enter the username and password again. On the other hand, if there is a match between the username and the password, the module checks for the correct positions of the time gaps between password characters. If the user enters his password with a relatively large delay time in the correct position(s) then the login is successful, otherwise the login fails.

In order to check the time delays, a detailed algorithm is suggested that relies on sorting the time delays and finding the differences between the delays. In our case, each password has 7 time delays. The main steps in this algorithm are given next and its flowchart is shown in Fig. 2

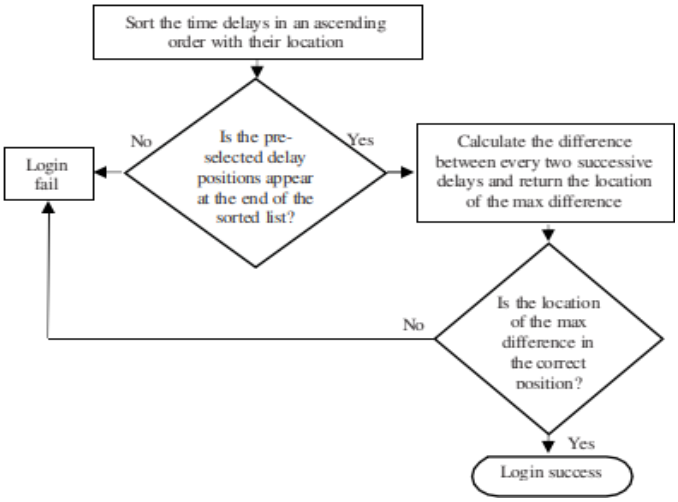


Fig. 2.Flowchart for time delay verification.

1. Sort the list of delays in an ascending order along with the original position of each delay value.
2. If the pre-selected delay locations didn't appear at the end of the sorted list (regardless of their order), then the login fails. This is because the user has left a large time delay(s) in other location(s) rather than the pre-specified positions at the sign-up phase. If the pre-specified positions appear at the end of the sorted list then this can be the first sign that the user has entered his password correctly.
3. In some cases, a user may leave more numbers of large delays than what was specified in sign-up (for example, the user may leave large delays after each character). For this reason, the algorithm computes the difference between every two successive delays in the sorted lit and returns the location of the maximum difference.
4. If the maximum difference is between the first large delay in the sorted list and its previous delay then the login succeeds, otherwise the login fails. For example: if the user selects three positions to add a large time delay then the maximum difference should be between the third delay from the end of the sorted list and the one before it. To clarify the previous steps, suppose that the user decides to add a large time delay after the first and the second character of his/her password.

III. EMPIRICAL EXPERIMENTS AND RESULTS

In order to test the proposed mechanism, we conducted empirical experiments involving thirty participants: 17 females, and 13 males from various learning levels including secondary school, undergraduate, and postgraduate. Their ages were between 13 and 49 years old. They participated in the following phases:

1. **Registration Phase:** Each user provided his/her information by typing in their first name, last name, username, and password and specifies the time delay positions.
2. **Genuine Authentication Phase:** This phase aims to study the ratio of authenticating legitimate user login using the suggested mechanism. Once the sign-up information was obtained, each user attempted to login to his/her account five times. The password should be typed with a large time delay after the character(s) he/she selected at the sign-up time. This will add up to 150 trials by the genuine users. All the genuine sign-in trials are stored in a database.
3. **Imposter Detection Phase:** This phase aims to study the ratio of getting legitimate login as imposter users try to impersonate genuine users. Each imposter user was allowed five attempts against one randomly selected user account. Again add up to 150 trials by the imposter users. All the imposter sign-in data is stored in a database. Each imposter user was given a username and the associated password only. The imposters did not know any information regarding the positions of the specified delay(s). At the end of the experiments, the following

statistical metrics are used to evaluate the proposed mechanism:

1. **False Acceptance Rate (FAR):** FAR is the percentage ratio of all imposter users that have been accepted by the authentication system to all imposter users.
2. **False Rejection Rate (FRR):** is the percentage ratio of all genuine users that have been rejected by the authentication system to all genuine users.
3. **Precision:** is the percentage ratio of all genuine users that have been accepted to all accepted attempts regardless whether they were from genuine or imposter users.
4. **Recall:** is the percentage ratio of all genuine users that have been accepted to all the genuine users regardless whether they were accepted or rejected. The above metrics are frequently used in related studies [7], [8]. Table IV s shows the results obtained from the conducted experiments.

TABLE I: THE RESULTS OBTAINED FROM THE EXPERIMENTS.

case	N0		Metrics	%
Genuine Accepted	124		FRR	17.3
Genuine Rejected	26		FAR	11.3
Imposter Accepted	17		Precision	87.9
Imposter Rejected	133		Recall	82.6

A. Results and Discussion

In this section, the results obtained from the previous experiments are discussed in detail. Firstly, note that FAR is more critical to any secure system than FRR. Clearly, FAR is preferred to be small as much as possible since large FAR means that the system falsely accepts a large number of imposter users as genuine users and this decreases the security level of the system. On the other hand, large FRR means that genuine users may be rejected many times before they can access the system. This will not affect the security of the system but may annoy genuine users since they will try to login many times before they can access the system [6]. With this in mind, and according to the following arguments, the proposed system gives good results:

1. It is important to realize that during the experiments all imposter users are given the real passwords and have been told about the system and the essence of adding time delay(s) between certain password characters.
2. Most authentication systems block the user account if someone tries to login into the system after a certain number of failed attempts. Hence, FAR (= 11.3%) can be acceptable.
3. A value of 17.3 % for FRR seems acceptable since such a large percentage will not affect the system's security. This ratio is relatively large since we assume that genuine users still need more practice time to take control over the new system. In order to verify this assumption, FRR is recalculated again using only the fifth trial of genuine users. The FRR decreased to 13.3% and it will decrease more with more practice.
4. The average number of succeeding genuine user attempts was 4.13 out of 5. This proves the simplicity of the system and the positive effect of a short training period.
5. Most of the failed genuine attempts were because users leave many large time delays that are not close to each other. To ensure the success of the proposed system, users should use large time delays that are close to each other and the same thing to the small time delays. This will make the large delay difference located between the two sets.

IV. SECURITY ANALYSIS OF THE PROPOSED MECHANISM

Following are some security aspects of this new type of passwords.

1. Low implementation cost and no additional equipment are required, just a keyboard is enough. Other systems such as biometric authentication systems need special hardware in order to make it work.
2. The user interface for sign-in operation does not tell any person about the use of time delay. Hiding this information will make the process of attack difficult.
3. Adding time delay(s) into the password increases the password length and this enhances the password strength.
4. Using the proposed idea makes the brute force attack and dictionary attack useless. Obviously, these types of attack will try every possible combination of letters and any available information about the user in order to guess the password. Knowing the password only will not let an imposter user to access the system without adding appropriate time delay(s). This also applies to shoulder

surfing attack and keyloggers, where the attacker observes the user; what keys of keyboard he/she has pressed.

5. Remembrance of passwords is one of the cornerstones of the current password-based authentication system. However, human memory is in conflict with most password policies[10]; most systems enforce password rules about the length of the password and the diversity of the characters that comprise it in order to increase the password resistance to brute force attack. It will - on the other hand- reduce the ability of users to remember their passwords. Many studies have focused on the issue of what an individual can remember [11]. For example, Phone numbers are split into chunks to assist the memory. In this proposed system, user can select passwords that fulfill most of password rules and easy to remember such as Sun•Star•@•3, where • indicates time delay.
6. The proposed system can be used as an intrusion detection system (IDS). As an illustration, if an attacker uses the correct password to login into the system without adding time delay(s), then the system can act in different ways. One such way is to let him/her access the system and provide a fake data for the attacker and watch how he/she is doing with the data. Another method is to block the account and inform the administrator.
7. All users with different typing speed can use this system correctly. Each user will leave a time gap that is larger than his normal typing speed. Even though the user's typing speed may be changed the time nothing will be changed. On the other hand, authentication methods based on keystroke dynamics are not adaptive to any change in users' typing rhythm which may be affected by external factors such as injury, tiredness, or even typing speed change.

V. CONCLUSIONS AND FUTURE WORKS

The proposed authentication mechanism hardens the text passwords and makes using text-based password approach more secure as compared to conventional password based authentication systems. Adding time gaps between password characters enhances the resistance of the security system against password attacks such as shoulder surfing, brute force, keyloggers etc. In our authentication system, the attacker cannot get into the system even if he/she gets the correct password alone. Even though the proposed system has been tested using a small set of users, it produces good results and we believe that the system will give better results if we expand the set of users.

This method of authentication can be used to detect the theft of password files. A password file contains all information related to all users as well as usernames and passwords. Many methods were developed in the literature in order to detect this type of theft such as honeywords, Ersatz Passwords, PolyPasswordHash and SAuth . We believe that this method can be used in such problems [18].

This system uses only two types of time gaps (small and large). This can be a motivation to develop a system that can use many types of gaps such as small, medium and large. Surely this will increase the security of the system.

REFERENCES

[1] S. Venkatesh and K. Palanivel, "A Survey on Password Stealing Attacks and Its Protecting Mechanism," Int. J. Eng. Trends Technol.,vol. 19, no. 4, pp. 223–226, 2015.

[2] D. Mirante, "Understanding Password Database Compromises Technical Report," Tech. Rep. TR-CSE-2013-02, Polytech. Inst. NYU, 2013.

[3] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in Proceedings - IEEE Symposium on Security and Privacy, 2012, pp. 538–552.

[4] A. Conklin, G. Dietrich, and D. Walz, "Password-based authentication: A System Perspective," in 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the, 2004, vol. 0, no. C, pp. 1–10.

[5] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric
Recognition," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1,
pp. 4–20, 2004.

[6] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," The Scientific World Journal, vol. 2013. 2013.

[7] A. Mehler and S. Skiena, "Improving Usability Through Password- corrective Hashing," Proc. 13th Int. Conf. String Process. Inf. Retr., pp. 193–204, 2006.

[8] A. Juels and R. L. Rivest, "Honeywords: Making Password-Cracking Detectable," Proc. 2013 ACM SIGSAC Conf. Comput. Commun. Secur. CCS '13, pp. 145–160, 2013

[9] I. Erguler, "Some Remarks on Honeyword Based Password-Cracking Detection," IACR Cryptology. ePrint Arch., p. 323, 2014.

[10] A. H. Y. Makableh, "A New Approach To Detect Passwords File Theft And Intrusion Attempts," Zarqa University, 2016.