ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 31-34 (2013)
Special Issue of ICCTE 2013 - Held during 11-12 March, 2013 in Hotel Crowne Plaza, Dubai

Multifactor Strong Authentication Method Using Keystroke Dynamics

Dr. Abdulameer Khalaf Hussain [1], Mustafa Nouman Al-Hassan[2]
[1] Irbid National University, Jordan, abdulameer.hussain@yahoo.com
[2]Middle East University, Jordan, mustafa@orasnet.com

Abstract : This paper presents a multifactor authentication scheme using the keystroke dynamics. The scheme is composed of two keystroke patterns levels. In this first level, the speed of each user's password is measured depending on the comparisons between the registered speeds with different calculated thresholds. In the case of large deviations in the typing speeds, the user must transit to the second authentication level. In this level, the user decrypts ciphered thresholds by using his/her private key. The objective of this paper is to provide a strong authentication method and to assign a high accepted value of the typing speed by training the user to type his/her password 100 times as a pretest before the actual registration level . In this case , the system results in a high precise speed values . The system is analyzed using different statistical measurements.

Keywords : Security **,** Authentication , keystroke , typing speed , thresholds.

INTRODUCTION

In security systems, there are two complementary mechanisms used to determine the access or modification of data. These mechanisms are: authentication and authorization. Authentication is the process for identifying and verifying who is the person,(this process is used at the initial register of the system, whereas authorization is the process of giving persons access to system objects based on their identity [1].

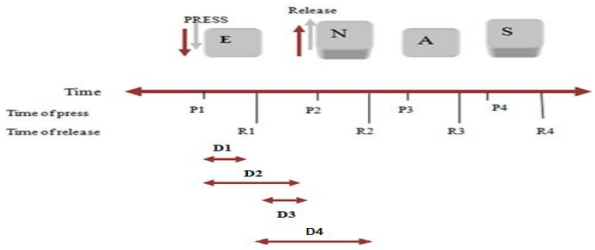
Authentication can be categorized into one of three classes:

1: Providing a secret knowledge (e.g. a password). 2: Providing a unique piece of hardware that can be matched to the user identity (e.g. an ID card), which cannot be shared at the same time as sharing passwords, but can be forgotten, lost or even stolen by attackers. 3: The biometric features by measuring some unique attribute of the user, either the physiological features (e.g. Face recognition, eye (normally, retinal or iris patterns), hand Geometer), or the behavioural features (e.g. Biometric property, DNA, keystroke dynamic). [2,3]

Keystroke dynamics is the process of analyzing and measuring the style in which any user types the elements of his/her secret information by the keyboard. This process is either done at either during a complete session, which is called continuous keystroke dynamics [4], or during login time or after a predetermined period of time which is called static keystroke dynamics and [5,6]. Keystroke dynamic authentication is the more secure option as it is impossible for an attacker to learn someone's typing characteristics and it is easier to implement into an existing security system. Also this method does not require an additional hardware because it uses the keyboard to measure keystroke dynamic. As well as, the keystroke provides stronger, more user friendly password and inexpensive method. Many research proved that keystroke rhythm is an effective mean for ensuring the user's identity, this means that we can authenticate people in a strong manner using keystroke dynamic, [7] .

Keystroke dynamic system consists of three main steps: data collection, feature extraction and pattern classification. In data collection, raw data is collected and the registered to include all typed keys, related timing information and key events (press or down key, and release or up key) in profile. Feature extraction is concerned with different measurements used by keystroke dynamics. Theses measurements are calculated when the user presses keys using the keyboard. Possible measurements consist of at least the following features (as shown in Fig.1) [8,7].

Fig: 1 – Extracted features of the Keystroke Dynamic



Press-Release (D1): This parameter calculates how long it takes a specific key is pressed until the user releases that key (sometimes is called Dwell time or hold time).This parameter can be calculated as: D1 = R1 – P1, where R1 represents the release, while P1 represents the press.

2: Press-Press (D2): This represents the interval between 2 successive key presses (sometimes is called Keywait) and is represented as follows : D2 = P2 – P1.

3: Release - Press (D3): This measures the interval time between a key release and the next key press time, (also is called flight time) , and is represented as D3 = P2 – R1.

4: Release- Release (D4): In this measurement interval between 2 successive key Releases are calculated, and is represented as D4 = R2 – R1.

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 31-34 (2013)
Special Issue of ICCTE 2013 - Held during 11-12 March, 2013 in Hotel Crowne Plaza, Dubai

All typed keys must be stored with related timing information and also key events (press or down key, and release or up key) in a database as reference profile "RP" just when create a new account.

In pattern classification, the classifier is responsible for the process of decision whether the authentication procedure accepts or rejects the user by comparing a references profile "RP" of a certain user and a test profile of unknown user.There are two main keystroke analysis approaches for verifying the user's identity. These approaches are: statistical techniques and neural networks techniques. [9].

RELATED WORKS

In [10] a study provided quantitative information related to the values of specific parameters such as attribute acceptance thresholds, the number of accepted attributes, and the effect of contiguity. This study explained the possibility of using keystroke dynamics as a tool for user identification.

Pin Shen .et al [11] proposed training and testing method for collected data and extracted four different features: (D1= dwell time, D2= Interval between 2 successive key PRESS, D3= Interval between a key release and the next key press time, D4= interval between 2 successive key releases) from a raw data. This proposal calculated the mean and standard deviation values, Gaussian probability density function used to transform these test feature data into the scores. The authors proposed Direction Similarity Measure (DSM) to measure the scores. The best result of EER was 6.36%, in addition results of the experiment shows that the combination of dwell time and flight time (D1+D4) yields a better result than using them individually (D1, D2, D3, D4).

Peng .et al proposed a novel and simple statistical fusion method to be used as the classifier in password-based authentication systems to enhance their security. The proposed method used to extract keystroke data based on the time instances of pressing and releasing a key. Keystroke duration (hold time) and four keystroke latencies were calculated using these data, FAR is 1.035% and FEE is 0% was achieved. [12].

In [13] a proposal of several ways dedicated to improve uniqueness, depending on the peculiarity of the typing style in keystroke dynamics: inserting any number of pauses, typing a password according to a rhythm from certain tune, or a style, and typing a password with a minimum duration time of each character. Also this proposal suggested means to improve consistency, which depends on the typing skill and the concentration level of the user, use of visual, audio and audiovisual cues was suggested, the ideas and results presented in this work are preliminary.

THE PROPOSED KEYSTROKE DYNAMIC SYSTEM

The proposed keystroke authentication consists of two authentication levels. The first level is dedicated to measure the user's typing speed of the password characters and the second level provides a strong authentication procedure by checking the authenticity of each user using private information derived from the first level. This proposed system is different from other keystroke system in that each authenticated user is trained on the password for 100 times before registering the typing speed of that user in a reference file.

Level 1: Typing Speed Measurement (Training level)

As the user types, ticks are recorded and held in an array. The password is requested ten times. For each of the ten attempts that are counted, ticks between keystrokes are recorded in a two dimensional array. If the password is n+1 characters long ,then row j of the array will contain n entries ,the ith one corresponding to the number of ticks that occurred between the ith and i+1th keystroke during the jth trial. When all of the copies are entered, the program computes averages and standard deviations for each of the columns of the array. These are held in a signature file for the user.

The steps of this level are:

1: Enrolment step: In this step , the two dimensional matrix is constructed to contain 10 trials of user's typing speed as mentioned above .This matrix is illustrated in table 1.

Table 1: Enrolment matrix

First character	Second character	.. n character	Row Averages	Standard Deviations	K-nearest Neighbored
X_{11}	X_{12}	....X_{1n}	Av_1	SD_1	K1
X_{21}	X_{22}	....X^{2n}	Av^2	SD_2	K_2
. . . X_{m1}	. . . X_{m2}	. . . X_{mn}	. . . Av_n	. . . SDn	. . . K_n

Each Xi represents the time between successive characters, Avi denotes the average of each row, SDi represents the standard deviation for each row and Ki is the calculated K-nearest from additional trial of the same password which represents a test sample of the password. K-nearest neighbor of each row is calculated as follows:

$Ki = \sqrt{(Avgtest - AVGi)2 + (Stdtest - STDi)2}$, where Avgtest and Stdtest represent the average and standard deviation of a unique test data .

This procedure is applied without training the users in order to compare it with the proposed system with a pretest training and in this case we calculate the averages of each row in addition to the k-nearest neighbor.

These collected data are stored in a profile and this file contains information for each authenticated user to be used for next step (authentication phase).

The authenticated user must maintain the upper average value UR and the lower average value LR to be used in level 2 which is responsible for checking user's private information in the case of high deviation in typing speed. These ranges must be calculated for each user and stored in a table called Range_Table (Table2):

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 31-34 (2013)
Special Issue of ICCTE 2013 - Held during 11-12 March, 2013 in Hotel Crowne Plaza, Dubai

Table2 : Range_Table

USERS	UPPER RANGE	LOWER RAMGE
U_1	UR1	LR1
U_2	UR2	LR2
Un	Um	LRn

2: Test Step: The authenticated user must be tested by typing the password for 50 trials and in each trial, the typing speed is measured 7 times. This procedure gives an indication of whether the calculated ranges are measured correctly or not and to count the number of matching typing speeds with the stored typing speed in the matrix. So, the experiments are an important factor to get more accurate typing speed and lies within the typing ranges. The proposed system results in a high percent of matching the test typing speed with those registered in the enrolment step and with a very low deviation between the upper average speed and the lower average speed as will be shown in result section.

3: Authentication step

After this extensive training with password typing, the user enters his/her password for each login session. The typing speed is calculated as in enrolment step but in this step, the user types the password for one time and the average is calculated (Avgauth). If the average is between the upper and lower ranges (UR and LR) of that user ;ie LRi<= Avgauthi >=URi, this user is considered an authenticated user otherwise that the proposed system checks that user in the second level.

Level 2: Checking user's private information

The first level was dedicated to test users' typing speed within a narrow range of their typing speed, but, in some situations there are variances or deviations found in the typing speed for the authenticated user .This problem is common for almost keystroke authentication approaches. In order to solve this problem for the authenticated user and to prevent the impostor user, the proposed system tests secret private information of that user generated which was generated in the level 1.

In this level, each authenticated user must know the values of upper and lower ranges of typing speed stored in Range_Table which each element represents private information for that user. The system asks the user to enter these ranges. In order to get strong authentication, the user encrypts these values by his/her private key which represents the user's signature. For this reason, the proposed system uses the RSA scheme.

These steps are illustrated below:

$C_1 = (URi)^d \bmod n$

$C_2 = (LRi)^d \bmod n$

C_1 and C_2 represent ciphier1 and cipher2. , d is the user's private key , n is the product of two prime numbers p and q (n=pxq). C_1 and C_2 are sent to the system. The system decrypts these two ciphers using the public key (e) for that user to get the corresponding upper and lower typing speed ranges for that user as follows:

$Uri = (C_1)^e \bmod n$

$LRi = (C_2)^e \bmod n$

The system matches these ranges with stored ranges, if they are equal, the system considers that user as an authenticated user although of the deviations in typing ranges occurred in level 1, and otherwise, the user is rejected.

RESULTS

When using the training test of 100 trials for each password we get averages of speeds that contain more matching vales as shown in Table3. For example if we take the integer values for each average for 10 trials we get 2 matching values for 4 values with matching percentage of 20.

Table3: Registration Table

Speed of 1st and 2nd chs	Speed of 2nd and 3rd chs	Speed of 3rd and 4th chs	Speed of 4th and 5th chs	Speed of 5th and 6th chs	Speed of 6th and 7th chs	Speed of 7th and 8th chs	Speed of 8th and 9th chs	Speed of 9th and 10th chs	Average of each row
93	219	171	94	156	203	156	234	234	173.333
109	234	187	94	172	202	156	234	219	178.556
94	234	187	78	140	219	109	234	249	171.556
93	234	188	93	156	203	156	265	172	173.333
125	219	202	94	141	202	94	218	234	169.889
109	218	203	109	125	218	125	234	265	178.444
110	234	187	125	124	203	141	234	234	176.889
109	203	172	124	110	234	109	234	249	171.556
125	203	187	125	125	218	109	250	234	175.111
93	234	172	109	141	202	141	249	234	175.000

For other trials with 7 attempts of the same user we get matching averages of 42.9% as shown in the Table4:

Table4: Training Table

Speed of 1st and 2nd chs	Speed of 2nd and 3rd chs	Speed of 3rd and 4th chs	Speed of 4th and 5th chs	Speed of 5th and 6th chs	Speed of 6th and 7th chs	Speed of 7th and 8th chs	Speed of 8th and 9th chs	Speed of 9th and 10th chs	Average of each row
156	234	187	109	125	219	46	250	203	169.8889
141	234	187	125	124	203	125	218	234	176.7778
140	250	187	140	94	218	94	234	234	176.7778
125	234	187	125	93	219	78	234	218	168.1111
125	281	187	140	125	203	109	234	203	178.5556
125	234	187	140	94	249	47	250	234	173.3333
125	234	187	125	124	203	141	234	218	176.7778

If we compare the above results with the same trials without pre-test we find no matching average values and there is a high deviation between the upper average value and the lower average value as shown in Table 5.

International Journal of Science and Applied Information Technology (IJSAIT), Vol.2 , No.2, Pages : 31-34 (2013)
Special Issue of ICCTE 2013 - Held during 11-12 March, 2013 in Hotel Crowne Plaza, Dubai

Table 5: Traditional keystroke authentication

Speed of 1st and 2nd chs	Speed of 2nd and 3rd chs	Speed of 3rd and 4th chs	Speed of 4th and 5th chs	Speed of 5th and 6th chs	Speed of 6th and 7th chs	Speed of 7th and 8th chs	Speed of 8th and 9th chs	Speed of 9th and 10th chs	Av. each row	K-Nearst
234	125	250	203	187	327	203	234	203	218.444	41.040
234	94	234	218	140	406	281	218	250	230.556	48.019
234	109	234	109	94	374	250	202	234	204.444	24.392
234	94	203	140	109	312	156	219	234	189.000	8.263
218	109	219	125	124	297	171	203	203	185.444	16.554
203	62	203	141	171	281	218	234	203	190.667	16.051
187	78	219	156	93	312	187	203	188	180.333	8.079
203	62	234	234	141	312	234	203	202	202.778	20.260
219	47	218	172	124	281	172	203	156	176.889	12.244
203	94	202	125	78	281	141	187	156	163.000	24.812

ANALYSIS

The proposed system presents a strong method for user's authentication using the key stokes dynamics. Most of the traditional keystroke authentication depends on a registration trial for the password and when these systems checks the authenticity of that user after the registration phase the authentication phase contains a high deviation between the registered password speed and that in the authentication phase . In addition to that the authentication phase has no matching average speeds, but in this proposed system we train the user for typing his/her password before the registration phase. So the system results in various matching values in the authentication phase and we get matching values of 42.9 % and a low deviation between the low and high averages. Also we get a low deviation between low and high average values.

CONCLUSION

This paper presents a strong multifactor method using key stroke authentication. This system is considered the first application that uses the training of typing speed before the actual registration phase. This pretest training results in a more precise speed because the user will be accustomed with the best typing speed . Also this system considers a rare occurred case when we get a very high deviation not because the this applied system but because some environmental and physiological conditions .In this case the user is checked by providing private information which is the second level of this proposed system. This leads to a strong authentication scheme.

REFERENCES

[1] Ilonen, Jarmo. "Keystroke dynamics" .Advanced Topics in Information Processing–Lecture (2003). Available: http://www2.it.lut.fi/kurssit/03-04/010970000/seminars/Ilonen.pdf

[2] Barghouthi .H (2009) ," Keystroke Dynamics How typing characteristics differ from one application to another", *MSc Thesis, Gjøvik University College*.

[3] Checco, J. C. (2003). "Keystroke dynamics & corporate security". *WSTA Ticker Magazine*. Available: http://www.checco.com/about/john.checco/publications/2003_Keystroke_Biometrics_Intro.pdf

[4] Bergadano, F., Gunetti, D., and Picardi, C. (2002). "User authentication through keystroke dynamics". *ACM Transactions on Information and System Security (TISSEC)*,vol. 5 no. 4, pp.367-397.

[5] Monrose, F., and Rubin, A. D. (2000). "Keystroke dynamics as a biometric for authentication". *Future Generation Computer Systems*, vol.16 no. 4,pp. 351-359.

[6] Zhou, C. (2008). "A Study of Keystroke Dynamics as a Practical Form of Authentication". *MSc Thesis ,Pomona College*.

[7] Joyce, R., and Gupta, G. (1990). "Identity authentication based on keystroke latencies". *Communications of the ACM*,vol. 33 no.2, pp.168-176.

[8] Jain, A. K., Ross, A., and Prabhakar, S. (2004). "An introduction to biometric recognition". *Circuits and Systems for Video Technology, IEEE Transactions* on,vol. 14 no.1, pp.4-20.

[9] Bours, P." Authentication course", *IMT 4721, Gjovik University College*. Reader for IMT, 4721.

[10] Revett, K. (2009)." A bioinformatics based approach to user authentication via keystroke dynamics". *International Journal of Control, Automation and Systems*,vol. 7 no.1,pp. 7-15.

[11] Teh, P. S., Teoh, A., Ong, T. S., and Neo, H. F. (2007). "Statistical fusion approach on keystroke dynamics". *In Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference* , pp. 918-923.

[12] Cheng, P. C., Chang, T. Y., Tsai, C. J., Li, J. W., and Wu, C. S. (2011). "A novel and simple statistical fusion method for user authentication through keystroke features". *Journal of Convergence Information Technology*, val. 6 no. 2.

[13] Cho, S., and Hwang, S. (2005). "Artificial rhythms and cues for keystroke dynamics based authentication". *Advances in Biometrics*, pp. 626-632.