International Journal of Computer Science and Software Engineering (IJCSSE), Volume 6, Issue 11, November 2017
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 266-272

Investigating the Effectiveness of Arabic Language for Free-text Keystroke Dynamics Authentication

Suliman A. Alsuhibany[1], Muna Almushyti[2], Fatimah Alkhudier[3] and Noorah Alghasham[4]

[1] Department of Computer Science, Qassim University, Qassim, Saudi Arabia

[2, 3] Deanship of Educational Services, Qassim University, Qassim, Saudi Arabia

[4] Information Technology Qassim University, Qassim, Saudi Arabia

[1]salsuhibany@qu.edu.sa, [2]{m.almushyti, [3]f.alkhudhayr}@qu.edu.sa, [4]n.alghasham@qu.edu.sa

ABSTRACT

Online services depend greatly on authentication mechanisms in order to give a sufficient protection to the provided data. Since various security and usability drawbacks have been reported, this paper investigates a free-text keystroke dynamic authentication approach which provides acceptable security and usability levels. In particular, this study extends previous studies in which the Arabic language is incorporated by investigating the utilization of different timing features, as well as applying different classification methods. Based on a controlled experiment for Arabic language, a better False Acceptance Rates (FAR) was achieved, which is equal to 2% by using both Euclidean and Bhattacharyya distance measures, while the False Rejection Rates (FRR) was 0.0 using Euclidean Distance and 0.06 using Bhattacharyya Distance. The results showed that the system performance is improved compared to other studies.

Keywords: *Keystroke Dynamic, Authentication, Security, Usability, Bhattacharyya Distance, Euclidean Distance, Free-text, Arabic language.*

1. INTRODUCTION

Online services have become an important part for executing numerous tasks in almost all aspects of our lives in which the trust plays an important role in executing these tasks. In order to perform a trusted communication between these services and the user, the user authentication, which is the process of confirming an alleged identity, has proposed. The traditional approach for user authentication is the password (i.e. the knowledge-based authentication based on something one knows) [2]. Unfortunately, this traditional approach suffers from the security-usability trade-off dilemma. Not only this, but a number of methods can be used to crack passwords, such as spyware, social engineering, a dictionary attack, and even a brute force attack. These reasons oblige users to apply extreme measures with a view to safeguarding their passwords, for example, using complex passwords and updating their passwords periodically. Thus, the need for an alternative authentication method, which can provide an ease of use to the user in addition to robust security, has turned into a necessity.

One of the alternative authentication methods is keystroke dynamics, which is a behaviour-based approach that utilizes a person's typing patterns to validate her/his identity by monitoring the keyboard. Keystroke dynamics are based on timing features that can be extracted from the time lapses between two actions on the keyboard, such as the release of the first key and the depression of the second one. Moreover, a keystroke dynamic is, as stated in [1] "not what you type, but how you type." Therefore, this alternative method of authentication provides a high level of usability while maintaining a strong system protection [4].

So far, a keystroke dynamic authentication has two main classes: free-text and fixed-text [2]. The free-text keystroke dynamic authentication class uses the typing patterns of the user without entering a predefined text, while the fixed-text class uses the typing patterns of the user with entering a predefined text. In the latter class, a user needs a training session, and then remembering the text at the log-in time. On the other hand, the free-text class overcomes the problem of memorizing the text, as the text used for the enrolment session does not have to be the same as the text used for the log-in session. Furthermore, the free-text class can be used in a number of applications, such as enhancing security by continuous and nonintrusive authentication [3]. Thus, this paper focuses on the free-text class only.

Generally speaking, the free-text keystroke dynamic has only been comprehensively studied using English language input [4]. Although there has been very little research carried out relating to other languages, such as in [5], these languages share the same alphabet with English. However, very recent studies in [6] and [17] have studied the free-text keystroke dynamic using Arabic language input. The Arabic and English languages are very different to each other. For example, Arabic is a Semitic language belonging to the Afro-alphabet language family, whereas English is a Germanic language from the Indo-European language family [18].

Therefore, this paper extends the previous studies in [6, 17] by investigating the utilization of different timing features and applying different classification methods. In particular, the study in [6] only examined Arabic input in keystroke dynamic authentication by using the keyboard's key-layout as a timing feature extraction method, and Support Vector Machines (SVMs) and Decision Trees (DTs) as classification methods, while the study in [17] used only Euclidean distance as a classification method.

In our study, we applied three timing features: keystroke duration, di-graph duration, and latencies. These features, thus, can be used to construct a unique signature for each user. Furthermore, we used two different distance measures: Bhattacharyya and Euclidean, to find the level of similarity between a user's log-in data and a user's profile. Therefore, the results showed that our approach offers better performance than the previous results in [6, 17] with a FAR of 0.20 and a FRR of 0.06.

The rest of this paper is organized as follows. Section 2 discusses similar prior works in the area of keystroke dynamics user authentication. Section 3 describes the methodology used, while the experimental study is explained in Section 4. Section 5 presents the results. Section 6 discusses the results. Section 7 concludes the paper and points out our research contribution to future work.

2. RELATED WORK

This section mainly highlights related works on the free-text keystroke authentication class. This class is considered in this paper due to its applications in many useful settings in order to assist in real life situations, in addition to its usability and security aspects.

A recent comprehensive survey of free-text keystroke dynamics methods was done in [4]. In this study, various factors that may affect the performance of the authentication system are discussed. Moreover, the methods used for feature extraction and classifications are also compared, while several security issues that can be defeated by the free-text keystroke dynamics authentication approach are discussed. This study concludes that it is not a straightforward task to determine the best method to achieve the best authentication accuracy. Furthermore, although fixed-text keystroke dynamics authentication is obviously more accurate than free-text keystroke authentication, it may be good practice to take into consideration the key hold time, the di-graph's duration and latency times for the free-text keystroke dynamics authentication.

As English language input has been considered in almost all research into keystroke dynamics, a study in [5] provided an evidence that the typing dynamics of free-text can be used for user authentication even when typing samples are written in different languages, which is the first attempt in this direction of research. Specifically, the Italian language is used in [5] to type samples, and to compare these samples with others typed in English language. The results showed that free-text can be a useful tool for user authentication even when typing dynamics stem from the use of different languages.

Moreover, the Arabic language has completely different characteristics. In 2016, a ground-breaking paper in which Arabic language input is used for free-text keystroke dynamic authentication was proposed in [6]. This study examined the usefulness of applying the keyboard's key-layout based method, as this method classifies every two characters typed consecutively based on their relationship to each other, and their overall location on the keyboard. Additionally, the SVMs and DTs were utilized to classify individuals based on the timing features. The results of this study showed that Arabic input was successfully used for free-text keystroke dynamics user authentication. Furthermore, Alsuhibany et al. [17] extended this study using Euclidean distance.

Our paper, therefore, extends these studies [6, 17] by investigating the utilization of different timing features and applying different classification methods, as will be shown in the following sections.

3. METHODOLOGY

The feature definition, extracting timing vectors, typing data, finding distance, and classification method are discussed in this section.

3.1 Feature Definition

Timing features of keystroke dynamics can generally be extracted from two actions on the keyboard: the depression, i.e. the press, (Dn) and the release (Un) of a key (n) in milliseconds. These features were extracted for every key at the log-in session in order to compare them with the timing features extracted at the enrolment session. In this study, as suggested in [4], the following three timing features were extracted:

- *Keystroke duration or hold time*: This is the time when the key is pressed until it is released.
- *Keystroke latencies or flight time*: This is also known as Down-Down (DD) or Press-Press (PP), which is the time between two consecutive keys presses.
- *Di-graph duration*: This is the elapsed time between the release of the first key and the depression of the second one.

3.2 Extracting Timing Vectors

Once timing features have been defined and extracted, we observe a number of outliers and noisy data, such as when two keys are mistakenly pressed together by a user. These outliers and noisy data lead to a very large or very small data point. Therefore, this outlying data has been removed during the pre-processing phase. While the timing vector in our study has the mean of keystroke duration and keystroke latencies, this time vector was extracted and stored in the database as the user's profile.

3.3 Typing Data

The following presents the typing text that participants were asked to type in the log-in phase, while the sign-up typing text will be discussed in Section 4. In the interest of comparing Arabic input performance with another, we asked the participants to type English text. For the usability aspect, we seek to make the typed texts as short as possible empirically with commonly used words, although they seem long.

- *Arabic Text:* The Arabic text contains 199 characters as follows:

"الاسلام منهج شامل لجميع جوانب الحياه وسلوك الانسان وهذا الشمول

لايقبل الاستبعاد ولا التخصيص بل هو كامل تام بكل ماتحمله

الشموليه ومن البديهيات في الاسلام انه لعموم البشر وكافه

الخلق"

- *English Text*: The English text contains 92 characters as follows: "very little is needed to make a happy life it is all within yourself in your way of thinking."

3.4 Finding Distance and Classification Methods

In order to find out how much user's test data are close to the user's profile, two distance measures are used: Bhattacharyya and Euclidean distances.

For Bhattacharyya distance, we calculate the distance between two timing vectors by applying Bhattacharyya Distance's equation D_B [7]:

$$\mathbf{D_B} = \frac{1}{8}(\boldsymbol{M_2} - \boldsymbol{M_1})^T \left[\frac{\Sigma_1+\Sigma_2}{2}\right]^{-1} (\mathbf{M_2} - \mathbf{M_1}) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}} \quad (1)$$

where M_1 is the mean of vector1, M_2 is the mean of vector2, and Σ_1 and Σ_2 are the covariance of vector1 and vector2, respectively. Also, $0 \leq D_B \leq \infty$, which means if the vectors are similar, then the distance will be 0 [7].

The second measure was Euclidian distance. We measure the distance between two vectors based on Euclidean distance equation in three-dimensional space [8]:

$$\boldsymbol{d(x,y)} = \sqrt{(\boldsymbol{x_1} - \boldsymbol{y_1})^2 + (\boldsymbol{x_2} - \boldsymbol{y_2})^2 + (\boldsymbol{x_3} - \boldsymbol{y_3})^2} \quad (2)$$

where *x and y* are two vectors. In this study, *x* represents the user's log-in vector and *y* represents the user's profile vector. Also, $d(x,y)$ is equal or greater than 0 (i.e. $d(x,y) \geq 0$) [8].

To classify the data, we need to determine a threshold for both Bhattacharyya and Euclidian distances, which helps us to identify users. Based on the observation, we specify the threshold of D_B empirically to be 0.040 for English and 0.10 for Arabic. This means that if the distance between a user's login vectors and a user's profile exceeds this determined threshold, the system will not allow the user access. For Euclidian distance, we specify the mean and SD (Standard Deviation) of a user's profile vector as the thresholds between two user vectors. Section 5 shows the results of choosing these thresholds.

3.5 Software and Computing

To analyse the timing vector's data, we used MATLAB, which provides a statistical toolbox that helps us to calculate the distances between vectors.

4. EXPERIMENTAL STUDY

We conducted a controlled laboratory experiment in which the subjects were asked to type texts in both Arabic and English languages. In this section, the setup and the procedure of the experiment are explained.

4.1 Experiment Setup

The experiment involves a number of subjects acting as normal users. The following details the system, participants, and materials of the experiment.

System: we used a Java Programming language in order to develop a friendly Graphical User Interface (GUI) to collect the data. This data then is stored in a MYSQL database. This GUI is installed in a *MacBook Pro* laptop. There are two main GUIs: sign-up and log-in. For the sign-up GUI, there are two areas to be filled: an email box and typing a displayed text. It is interesting to note that we used the email address as an identity for each user. Fig. 1 shows the English language sign-up GUI; the same GUI but for the Arabic language is shown in Fig. 2.

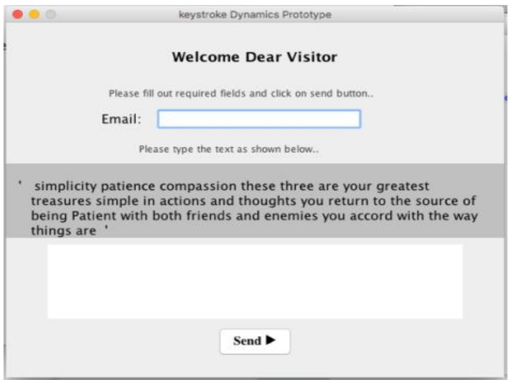

Fig. 1. Sign-up GUI in English.

Fig. 2. Sign-up GUI in Arabic.

We extract timing features only when the user starts typing the displayed text. Based on the free-text keystroke definition [4], the text used for the training phase is not necessarily the same text used for the testing phase. For purposes of usability, the English text did not include upper-case letters, numbers or punctuation marks. Similarly, for Arabic text, we avoided including letters that required the need to press two keys together; for example, to type a letter “ؤ” you need to press the shift key and the M key together.

Participants: Thirty users participated over two weeks. Participants had different levels of typing skills and were aged between 19 and 50. Furthermore, they were native Arabic speakers with various levels of English skills. Also, they were familiar with the keyboard of the *MacBook Pro* laptop.

Materials: stimulus and rational: The stimulus material provided to participants consisted of two texts: sign-up text and log-in text. The sign-up text in Arabic was 199 characters in length, while it was 208 characters in length for the English text. For the log-in text, it was 182 characters for Arabic, whereas it was 92 characters in English. Although several studies have shown an interest in using short free-text [e.g. 1, 10, 11], it seems not enough to use short texts to analyse keystrokes, as it does not provide an adequate amount of information to discriminate between users [12, 13, 14]. Therefore, we decided to use these lengths of characters for both languages.

Furthermore, the previous study [6] utilized a standard Arabic keyboard [15] for conducting the experiment. However, according to [16], the system has a good chance of accurately identifying a user while the user uses the same type of keyboard for training and testing. Thus, since two languages were used in this study, we used the default keyboard for the *MacBook Pro* laptop shown in Fig. 3, where Arabic and English letters appear in all keystrokes. For example, a keystroke represents the letter “H” in English and the letter “ا” in Arabic.

Fig. 3. Arabic-English keyboard of the MacBook Pro laptop.

4.2 Experiment Procedure

In this section, we describe how we ran the experiment, i.e. instructions to the subjects, procedures, and data collection.

Instructions to Subjects: Participants were instructed that the goal of this study was to type the given texts as normal. Participants were asked to keep their phones off (or on silent) and avoid chatting with friends, to prevent any interruption during typing the text. Participants were instructed that they needed to sign-up, first by entering the email, then typing the given text. After the sign-up phase, participants were instructed to log-in to the system by providing the entered email in the sign-up phase and typing the given text. During typing, participants were instructed that they are allowed to use the spacebar and backspace keys during their typing.

Finally, a confirmation message will appear to show the end of the experiment.

Procedure: The experiment was conducted in a controlled laboratory environment to avoid any distractions, and to collect the desired data without any biases. Each user was asked to sign-up first for Arabic input by entering an email and typing a given text. Then, the users can log-in to the system by using the previous email used in the sign-up phase and typing a different text. After taking a break, the user is asked to do the same task for English input.

Collected Data: The system records typing attempts for Arabic and English inputs by each participant. For each language, as we mentioned previously, each di-graph was captured three times in milliseconds.

5. RESULTS

In our experiment, all participants successfully completed their given task. In order to infer the performance of the system, two error rates are used, which are False Acceptance Rate (FAR) and False Rejection Rate (FRR) [9]. FAR is the ratio of unauthorized users who have successfully gained access to the system. We determined this by comparing the user's log-in data with two users randomly **based** on their training data. On the other hand, FRR is the ratio of authorized users who are ***falsely*** rejected access to the system. This was found by comparing the authorized user log-in data with its profile data. If the distance exceeds the determined threshold that we described in the methodology section, then the user will be rejected incorrectly.

For Euclidean distance, we have chosen two thresholds for each language; these thresholds were explained in Section 3. First, we select the *mean* of the user's profile data as a threshold. As a result, we got 0.46 FAR without rejecting any authorized users for English input, while we got a higher value for FAR, which reaches 0.52 for Arabic input. These values are dramatically decreased when we used the *Standard Deviation* (SD) as a threshold, as will be seen later on. Interestingly, a better FAR for both languages was achieved, which is equal to 0.22 for English and 0.20 for Arabic. Table 1 summarizes these results.

Table 1: Error rates of using Euclidean Distance

Distance Measures	**English Text**				**Arabic text**			
Threshold	**Mean**		**SD**		**Mean**		**SD**	
Euclidean Distance	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR
	0.46	0.0	0.22	0.0	0.52	0.0	0.20	0.0

For Bhattacharyya distance, we selected 0.40 and 0.10 as a threshold for English and Arabic texts, respectively. For Arabic, we achieved 0.20 FAR and 0.06 for FRR, while we got 0.103 FAR and 0.17 FRR for English texts. Table 2 summarizes these results.

Table 2: Error rates of using Bhattacharyya Distance

Distance Measures	**English Text**		**Arabic text**	
Threshold	0.40		0.10	
Bhattacharyya Distance	FAR	FRR	FAR	FRR
	0.103	0.17	0.20	0.06

6. DISCUSSION

The accuracy obtained in our experiment specifies that the approach of applying timing features can enhance the performance of the system while typing Arabic input. In particular, the combination of three features: KD, DD and UD with using different parameters, yields interesting results. That is, using the SD as a threshold in the Euclidean distance is better than the mean in the case of the FAR. Not only this, but FRR was also 0%, which means all authorized users can access the system without a rejection. Based on this, we recommend using the SD as a threshold for both languages (i.e. Arabic and English) when applying the Euclidean distance due to the accuracy and security levels shown in this study.

Moreover, the results of our experiment are comparable with other studies. For the text length, in [9] the text length is longer than the text typed in this study for building a user's profile (i.e. the enrolment phase). That is, 208 characters were used in this study; while in [9] 380 characters were used. Although it is stated in [12, 13, 14] that a longer text is recommended for free-text keystroke in order to increase accuracy, our results have greater accuracy; even the typed text was shorter than the used text in other studies. In particular, the authors in [9] showed 28% FAR in combining hold times and Up-Down time for English input, whereas we got 0.2% FAR by the selected features in our study.

Additionally, very interesting results have been achieved by Bhattacharyya distance. That is, although FAR was almost the same as in [6] for Arabic input, the FRR (0.06) was far less than in the previous study [6]. Even though FAR is more important than FRR in terms of security, the result of the FRR in this study can be an enhancement to the previous study while keeping the same FAR (i.e. 0.2). Furthermore, we empirically observe that applying different thresholds for Arabic and English languages can increase the performance of the system efficiently. This is due to the different characteristics between the Arabic and English languages.

It is interesting to note that the FRR was 0.06 in our study compared to 0.5 and 0.4 (for both classifiers) in the previous study [6]. The reason behind this might be that in our study all participants were native Arabic speakers, while in [6] the participants were either native to Arabic or more familiar with Arabic.

Since a controlled experiment type was applied in our study, this type may not have the same characteristics as those in realistic situations. However, through this type of experiment, biases, which are difficult to control, can be avoided compared to an uncontrolled experiment type. That is, each user's surroundings can be very different, which leads to difficulties in analysing the data. Thus, applying a controlled experiment can provide consistent keystroke data from users, which affects experimental control over unanticipated biases.

7. CONCLUSIONS

This paper extends the previous studies by investigating Arabic language input for free-text keystroke dynamic user authentication. In particular, three features were examined in this paper: keystroke duration, di-graph duration and latencies. Through a controlled experiment, the results showed that this combination leads to acquiring more accurate results. Furthermore, we applied Bhattacharyya and Euclidean distances for classifying the collected data. In the case of Bhattacharyya distance, different thresholds were empirically determined for each language to enhance the accuracy of the results. Also, different parameters were investigated as a threshold for the Euclidean method. Overall, Bhattacharyya distance is more accurate in its application to both English and Arabic inputs with a low FAR.

This study can be extended by applying these methods to obtain data from non-Arabic speakers. It can also be used to compare Arabic with languages other than English, while different features can be investigated using this study method. Finally, these results can be improved by adding more features or by using different classification methods.

REFERENCES

[1] Monrose, Fabian, and Aviel Rubin. "Authentication via keystroke dynamics." In Proceedings of the 4th ACM conference on Computer and communications security, pp. 48-56. ACM, 1997.

[2] Karnan, Marcus, Muthuramalingam Akila, and Nishara Krishnaraj. "Biometric personal authentication using keystroke dynamics: A review." Applied Soft Computing, vol. 11, no. 2, pp. 1565-1573, 2011.

[3] Bours, Patrick, and Soumik Mondal. "Performance evaluation of continuous authentication systems."IET Biometrics, vol. 4, no. 4, pp. 220-226, 2015.

[4] Alsultan, Arwa, and Kevin Warwick. "Keystroke dynamics authentication: a survey of free-text methods." International Journal of Computer Science Issues, vol. 10, no. 4, pp. 1– 10, 2013.

[5] Gunetti, Daniele, Claudia Picardi, and Giancarlo Ruffo. "Keystroke analysis of different languages: a case study." In International Symposium on Intelligent Data Analysis, pp. 133-144. Springer Berlin Heidelberg, 2005.

[6] Alsultan, Arwa, Kevin Warwick, and Hong Wei. "Free-text keystroke dynamics authentication for Arabic language." IET Biometrics (2016).

[7] Mak, Brian, and Etienne Barnard. "Phone clustering using the Bhattacharyya distance." InSpoken Language, ICSLP 96. Proceedings., Fourth International Conference on, vol. 4, pp. 2005-2008. IEEE, 1996.

[8] Waterman,G. "Euclidean Space and Vectors," in Linear Algebra I, Oregon Institute of Technology, (pp.43-36).Available: http://math.oit.edu/~watermang/math_341/341book.pdf [June 2017].

[9] Alsultan, Arwa, and Kevin Warwick. "User-friendly free-text keystroke dynamics authentication for practical applications." In 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4658-4663. IEEE, 2013.

[10] Singh, Saurabh, and K. V. Arya. "Key classification: a new approach in free text keystroke authentication system." In Circuits, Communications and System (PACCS), 2011 Third Pacific-Asia Conference on, pp. 1-5. IEEE, 2011.

[11] Chantan, Charoon, Sukree Sinthupinyo, and Tippakorn Rungkasiri. "Improving Accuracy of Authentication Process via Short Free Text using Bayesian Network." International Journal of Computer Science Issues (IJCSI), vol 9, no. 2 (2012).

[12] Dowland, P., H. Singh, and S. Furnell. "A preliminary investigation of user authentication using continuous keystroke analysis." InProceedings of the International Conference on Information Security Management and Small Systems Security, pp. 215-226. 2001.

[13] Gunetti, Daniele, and Claudia Picardi. "Keystroke analysis of free text." ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 3 (2005), pp. 312-347.

[14] Sim, Terence, and Rajkumar Janakiraman. "Are digraphs good for free-text keystroke dynamics?." In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-6. IEEE, 2007.

[15] Malas, Tareq M., Sinan S. Taifour, and Gheith A. Abandah. "Toward optimal Arabic keyboard layout using genetic algorithm." Proc. 9th Int'l Middle Eastern Multiconf. on Simulation and Modeling (MESM 2008) (2008): 50-54.

[16] Villani, Mary, Charles Tappert, Giang Ngo, Justin Simone, H. St Fort, and Sung-Hyuk Cha. "Keystroke biometric recognition studies on long-text input under

ideal and application-oriented conditions." In 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pp. 39-39. IEEE, 2006.

[17] Alsuhibany, S. A., Almushyti, M., Alghasham, N., & Alkhudier, F. "Analysis of free-text keystroke dynamics for Arabic language using Euclidean distance." In Innovations in Information Technology (IIT), 2016 12th International Conference on (pp. 1-6). IEEE.

[18] Katzner, K., 2002. The languages of the world. Routledge.