International Journal of Engineering Research and Technology. ISSN 0974-3154 Vol.13, No.4 (2020), pp. 625-630
 https://dx.doi.org/10.37624/IJERT/13.4.2020.625-630

A Proposed Methodology for Detecting Human Attacks on Text-based CAPTCHAs

Latifah A. Alreshoodi[1] and Suliman A. Alsuhibany[2]

Computer Science Department, College of Computer, Qassim University, Buridah, Saudi Arabia.
ORCIDs: 0000-0002-6473-915X (Latifah), 0000-0001-7735-9781(Suliman)

Abstract

A CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) is a simple test that is applied on websites to differentiate between human users and automated programs, which indulge in spamming and other fraudulent activities. A text-based CAPTCHA is the most popular security technique used by many websites on the internet, such as Microsoft, eBay, Google and so on. A text-based CAPTCHA human attack means hiring third-party humans in order to solve the CAPTCHA tests. Consequently, a CAPTCHA, by design, is unable to differentiate between a human-based attacker and a legitimate human user. This paper proposes a new methodology for detecting human attacks on text-based CAPTCHAs.

Keywords: Text-based CAPTCHA, Keystroke Dynamic, Authentication, Network Security.

I. INTRODUCTION

With the growth in use of internet resources nowadays and increasing threats to the security of the World Wide Web, we need a secure way to protect online resources and services from attacks. A CAPTCHA (Completely Automated Public Turing Tests to Tell Computers and Humans Apart) is one of the secure techniques commonly used on the internet nowadays. It is used to differentiate between a human user and malicious bots (i.e. automated computer programs) [1].

There are many types of CAPTCHA, such as audio-based CAPTCHAs, image-based CAPTCHAs and text-based CAPTCHAs. Text-based CAPTCHA is the most popular security technique used by many websites on the internet, for example Microsoft, eBay, Google and so on, to secure their websites from bots. A text-based CAPTCHA automatically generates a test on an image in a form, such as a registration form, where that image contains some digits, upper- and lower-case letters distorted, for the users to solve. It is designed to be easy for a human to solve but difficult for a computer to solve [2]. Accordingly, this can allow a human attacker to solve the CAPTCHA correctly, so the CAPTCHA can be broken easily.

Human-based attack means malicious industries hiring third-party humans to conspire with attackers to solve the CAPTCHA tests [3]. Hence, the CAPTCHA system is no longer secure. Thus, we need an additional secure technique to differentiate between a legitimate user and a human attacker.

Some papers, such as [4] and [3], call human-based attack a third-party human attack. A third-party human attack tries to solve CAPTCHA challenges by redirecting a presented CAPTCHA to third-party users to help attackers break it [4]. It is important to note that this kind of attack is different from the man-in-the-middle attack that can be defined as a malicious bot eavesdropping on the communication channel between the two parties, which are the client and the server. When a client attempts to contact the server, a malicious bot intercepts the client's message, removes it from the communication channel, makes a fake client message and sends it to the server. In addition, it may impersonate the legitimate server and send the answers back to the client [5].

There are three methods of person authentication: possessions, knowledge and biometric. A possession is a unique item that the user has with him, like a passport, or a smartcard. These can be shared, duplicated, or maybe even lost or stolen. Secondly, knowledge is some secret information such as a password. Although this method is widely used, many passwords are easy to guess, shared with others, or may be forgotten. Finally, a biometric measure is a unique human characteristic or trait. Further, biometric can be classified as physiological and behavioral. Physiological modalities are related to the shape of the human body, like a fingerprint, face, DNA, hand geometry, or iris. Behavioral modalities are related to the behavior pattern of a person, like signature, gait, voice, keystroke, or mouse movement [6].

Keystroke dynamics, which are the user typing patterns, have neurophysiological factors that make them unique from others. A unique keystroke profile can be created from different typing features, such as typing speed, the duration between successive keys pressed, pressure applied on the keys, and finger positions on the keys [6]. Recognition based on the unique typing pattern is non-intrusive, cost-efficient and transparent to the user [7]. Furthermore, it is very easy to capture data as keyboards are common and no special hardware is necessary [8]. This type of biometrics will be applied in our proposal to detect human-based attacks.

Human-based attack on CAPTCHAs has been studied in [9], [4], [3] and [10] to differentiate between a legitimate user and an attacker user by using different approaches. However, using keystroke dynamics as a defense approach against this problem has not yet been considered. Therefore, this paper investigates the application of this approach by developing a text-based CAPTCHA that includes a keystroke dynamics tool as one system. We aim to provide a secure text-based CAPTCHA system to protect web resources and services

against human attackers by using keystroke dynamics, which are believed to be unique for each user, in order to authenticate the legitimate users and detect human-based attacks. To the best of our knowledge, this proposed approach has not been investigated before. The evaluation of this proposed approach will be one of our future works.

The rest of this paper is structured as follows. Section II discusses the related works. Section III explains the proposed methodology. Section IV concludes the paper with proposals for future works.

II. BACKGROUND AND RELATED WORKS

This section highlights the background of CAPTCHA and keystroke dynamics in general. Many related works are also highlighted.

1) Background

A CAPTCHA is a security technique used to distinguish between humans and computers. It has been used to defend against malicious software [1]. There are many categories of CAPTCHA, such as audio-based CAPTCHAs, image-based CAPTCHAs and text-based CAPTCHAs. Text-based CAPTCHA is the most popular security technique used by many websites on the internet, for example Microsoft, eBay, Google and so on, to secure their websites from bots. A text-based CAPTCHA automatically generates a test on an image in a form, such as a registration form, where that image contains some digits, upper- and lower-case letters distorted, for the user to solve. This test is designed to be easy for the user to solve but difficult for a computer to solve. If the user solves the test correctly, it is confirmed that he is a human not a computer and registration is successful [2].

We should design text-based CAPTCHAs to be segmentation-resistant, as the robustness of text-based CAPTCHAs depends on the difficulty of detecting "where" each character is instead of "what" it is. Computers can recognize individual characters better than humans. The CAPTCHA is effectively broken if it is vulnerable to a segmentation attack [1].

There are some weaknesses of text-based CAPTCHAs. One of the common problems is that they can be recognized by the OCR (Optical Character Recognition) technique because the number of characters and numbers is very small. If we add noise or distortion to text-based CAPTCHAs, the recognition process can be difficult by OCR. Another problem that may appear if we add some distortion to the CAPTCHAs is the confusion between some characters or numbers, such as between "cl" and "d" (i.e. the letters may be unreadable) [11].

There are some guidelines for designing robust text-based CAPTCHAs provided in [1] after analyzing and evaluating their attack results on the CAPTCHAs adopted by the top 20 websites in Alexa. These guidelines are: increase the CAPTCHA string length; increase the differences in character widths; increase the alphabet size; use a short expiry time for every image, with a rapid refreshment rate; make confusion in the background; and insert noise into the CAPTCHA. While these techniques can make attacks hard, it also may reduce the usability by decreasing the recognition rate by humans at the same time. This may be a problem because we have to make a balance between security and usability when we design a CAPTCHA [1].

One of the alternative authentication methods is keystroke dynamics, which is a behavior-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, keystroke dynamics is, as stated in [12], "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 [7].

So far, keystroke dynamics has two main categories: free-text and fixed-text [13]. Free-text uses the typing patterns of the user without entering a predefined text, while fixed-text uses the typing patterns of the user by entering a predefined text. In the latter, the user needs a training session and then remembers the text at the log-in time. Conversely, free-text overcomes the problem of memorizing the text, as the text used for the enrollment session does not have to be the same as the text used for the log-in session. Furthermore, free-text can be used in a number of applications, such as enhancing security by continuous and nonintrusive authentication [14]. Thus, this project focuses on free-text only.

2) Related works

Most of the existing CAPTCHA systems are vulnerable to a third-party human attack because they only differentiate between humans and bots and cannot differentiate between a legitimate user and a human attacker. A third-party human attack tries to solve CAPTCHA challenges by redirecting a CAPTCHA to third-party users to help attackers break it. Therefore, these CAPTCHA systems will not be secure. Some research has been done to protect CAPTCHA against third-party human attacks, for example [4], which designed the Drag-n-Drop Interactive Masking CAPTCHA (DDIM CAPTCHA). The authors applied drag-n-drop, interaction, and masking techniques with text-based CAPTCHA to differentiate legitimate users from computer bots and third-party human users. These techniques made their CAPTCHA system able to resist traditional attacks and third-party human attacks.

Also, in [3], the authors propose GeoCAPTCHA to defend against third-party human attack by combining personalized information with image-based CAPTCHA. The answer is only known by the user so it can successfully prevent human attacks. A user has to remember a geolocation street-view scene image that is pre-selected by him and pre-registered with the server. To pass the test, he has to rotate a given street-view challenge to match the preregistered scene image. Hence, the technique can determine if the user is a third-party solver or not.

In addition, a set of research papers used behavioral biometrics integrated with CAPTCHA systems; for example, in [10] they used image-based CAPTCHA integrated with Mouse Dynamics. This technique is one of the behavioral biometrics, in that it monitors the mouse interaction for a

human user. To solve the challenge, the user must identify and select a certain class of images. While the user tries to solve the CAPTCHA, the ways in which he/she interacts with the mouse, i.e. mouse clicks, mouse movements, mouse cursor screen coordinates, etc. are recorded. These recorded mouse movements form the Mouse Dynamics Signature (MDS) of the user. This MDS gives an extra secure technique to differentiate humans from bots. The authors tested the security of the CAPTCHA by an adversary executing a mouse bot attempting to solve the CAPTCHA challenges. They observed that their linear support-vector machine (SVM) classifier performed well in detecting the bot with 100% accuracy, whereas it had an accuracy of close to 86% in detecting humans attempting to solve the CAPTCHA samples.

Finally, a study in [9] proposed Interactive CAPTCHA (iCAPTCHA), which is a text-based CAPTCHA to defend against third-party human attacks. First, they developed a streamlined human-based CAPTCHA attack that used Instant Messenger (IM) infrastructure to clarify the threat of the human solver attacks. This attack allows iCAPTCHA challenges to be delivered to third-party human solvers by IM technology at speeds that defy detection by CAPTCHA timeout values (30 seconds). Finally, they proposed a defense system called iCAPTCHA, which requires a user to solve a CAPTCHA test by a series of user interactions. The multi-step back-and-forth traffic between client and server amplifies the statistical timing difference between a legitimate user and a human solver, which enables better attack detection performance.

In sum, a CAPTCHA system still needs considerable research and enhancement to defend against a human attacker, in addition to being more secure and usable. However, to the best of our knowledge, there has not been a study done in the area of differentiating between a legitimate user and a human attacker for text-based CAPTCHA using keystroke dynamics (behavioral biometrics). Therefore, this paper introduces the application of this approach by developing a text-based CAPTCHA that includes a keystroke dynamics tool as one system.

III. PROPOSED METHODOLOGY

In this section, we describe our proposed system and how it can detect human-based attacks. As we mentioned previously, our proposed system is a text-based CAPTCHA that includes a keystroke dynamics system as one system to detect human-based attackers.

1) An overview

The proposed methodology contains a web page that includes a text-based CAPTCHA image with a text box. The CAPTCHA image is generated by a text-based CAPTCHA generator. The user is asked to type the CAPTCHA word, which appears in the image, in the text box. Then, on the client side, the system will record the raw keystroke times while the user is typing the CAPTCHA word, then generate the time features that represent the typing patterns for the user and send them to the server side in order to store them in the database as a user profile.

The application also captures the IP address of the user, which will be stored with the time features in the database as an identifier for each user. We were inspired by [15], as they used IP addresses to reduce attacks by limiting the number of attempts to solve the CAPTCHA for each IP address.

The application allows the user to solve a CAPTCHA test successfully up to 99 times in an hour, and consider him as a legitimate user. At 99 times, the enrollment phase is ended, and the verification phase is started at 100 times; this is inspired by the detection method of spam email that has been proposed in [16]. In addition to this study, a study in [17] mentioned that a real human attacker solves 1,000 CAPTCHA tests in an hour.

In the verification phase, the application tries to detect the human attacker by checking if the user is a human attacker who solved the previous tests and is trying to attack the application. This verification is done by using keystroke dynamics.

2) The text-based CAPTCHA generator

The text-based CAPTCHA type is the most common type of CAPTCHA used by most major websites such as Google, Microsoft and Amazon. Furthermore, the popularity of this type is due to its simplicity to implement [18] and it is easy to solve by users worldwide without much instruction as it requires recognizing characters and/or digits only [1].

Hence, a text-based CAPTCHA generator is proposed that can randomly generate a word containing *ten* English characters and/or Arabic digits (A–Z, a–z, 1–9). In addition, the length of the word is selected to be *ten* characters. The reason for this is to get more timing data for keystroke dynamics authentication; the study in [19] concluded that using a text length of *ten* characters for keystroke dynamics authentication is typical. Therefore, we can classify users more accurately and detect a human attacker. The generated CAPTCHA word will be printed on a background image. It will then be displayed to the user on a web page. For a new request, the generator generates a new CAPTCHA for each request from the user.

It is interesting to note that in our proposed generator, we have focused more on usability features than security features such as anti-segmentation and anti-recognition features. Although the security features are very important in terms of protecting against automated attacks, they are waived intentionally here in our paper only for testing our proposed system (i.e. applying the keystroke dynamics approach) against human-based attacks.

3) Methodology

In the proposed methodology, since the authentication means ensuring that the user is legal and authorized [8], we have applied the keystroke dynamics authentication system for detecting a human-based attack for text-based CAPTCHA. This is done by verifying the identity of the user and differentiating a legitimate human user from an attacker user. Additionally, we have chosen the keystroke dynamics authentication system to detect a human-based attack on text-based CAPTCHA for the following advantages [20]:

- A user's typing patterns are unique and are hard to reproduce as the keystroke dynamics system can calculate keystroke action up to millisecond precision.
- Typing patterns cannot be lost or shared.
- It is easy to integrate with other existing systems without requiring extra hardware.
- It is considered the least expensive biometric authentication method. This is because it is a software-based technique and requires only a keyboard, which is a necessary part of any computer [8].
- It is transparent and non-intrusive, as the user's typing pattern on a keyboard can be collected without the knowledge of the user. Therefore, there is no need to change the user behavior and no need for additional work to be done by the user to authenticate [7].

In order to detect human-based attackers for text-based CAPTCHA, the proposed methodology consists of two phases: enrollment and verification. In the former phase, the user is asked to solve the CAPTCHA test several times. Specifically, the user will type the CAPTCHA word, which appears in the image, in the text box. Meanwhile, the system collects the raw keystroke timing data (i.e. the press and release timestamp of each key typed to solve the CAPTCHA) and extracts the timing features that represent the typing patterns of the user. If the user's answer is correct, then the system will calculate the average for each time feature to create the time vector. This phase starts from attempt 1 to solve the CAPTCHA test until attempt 99, as explained previously.

In the latter phase, the keystroke dynamics system tries to detect the human attacker by verifying if the previous CAPTCHA tests are solved by the same user who has the same IP. This phase starts when the user reaches attempt number 100 to solve the CAPTCHA. Thus, the system is collecting the user's keystroke times and then extracting the time features and creating the user's profile in the same way as in the enrollment phase. The system then computes the Euclidean distance [21] between the new user's profile and the stored user profiles, which have the same IP address. Based on the computed results, if it falls within the threshold, which is the standard deviation (SD) of the stored user's profile, then the users' profiles are similar and belong to the same user. Then, if the number of similar user profiles are equal or more than a similarity threshold, which is an empirical threshold that will be determined in a primary experiment, the user will be classified as an attacker, blocked and his IP address added to the block list. Otherwise, the user will be classified as a legitimate user, his typing information will be stored in the database and he will be granted access. Figure 1 shows an overview of these phases.

It is interesting to note that our keystroke dynamics authentication system includes four stages: data collection; feature extraction and file creation; classification method; and evaluation. We have four stages in our system because the decision-making stage is included in the classification stage. Also, some of the keystroke dynamics systems do not include the retraining stage [6].

Moreover, the features of keystroke dynamics can be calculated from key press timestamp and key release timestamp in milliseconds for each key typed. Furthermore, we propose three di-graph timing features in our paper, as suggested in [22]:

- Hold time or dwell time: This is the time period between a key being pressed and released. For example, the hold time for key 1 = release time of key 1 – press time of key 1.
- Flight time (or latencies): We propose to apply two types:
  a. Down–Down (DD) (or Press–Press) time: This is the time difference between a key press and a press of the next key.
  b. Up–Down (UD) (or Release–Press) time: This is the time difference between a key release and a press of the next key.

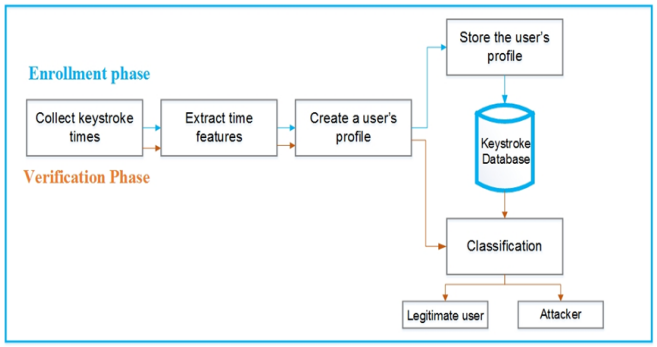

Fig. 1. Enrollment and verification phases of the proposed system

Figure 2 shows an example of extracting the used keystroke time features (Down–Down time, Up–Down time and Hold time) for two keys. Afterward, the system will compute the average for each time feature, i.e. the Down–Down time, Up–Down time and Hold time, to form a timing vector that will be stored in the database as the user's profile to be used for classification, as suggested in [22].

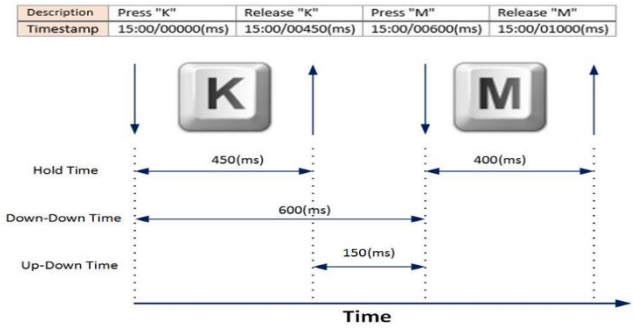

Fig. 2. An example of extracting the used keystroke time features (Down–Down time, Up–Down time and Hold time) for two keys.

We propose the Euclidean distance [21] to be applied as a classification method for user profiles of typing patterns in our paper. This is because it is a simple statistical method and has been used as a classification method in several studies of keystroke dynamics, such as [22] and [23]. Also, the performance of the classification method was very good – in [23] they achieved 0.02% FAR (False Acceptance Rate) and 0.04% FRR (False Rejected Rate) and in [22] they achieved 0.22% FAR and 0.0 % FRR, for English text.

We have to find a threshold for Euclidean distance to classify the data and identify the users. We propose the SD of a user's profile vector as the threshold among two time vectors of the user, as used in [22]. SD is commonly used to measure the variability or diversity of a set of values [24].

IV. CONCLUSION AND FUTURE WORKS

A CAPTCHA is a simple test that is applied on websites to differentiate between human users and automated programs, which indulge in spamming and other fraudulent activities. A text-based CAPTCHA automatically generates a test on an image. This test is designed to be easy for users to solve but difficult for computers to solve. A text-based CAPTCHA human attack is when malicious industries hire third-party humans to conspire with attackers to solve the CAPTCHA tests. Therefore, a CAPTCHA, by design, is unable to differentiate between a human-based attacker and a legitimate human user. Therefore, this paper proposes a new methodology for detecting human attacks of text-based CAPTCHAs using the keystroke dynamics approach. This methodology is based on timing features that can be extracted from the time lapses between two actions on the keyboard.

One of our future works will be implementing and evaluating the proposed methodology. Furthermore, it will be interesting to investigate anti-automated attack tools in order to be involved in the proposed text-based CAPTCHA generator.

ACKNOWLEDGMENT

The authors would like to thanks Qassim University for supporting this research.

REFERENCES

[1] H. Gao, X. Wang, L. Lei, X. Liu, Z. Zhang, F. Cao, and J. Qi, "Robustness of text-based completely automated public turing test to tell computers and humans apart," IET Information Security, vol. 10, no. 1, pp. 45–52, 2016.

[2] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," Commun. ACM, vol. 47, no. 2, pp. 56–60, 2004.

[3] T. E. Wei, A. B. Jeng, and H. M. Lee, "GeoCAPTCHA - A novel personalized CAPTCHA using geographic concept to defend against 3rd party human attack," 2012 IEEE 31st Int. Perform. Comput. Commun. Conf. IPCCC 2012, pp. 392–399, 2012.

[4] Q. Bin Ye, T. E. Wei, A. B. Jeng, H. M. Lee, and K. P. Wu, "DDIM-CAPTCHA: A novel drag-n-drop interactive masking CAPTCHA against the third party human attacks," Proc. - 2013 Conf. Technol. Appl. Artif. Intell. TAAI 2013, pp. 158–163, 2013.

[5] U. Ferraro Petrillo, G. Mastroianni, and I. Visconti, "The design and implementation of a secure CAPTCHA against man-in-the-middle attacks," Secur. Commun. NETWORKS, vol. 7, no. 8, pp. 1199–1209, 2014.

[6] M. L. Ali, J. V. Monaco, C. C. Tappert, and M. Qiu, "Keystroke Biometric Systems for User Authentication," J. Signal Process. Syst., vol. 86, no. 2–3, pp. 175–190, 2017.

[7] A. Alsultan and K. Warwick, "Keystroke Dynamics Authentication : A Survey of Free-text Methods," Int. J. Comput. Sci. Issues, vol. 10, no. 4, pp. 1–11, 2013.

[8] G. Al-naymat, M. Al-kasassbeh, A. Hassanat, and A. Al-tarawneh, "Dynamics-Based Approach for Accurate User Identification and Authentication using Machine Learning Techniques," Int. Conf. Inf. Technol. Appl., 2016.

[9] H. D. Truong, C. F. Turner, and C. C. Zou, "iCAPTCHA : The Next Generation of CAPTCHA Designed to Defend Against 3rd Party Human Attacks," Commun. (ICC), 2011 IEEE Int. Conf., pp. 1–6, 2011.

[10] D. F. D. Souza, "Avatar captcha : telling computers and humans apart via face classification and mouse dynamics," (2014). Electronic Theses and Dissertations. Paper 1715.

[11] R. U. Rahman, "SURVEY ON CAPTCHA SYSTEMS," J. Glob. Res. Comput. Sci., vol. 3, no. 6, pp. 54–58, 2012.

[12] F. Monrose and A. Rubin, "Authentication via Keystroke Dynamics," 4th ACM Conf. Comput. Commun. Secur., pp. 48–56, 1997.

[13] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics : A review," Appl. Soft Comput. J., vol. 11, no. 2, pp. 1565–1573, 2011.

[14] P. Bours and S. Mondal, "Performance evaluation of continuous authentication systems," IET Biometrics, vol. 4, no. 4, pp. 220–226, 2015.

[15] E. Bursztein and S. Bethard, "Decaptcha: Breaking 75% of eBay audio CAPTChas," 3rd USENIX Work. Offensive Technol. WOOT 2009, pp. 1–7, 2009.

[16] K. Yoshida et al., "Density-based spam detector," KDD-2004 - Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 486–493, 2004.

[17] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: Captchas – Understanding CAPTCHA-solving services in an economic context," Proc. 19th USENIX Secur. Symp., pp. 435–452, 2010.

[18] V. P. Singh and P. Pal, "Survey of Different Types of CAPTCHA," Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 2, pp. 2242–2245, 2014.

[19] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," Proc. Int. Conf. Dependable Syst. Networks, pp. 125–134, 2009.

[20] P. S. Teh, A. J. Teoh, and S. Yue, "A Survey of Keystroke Dynamics Biometrics," Sci. World J., vol. 2013, 2013.

[21] G. Waterman, "Euclidean Space and Vectors," in Linear Algebra I, Oregon Institute of Technology, 2007, pp. 43–59.

[22] S. A. Alsuhibany, M. Almushyti, N. Alghasham, and F. Alkhudier, "Analysis of free-Text keystroke dynamics for Arabic language using Euclidean distance," Proc. 2016 12th Int. Conf. Innov. Inf. Technol. IIT 2016, pp. 185–190, 2017.

[23] S. Singh and K. V. Arya, "Key classification: A new approach in free text keystroke authentication system," Proc. - PACCS 2011 2011 3rd Pacific-Asia Conf. Circuits, Commun. Syst., pp. 1–5, 2011.

[24] M. Rouaud, "Probability, statistics, and estimation: propagation of uncertainties in experimental measurement," in Probability, statistics, and estimation, 2017, pp. 3–6.