Generating Arabic Handwritten CAPTCHA for Cyber Security

Suliman A. Alsuhibany

Department of Computer Science, Qassim University, Buridah, Saudi Arabia

Abstract
The exponential growth of electronic services has led to the misuse of automated bots, which has resulted in serious security issues in terms of these services. Therefore, a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a test to limit the ability of cyber attackers when it comes to scaling their activities using automated bots. Since several existing Latin script-based CAPTCHAs have been broken, this paper proposes a method for using Arabic handwritten text to generate CAPTCHAs. The rationale behind this method is that Arabic script and handwritten text have a number of advantages that may be beneficial for cyber security. Accordingly, an Arabic handwritten CAPTCHA generator is developed that produces Arabic handwritten CAPTCHAs. In order to assess the proposed method, experimental studies are conducted. The results showed a gap between machine and human recognition abilities while using Arabic handwritten script.

Key words:
Arabic script; Handwriting; CAPTCHA; Cyber security; Web security

1. Introduction

The websites have gradually become a platform for the organization of complex applications. For example, a large amount of today's internet financial systems revolve around advertising revenue. Consequently, a vast array of services – including social networking and email – are now available to new users on a basis that is both free and largely anonymous. This highly dynamic and evolving context leads to problems. In particular, attackers have sought to exploit this freedom, and acquire large numbers of resources under singular control, which can in turn be monetized. That is, attackers try to misuse automatically the resource in question, e.g. by sending thousands of free e-mail accounts for sourcing spam email messages. As a result, a Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is developed in [1] as an approach to reduce the damage caused by attackers. Text-based CAPTCHAs are the most commonly-deployed type due to their many advantages [2]. Therefore, the term CAPTCHA in this paper refers to text-based schemes only.

Most websites nowadays adopt a text-based type that is based on Latin script [25]. Indeed, this type has a great deal of vulnerability as reported in, for example, [26-28]. Consequently, the use of handwritten text as a CAPTCHA image has been considered as a revolutionary idea by many researchers [10-19]. In particular, a groundbreaking paper was published by Rusu and Govidaraju [10] as a first step towards a handwritten CAPTCHA. That is, handwritten texts consisting of familiar text can be recognized by human beings more readily than unfamiliar texts [20]. Moreover, the difference in the abilities of human beings and machines has led to the application of handwritten text as a CAPTCHA scheme.

Arabic typed CAPTCHAs have attracted attention in a number of studies [3-8]. This is because CAPTCHAs based on Arabic script can be more convenient and readable for Arab users due to the fact that using CAPTCHAs based on the native language are more convenient as stated in [29]. With regard to handwritten, there have been few investigative studies involving Arabic handwritten CAPTCHA. That is, the first attempt to implement "Arabic Handwritten" as a CAPTCHA was in [30] in which the authors generated CAPTCHA images using the KHATT database of offline Arabic handwritten text [24], while a study in [9] proposed a handwritten CAPTCHA approach with four different languages, including Arabic, based on separating handwritten characters.

This paper introduces an Arabic handwritten CAPTCHA generator. This generator produces Arabic handwritten samples, and then applies various distortions to make the generated samples unreadable by automatic computer programs. For this, a set of meaningful Arabic words that include all 28 Arabic characters in their contextual shapes are written by different writers. Test results showed a large gap between machine and human recognition abilities.

The remainder of this paper is structured as follows: Section 2 highlights some related works. An overview about Arabic script is given in Section 3. The methodology is explained in Section 4. Section 5 describes the evaluation process. The results are presented in Section 6 and discussed in Section 7. Section 8 concludes the paper.

Manuscript received March 5, 2018
Manuscript revised March 20, 2018

2. Related Works

Although a number of studies have considered Latin handwritten CAPTCHAs [10-19], this section basically reviews the related works on handwritten CAPTCHAs in general, and Arabic handwritten CAPTCHAs in particular.

2.1 Handwritten CAPTCHAs

The first step in this direction is proposed by Rusu and Govidaraju [10] in terms of the differential between humans and computers in reading handwritten text being investigated. This study is extended in [12]. The same authors explored Gestalt psychology in [13] as a motivation to handwritten CAPTCHA transformations. A synthetic handwriting generation method is discussed in [16, 17] and the results show that machine recognition rates decrease considerably when the text lines generated are further obfuscated with a set of deformations, while human recognition rates remain the same. In [14], the authors investigated the fact that some recognition tasks are significantly easier for humans than for machines. The generation and use of handwritten CAPTCHAs are explored in [18]. In this study, the parameters that truly separate human and machine abilities are investigated. A novel method is presented in [19] which combines handwritten CAPTCHAs with a random tree structure and test questions. The tree and shape CAPTCHAs, in addition to handwritten CAPTCHAs, are described in [15]. The difficulty of recognizing handwritten text by an automated device is utilized in [17] as a key for securing CAPTCHA formation. Finally, an enhanced scheme involving coloured handwritten CAPTCHA is proposed in [11].

2.2 Arabic Handwritten CAPTCHAs

There have been few studies done investigating the application of Arabic Handwritten CAPTCHAs. The first attempt to implement an Arabic Handwritten approach as a CAPTCHA is described in [30]. This study proposed a novel approach by exploiting the PAWs (Parts of Arabic Words) and breaking Arabic connection rules by distorting connected characters, and displacing them from baselines. A finite dataset was used (i.e. a KHATT dataset). However, this can be a flaw as an adversary with access to the dataset can use dictionary/lexican-based brute force attacks to circumvent the system. Moreover, a study in [9] proposed a handwritten CAPTCHA of four different languages: English, Arabic, Spanish, and French. They proposed a handwritten CAPTCHA with separate letters, even for Arabic that is cursive in nature. However, these studies do not exactly fit with our approach.

3. Arabic Script: An overview

This section explains the characteristics of Arabic language in terms of writing direction, shapes and recognition. In particular, Arabic language has 28 letters, and Arabic writing is someway like Latin [21]. Although Arabic numbers are written from left to right, Arabic letters are written from right to left. A word includes a set of different letters. Moreover, the letters are connected during writing both in printed and handwritten texts.

Generally speaking, there are four different forms of writing a letter in Arabic language depending on its position in the word. For instance, the form of the letter *Sean* can be either "س", "ـس", "ـسـ", or "سـ", where it can be a single letter, at the end of the word, between two letters or in the beginning of the word, respectively. Moreover, several Arabic characters have similar shapes, e.g. ب ت ث ن , ج ح خ, ص ض, ع غ, ط ظ, د ذ ر ز و, ف ق. As stated in [22], this similarity confuses the OCR (Optical Character Recognition) to recognise characters correctly.

In contrast to Latin script, there are various features in Arabic scripts which make recognition process relatively more difficult. In Arabic writing, the lack of space between words is one of these features as words are not usually separated by space. This, therefore, makes not only the recognition process difficult, but also the segmentation phase in both printed and handwritten Arabic text [23]. Moreover, diacritics are used in Arabic text, e.g. shadda, maddah, hamza, tanween [6]. In addition, when typing Arabic text, there can be overlapping between characters in terms of space, e.g. "وا" in which "ا" overlaps with "و". This overlapping as a feature makes both recognition and segmentation process difficult as demonstrated in [27]. Mostly, Arabic OCRs are developed based on a few font types, thus when a text is written in a different font type, it is unable to be recognized [5].

4. Methodology

The motivation behind the methodology used, the character set, and the generation mechanism are discussed in this section.

4.1 Gestalt laws and Geon theory

The methodology used in this paper is motivated by the Gestalt laws of perception and by Geon theory. The Gestalt laws [31] are a set of principles in psychology that are organised into five categories: Proximity, Similarity, Continuity, Closure, and Connectedness. Thus, it is based on the observation that we often experience things that are not part of our simple sensations. This is similar to the holistic word-recognition methodologies that rely on recognising the

entire word as a group, without breaking it into unites of characters. More details can be found in [18].
The Geon theory of recognition by components [32] provides cues with regard to what it is desirable to be preserved in image reconstruction, namely, edges and intersections. Since humans recognise objects from different point of views, even views that have never been seen before, this theory explains how moderately occluded or degraded images, and new instances of subjects, are successfully recognised by the human visual system.

4.2 Dataset

In order to construct the generator dataset, six different writers were asked to write a set of meaningful Arabic words in which all the Arabic letters (i.e. 28 letters) are included in their contextual shapes: beginning, middle, end and isolated. These words are used as a basis for the Arabic handwritten character samples used in this paper. After cropping the written words, the character samples are transformed into digital format by scanning them as groups, each group representing a specific writer. For example, Fig. 1 shows the letter “Kaf” written by different writers, while Fig. 2 shows the letter “Seen” in different shapes (i.e. in different positions).

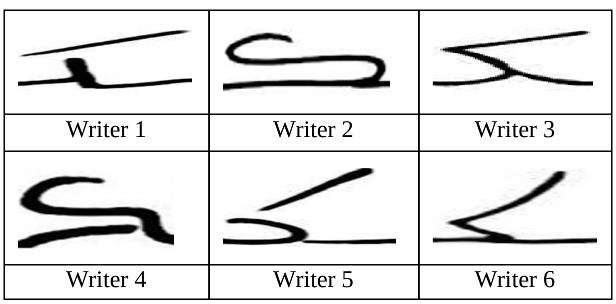


Fig. 1 The letter ’Kaf’ written by different writers

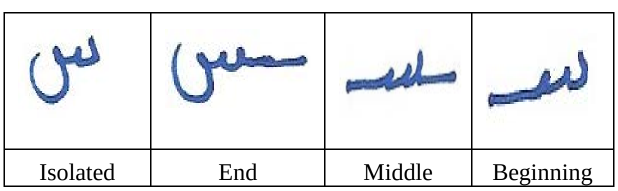


Fig. 2 The letter “Seen” written in different shapes

4.3 Arabic Handwritten CAPTCHA Generator

We developed a generation algorithm that generates pseudo words in a cursive Arabic handwritten style. These pseudo words, therefore, are distorted in different ways in order to make a final CAPTCHA challenge more difficult to automated bots. The following sub-section details the generation steps.

Generation Algorithm

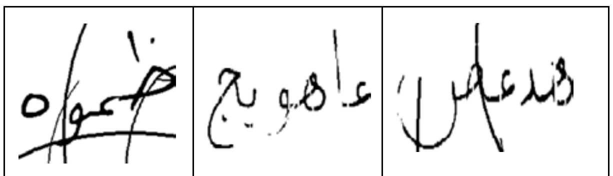
The Arabic handwritten generation algorithm that is developed in this paper involves six steps: (1) pseudo word generation, (2) converting characters to Unicode, (3) selecting a writer, (4) selecting a character image, (5) detecting joint points and connections, and finally (6) binarization. These steps are discussed in detail as follows:

- *Pseudo word generation:* Firstly, the algorithm generates a pseudo word by selecting a random number of Arabic letters, ranging from 3 to 8 letters. This range is found to be usable and secure in [27, 38].
- *Converting characters to Unicode:* Each character is converted to its Unicode by using the “word2uni” function [33]. This is an open source function that converts typed Arabic letters to their Unicode symbols. Thus, this function is used by our generation algorithm to return the correct Unicode of each letter, based on its position in the pseudo word.
- *Selecting a writer*: As was mentioned previously, a set of writers were asked to write a number of meaningful Arabic words. Therefore, the algorithm selects either a specific writer or a random writer in order to generate a CAPTCHA sample.
- *Selecting a character image:* Based on the chosen writer for each character, the image of each character is returned in order to be used in making the handwritten Arabic word. Moreover, we developed a background width function that automatically calculates the widths of each selected character image. In addition, it calculates the total width of all characters’ images. Once all characters’ images are selected, the background image is created. This includes several handwritten character images. In particular, the background image has a white background and a fixed high with 200 pixels, while the width depends on the returned value from the background width function.
- *Detecting joint points and connection:* Arabic writing reads from right to left. However, in our algorithm, the collection process of the characters’ images is achieved in the reverse direction (i.e. from left to right). That is, the generator firstly checks if the two adjacent letters are interconnected. If so, then if the left letter is in the middle, end or in isolated form, and connected to its predecessor’s beginning or middle form, then the algorithm searches vertically on the sides of the characters’ images to detect the position of the joint points. After determining the position of the joint points for each letter, the generator calculates the difference between the joint point’s heights, and changes the vertical position of the right image based on a calculated value.
- *Binarization:* The generated characters are in black and

white colours. However, they have some gray pixels around the characters' strokes. As the distortions used are in black or white only, the gray pixels can be considered as a security breach that facilitates the differentiation between the characters' strokes and the distortions used [27]. Therefore, the generated images are binarised to convert these gray pixels to either black or white, based on a determined threshold.

Distortion

The final step is to add noise and distortions to the final image. One of the most popular distortions when using CAPTCHAs is the *arc* technique. For this, we developed an algorithm that adds a number of arcs with different length and width to the generated CAPTCHA image. Since the generated images are in black and white, we decided to use three types of arc: black, white or a combination of both, as shown in Fig. 3.



Black arcs	White arcs	Black & white arcs

Fig. 3 Samples of applying different arc types

5. Performance Evaluation

The robustness and usability aspects of the generated samples are evaluated in this section. That is, for the security evaluation, different attacks that are usually used in breaking text-based CAPTCHAs [e.g. 26, 27, and 28] are utilized for evaluating the generated samples, whereas, for the usability evaluation, an experimental study is carried out.

Security evaluation

The attacks that have been applied in our evaluation include three main steps: (1) pre-processing, (2) segmentation, and (3) recognition attacks. For this, 60 samples are randomly generated from the same and different writers and categorized, based on their distortion: samples with black arc distortion, white arc distortion, and white and black arc distortion.

- **Pre-processing**

This is the first step in breaking CAPTCHAs where as much as possible of the background noise is removed. This step is accomplished by the commercial editor of automatic CAPTCHA solving software (GSA CAPTCHA Breaker) to pre-process the generated samples [34].

- **Segmentation**

In this step, only the samples that passed the pre-processing step were fed into the segmentation algorithm in order to extract the word's characters of the generated samples correctly, and in the right order. To do this, the Joint Point Segmentation algorithm was developed especially for evaluating Arabic script. This algorithm seeks to detect the joint points between any two connected letters.

- **Recognition**

The OCR engines that support the Arabic language are utilized for recognizing the segmented samples. In particular, Tesseract [35], ABBYY [36], and newocr.com [37] engines were selected for this step.

Usability evaluation

An experimental study is conducted to collect data on user performance in a laboratory environment. For this, a user interface was developed, using the PHP programming language. There were a total of 134 participants, all of whom were native Arabic speakers. We generated and randomly selected 60 samples from the same and different writers, with arc distortions. In this study, two metrics are measured:

- **Accuracy**: This measures the correctness of the entered solutions of a given CAPTCHA by the user.
- **Average Response time**: This represents the average time in seconds taken by the user to solve a given CAPTCHA.

6. Results

The results in terms of both security and usability evaluations are presented in this section.

6.1 Results of Security Evaluation

The result of testing 60 samples as part of the pre-processing step was that 32 out of the 60 samples did not pass. This means that 53.3% failed the pre-processing step, while 46.7 % passed. Moreover, as the samples that did pass the pre-processing step (i.e. 28 samples) were fed into the Joint Point Segmentation algorithm, the result was that 98.3% of the samples were not segmented correctly. This means that the joint point between the connected letters was not detected correctly. Finally, the remaining samples (i.e. those which did not pass) were tested on different

OCRs. The results showed that none of the OCRs used could recognise the given samples.

6.2 Results of Usability Evaluation

All participants completed the given task successfully. Table 1 shows the accuracy and the average response time of all arcs types: black, white and black and white.
Moreover, regarding the writers of the presented sample, Table 2 shows the success rate and the average response time of both the same and different writers.

Table 1: The success rate and the average response time of all arcs types.

Arc type	Accuracy	Average Response Time
White	83.88%	10.23 sec.
Black	66.57%	13.21 sec.
Black and White	67.01%	15.38 sec.

Table 2: The success rate and the average response time based on the writer.

Writers	Success Rate	Average Response Time
Different writers	70.55%	12.93 sec.
Same writer	74.43%	12.77 sec.

7. Discussion

The results of the evaluation show that the introduction of Arabic handwritten letters for a CAPTCHA generator indicates a good success rate in terms of both security and usability aspects. In particular, as shown in Section 6.1, 98.3% of the generated samples were not segmented correctly. Moreover, all the generated samples were not recognised using selected OCR engines.

It is important to point out that the security evaluation of the generated samples was accomplished using both segmentation and recognition attacks. On the other hand, the study described in [9] used only a recognition attack to evaluate its generated samples; the reason behind this may be that the authors' approach was based on separating handwritten characters. Moreover, even though a study in [30] used both segmentation and recognition attacks to evaluate its samples, the segmentation algorithm attack used was already utilised in the generation phase in order to displace character form baselines; in addition, this algorithm was applied on PAWs (Parts of Arabic Word) that were extracted from the KHATT database.

The results of the recognition evaluation were remarkable. Although the success rate of the recognition attack was 0% using the selected OCRs, it might be interesting to apply a holistic word recognition system, as in [30], to determine the robustness level of the generated samples against this system. Moreover, in spite of the fact that most of the OCR engines used in [9] are used in our security evaluation, the recognition success rate in our study was far less than in [9]. Consequently, the generated samples seem robust against this kind of attack.

The type of distortion can play an important role in terms of usability results. That is, the accuracy was 83.8% in our study when using the white arcs, while this percentage decreased when using both black and black and white arcs. Moreover, the accuracy is affected by the writer of the generated sample; for example it was 70.5% with different writers' samples, while it was 74.4% with the same writers' samples. Nonetheless, the accuracy in [9] was 92% when applying either thinner or thicker arcs with different colours, warping and overlapping. Similarly, the accuracy was more than 88% in [30] when using horizontal and vertical displacements of randomly-selected characters, and a polyline approximation of all segments with random rotation.

8. Conclusion and Future work

This paper proposed an approach in which Arabic handwritten text is exploited to generate CAPTCHAs. Our work is based on the advantages of Arabic script and handwritten text. Therefore, Arabic handwritten CAPTCHA generator has been developed. Through experimental studies, we were able to evaluate the proposed approach in terms of its robustness and usability characteristics. The results showed that the robustness against such attacks and the usability were 98.3% and 83.8%, respectively.

As an extension of this study, we plan to add more distortion techniques at both the whole character level and the character level, and see to what extent the quality of generated CAPTCHAs may serve according to different scenarios.

Acknowledgments

The author would like gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, on the material support for this research under the number (1120-coc-2016-1-12-S) during the academic year 1438 AH / 2016 AD.
Also, we would like to thank Fatimah N. Almohaimeed and Naseem A. Alrobah for contributions to this paper.

References

[1] Von Ahn, Luis, Manuel Blum, and John Langford. "Telling humans and computers apart automatically." *Communications of the ACM* 47, no. 2 (2004): 56-60.

[2] Chellapilla, Kumar, Kevin Larson, Patrice Simard, and Mary Czerwinski. "Designing human friendly human interaction proofs (HIPs)." In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 711-720. ACM, 2005.

[3] Khan, Bilal, Khaled S. Alghathbar, Muhammad Khurram Khan, Abdullah M. AlKelabi, and Abdulaziz AlAjaji. "Using Arabic CAPTCHA for Cyber Security." In *Security Technology, Disaster Recovery and Business Continuity*, pp. 8-17. Springer, Berlin, Heidelberg, 2010.

[4] Khan, Bilal, Khaled Alghathbar, Muhammad Khurram Khan, Abdullah M. AlKelabi, and Abdulaziz Alajaji. "Cyber security using arabic captcha scheme." *Int. Arab J. Inf. Technol.* 10, no. 1 (2013): 76-84.

[5] Shirali-Shahreza, Mohammad Hassan, and Mohammad Shirali-Shahreza. "Persian/Arabic Baffletext CAPTCHA." *J. UCS* 12, no. 12 (2006): 1783-1796.

[6] Shirali-Shahreza, M. H., and M. Shirali-Shahreza. "Persian/Arabic CAPTCHA." In *Proceedings of the IADIS International Conference on Applied Computing*, pp. 258-265. 2006.

[7] Shirali-Shahreza, M. Hassan, and Mohammad Shirali-Shahreza. "Multilingual captcha." In *Computational Cybernetics, 2007. ICCC 2007. IEEE International Conference on*, pp. 135-139. IEEE, 2007.

[8] Shahreza, M. Shirali. "Verifying spam SMS by Arabic CAPTCHA." In *2nd IEEE International Conference on Information and Communication Technologies (ICTTA'06)*, pp. 78-83. 2006.

[9] Aldosari, Maha H., and Ahmad A. Al-Daraiseh. "Strong multilingual CAPTCHA based on handwritten characters." In *Information and Communication Systems (ICICS), 2016 7th International Conference on*, pp. 239-245. IEEE, 2016.

[10] Rusu, Amalia, and Venu Govindaraju. "Handwritten CAPTCHA: Using the difference in the abilities of humans and machines in reading handwritten words." In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pp. 226-231. IEEE, 2004.

[11] Rao, Mukta, and Nipur Singh. "Random Handwritten CAPTCHA: Web Security with a Difference." *International Journal of Information Technology and Computer Science (IJITCS)* 4 (2012): 53.

[12] Rusu, Amalia, and Venu Govindaraju. "A human interactive proof algorithm using handwriting recognition." In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pp. 967-971. IEEE, 2005.

[13] Rusu, Amalia, and Venu Govindaraju. "Visual CAPTCHA with handwritten image analysis." In *Human Interactive Proofs*, pp. 42-52. Springer, Berlin, Heidelberg, 2005.

[14] Rusu, Amalia I., and Venu Govindaraju. "On the challenges that handwritten text images pose to computers and new practical applications." In *Document Recognition and Retrieval XII*, vol. 5676, pp. 84-92. International Society for Optics and Photonics, 2005.

[15] Rusu, Amalia, and Rebecca Docimo. "Securing the web using human perception and visual object interpretation." In *Information Visualisation, 2009 13th International Conference*, pp. 613-618. IEEE, 2009.

[16] Thomas, Achint Oommen, Amalia Rusu, and Venu Govindaraju. "Synthetic handwritten captchas." *Pattern Recognition* 42, no. 12 (2009): 3365-3373.

[17] Achint, Thomas, and Govindaraju Venu. "Generation and performance evaluation of synthetic handwritten captchas." In *Proceedings of the First International Conference on Frontiers in Handwriting Recognition, ICFHR*. 2008.

[18] Rusu, Am, Achint Thomas, and Venu Govindaraju. "Generation and use of handwritten CAPTCHAs." *International Journal on Document Analysis and Recognition (IJDAR)* 13, no. 1 (2010): 49-64.

[19] Rusu, Amalia I., Rebecca Docimo, and Adrian Rusu. "Leveraging Cognitive Factors in Securing WWW with CAPTCHA." In *WebApps*. 2010.

[20] Johnston, J. C., O. J. L. Tzeng, and H. Singer. "Perception of print: Reading research in experimental psychology." (1981).

[21] Zheng, Liying, Abbas H. Hassin, and Xianglong Tang. "A new algorithm for machine printed Arabic character segmentation." *Pattern Recognition Letters* 25, no. 15 (2004): 1723-1729.

[22] Sattar, Sohail A., Shamsul Haque, Mahmood K. Pathan, and Quintin Gee. "Implementation challenges for nastaliq character recognition." In *International Multi Topic Conference*, pp. 279-285. Springer, Berlin, Heidelberg, 2008.

[23] Atallah, AL-Shatnawi, and Khairuddin Omar. "A comparative study between methods of Arabic baseline detection." In *Electrical Engineering and Informatics, 2009. ICEEI'09. International Conference on*, vol. 1, pp. 73-77. IEEE, 2009.

[24] Mahmoud, Sabri A., Irfan Ahmad, Wasfi G. Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, and Gernot A. Fink. "KHATT: An open Arabic offline handwritten text

database." *Pattern Recognition* 47, no. 3 (2014): 1096-1112.

[25] Yan, Jeff, and Ahmad Salah El Ahmad. "Usability of CAPTCHAs or usability issues in CAPTCHA design." In *Proceedings of the 4th symposium on Usable privacy and security*, pp. 44-52. ACM, 2008.

[26] Yan, Jeff, and Ahmad Salah El Ahmad. "A Low-cost Attack on a Microsoft CAPTCHA." In *Proceedings of the 15th ACM conference on Computer and communications security*, pp. 543-554. ACM, 2008.

[27] Bursztein, Elie, Matthieu Martin, and John Mitchell. "Text-based CAPTCHA strengths and weaknesses." In *Proceedings of the 18th ACM conference on Computer and communications security*, pp. 125-138. ACM, 2011.

[28] El Ahmad, Ahmad Salah, Jeff Yan, and Lindsay Marshall. "The robustness of a new CAPTCHA." In *Proceedings of the Third European Workshop on System Security*, pp. 36-41. ACM, 2010.

[29] Fidas, Christos A., Artemios G. Voyiatzis, and Nikolaos M. Avouris. "On the necessity of user-friendly CAPTCHA." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2623-2626. ACM, 2011.

[30] Alsuhibany, Suliman A., and Mohammad Tanvir Parvez. "Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations." In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pp. 126-131. IEEE, 2016.

[31] Koffka, Kurt. "Principles of Gestalt Psychology, International Library of Psychology, Philosophy and Scientific Method." (1935).

[32] Biederman, Irving. "Recognition-by-components: a theory of human image understanding." *Psychological review* 94, no. 2 (1987): 115.

[33] Abd Allatif Eymsh, “word2uni,” GitHub. [Online]. Available: https://github.com/saditechseed/wordofquran/blob/master/libs/word2uni/word2uni.php. [Accessed: 16-Feb-2017].

[34] “GSA Captcha Breaker,” *GSA - Softwareentwicklung und Analytik GmbH*, 2016. [Online] Available: https://captcha-breaker.gsa-online.de/

[35] “tesseract-ocr,” GitHub. [Online]. Available: https://github.com/tesseract-ocr/.

[36] “OCR software for Windows - ABBYY FineReader 14,”OCR software for Windows - ABBYY FineReader 14. [Online]. Available: https://www.abbyy.com/en-me/finereader/.

[37] “Free Online OCR - Convert JPEG, PNG, GIF, BMP, TIFF, PDF, DjVu to Text,” Free Online OCR - Convert JPEG, PNG, GIF, BMP, TIFF, PDF, DjVu to Text. [Online]. Available: http://www.newocr.com/.

[38] Alsuhibany, Suliman A. "A benchmark for designing usable and secure text-based captchas." *International Journal of Network Security & Its Applications* 8, no. 4 (2016): 41-54.

Suliman Alsuhibany, PhD, is an Assistant Professor in the College of Computer, Computer Science department and the head of the department at Qassim University, Saudi Arabia. He received his PhD in Information Security from Newcastle University, UK, and MSc in Computer Security from Newcastle University, UK