Computer Systems Science & Engineering
DOI:10.32604/csse.2022.018929
Article

Tech Science Press

Usability and Security of Arabic Text-based CAPTCHA Using Visual Cryptography

Suliman A. Alsuhibany* **and Meznah Alquraishi**

Department of Computer Science, College of Computer, Qassim University, Buraydah, 51452, Saudi Arabia
*Corresponding Author: Suliman A. Alsuhibany. Email: salsuhibany@qu.edu.sa
Received: 26 March 2021; Accepted: 03 May 2021

Abstract: Recently, with the spread of online services involving websites, attackers have the opportunity to expose these services to malicious actions. To protect these services, A Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is a proposed technique. Since many Arabic countries have developed their online services in Arabic, Arabic text-based CAPTCHA has been introduced to improve the usability for their users. Moreover, there exist a visual cryptography (VC) technique which can be exploited in order to enhance the security of text-based CAPTCHA by encrypting a CAPTCHA image into two shares and decrypting it by asking the user to stack them on each other. However, as yet, the implementation of this technique with regard to Arabic text-based CAPTCHA has not been carried out. Therefore, this paper aims to implement an Arabic printed and handwritten text-based CAPTCHA scheme based on the VC technique. To evaluate this scheme, experimental studies are conducted, and the results show that the implemented scheme offers a reasonable security and usability levels with text-based CAPTCHA itself.

Keywords: Visual cryptography; Arabic text-based CAPTCHA; usability; security; printed and handwritten Arabic script

1 Introduction

With the expansion of the Internet in recent years, different online services have become available such as e-government and e-health to facilitate the use of these services at any time. These services may contain sensitive information that could be exploited by malicious users with the use of automated programs. Consequently, these services need to be secured against such attacks [1,2].

Different techniques have been developed to protect against a variety of attacks in order to keep e-services secure. A technique which is commonly used nowadays is CAPTCHA. CAPTCHA is an abbreviation for a security technique entitled "Completely Automated Public Turing Test to Tell Computers and Humans Apart". This technique protects website-based e-services from being attacked by malicious users [1].

Different types of CAPTCHA have been developed to support various fields. These types tend to be text-based, image-based or audio-based. Text-based CAPTCHA is used most widely due to its many advantages [1].

During the design of any type of CAPTCHA, two aspects should be considered: usability and security. In this context, usability is an aspect that enables the easy solving of a challenge on the part of humans, while the security aspect protects the website from attackers. Balancing usability and security in the design of any CAPTCHA may be a challenge [3].

Many services nowadays use Latin text-based CAPTCHAs. However, some alternatives exist with regard to different languages that may increase the usability aspect for their users. Since most Arabic countries use Arabic script in their transactions, involving Arabic text-based CAPTCHA may be better for their users as discussed in Fidas [4]. Furthermore, not only do Arabic countries use Arabic script, but so too do countries that use other languages such as Persian. In addition, using Arabic in CAPTCHAs designs may increase the resistance to attacks as Arabic script contains some unique characteristics such as the character shape [5,6].

Moreover, the designing of CAPTCHA is undertaken in different ways in order to increase the security aspect without affecting the usability aspect. For example, applying VC techniques can increase the level of security against such attacks [7,8]. In particular, VC is a cryptographic technique discovered by Naor and Shamir in 1995 [9]. It is used to encrypt images so that they can be shared securely. The encryption process is implemented by dividing an image into different shares and which can be decrypted easily by stacking them on each other using a human visual system, without any need for cryptographic computation [9]. Therefore, applying CAPTCHA using the VC technique may increase security, as it is an interactive approach that involves stacking shares on each other to solve the CAPTCHA challenge [8].

This paper extends previous work in Alsuhibany [10] by designing an Arabic text-based CAPTCHA scheme for both printed and handwritten texts using the VC technique as a novel scheme. In addition, the designed scheme is implemented and evaluated to measure its efficiency in terms of its resistance to attack, and the level of ease with which it can be solved by users. The results show that our Arabic text-based CAPTCHA scheme using the VC tool has a practical level of security and usability.

In particular, in terms of usability, our scheme that involved printed and handwritten texts that have meaning was not difficult for users to use. However, in terms of handwritten text from different writers that had no meaning, users had trouble identifying the text. With regard to security, our scheme has two security layers. The first layer involves interaction in that the shares need to be stacked on each other. Therefore, all text types in this layer (i.e., printed and handwritten) have the highest level of security. The second layer involves applying the features of Arabic script, with all text types in our scheme generally having a reasonable level of security.

To the best of our knowledge, no study has yet demonstrated the ability of attackers to break the first layer. However, in our paper we assume this to be possible (i.e., the attacker being able to stack the shares automatically) by conducting an experimental study in order to evaluate the robustness of the second layer. The results show that handwritten texts composed by different writers that have no meaning had the highest security level compared to printed and handwritten texts that have meaning.

Thus, our results show the effectiveness of applying the proposed scheme in terms of Arabic websites, with some improvements that could be applied in future. One of these improvements is developing a handwritten text generator that generates meaningful words which might be of interest in terms of improving both the security and the usability aspects.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 explains our VC tool for use with printed and handwritten Arabic texts. Section 4 describes the methodology. Section 5 presents the experimental studies for evaluating the usability and the security aspects of the proposed scheme. Section 6 shows the results which are then discussed in Section 7. Section 8 offers a conclusion.

2 Related Works

To the best our knowledge, designing an Arabic text-based CAPTCHA scheme for both printed and handwritten texts based on the VC technique has not yet been done. Thus, this section focuses on both printed and handwritten Arabic text-based CAPTCHAs in terms of the security and usability aspects. In addition, it considers several interactive CAPTCHA studies, as our scheme requires interaction in order to stack the two shares.

2.1 Arabic Text-based CAPTCHA

This section discusses studies carried out with regard to printed Arabic text-based CAPTCHAs and handwritten Arabic text-based CAPTCHAs.

2.1.1 Printed Arabic Text-based CAPTCHA

There are various studies that have focused on the use of printed Arabic CAPTCHA and the evaluation of their robustness and usability. Due to the challenges associated with the use of Arabic script in order to make it resistant to recognition, Zheng, Hassin and Tang [5] introduced a new algorithm that can be used to segment machine-printed Arabic characters, as there is a problem that any character shape depends on its location in the word. This algorithm can be exploited to evaluate the robustness of printed Arabic text-based CAPTCHAs.

In 2006, Shahreza [6] developed a new Arabic/Persian text-based CAPTCHA. The development of this CAPTCHA involved exploiting the characteristics found in the Persian/Arabic language such as dots in characters, the shape of characters based on their location, and so on that increase security. Since CAPTCHA technology is used for different purposes, Shahreza [11] used printed CAPTCHAs for verifying spam SMS. In addition, as the development of approaches and methods for CAPTCHAs is still ongoing in an effort to resist attacks, Shahreza [12] proposed a CAPTCHA approach called multilingual CAPTCHA that supports different languages, Arabic being one of these languages. In 2013 Khan et al. [13] developed a new Arabic text-based CAPTCHA scheme that exploits the difficulty that Arabic optical character recognition systems (OCRs) face in reading Arabic text such as adding some background noise and dots. The results of this study were encouraging.

In addition, an approach entitled Arabic reCAPTCHA was discussed in some studies [14–18] and was developed for enhancing the digitization of Arabic manuscripts. In particular, Bakry et al. [14] proposed the use of a reCAPTCHA (AreCAPTCHA) system that can be used to digitize Arabic text through native Arabic speakers. In addition, Akila et al. [15] proposed a new system called "Kalema" that is used for the digitization of scanned Arabic documents which can be converted to an audio format. The Arabic reCAPTCHA service was designed in such a way as to have an architecture based on the use of the Cloud by the authors in Abubaker [16] and [17]. In addition, more features and enhancements were introduced in Abubaker [18] to extend studies [16] and [17]. In 2017, Alsuhibany et al. [19] evaluated the robustness of the Arabic text-based CAPTCHA. The authors found that some Arabic CAPTCHAs could be broken, and provided a set of recommendations that CAPTCHA designers should follow in terms of the design of robust Arabic text-based CAPTCHAs.

In considering works that have studied the usability of printed Arabic text-based CAPTCHAs, based on our knowledge, no studies have attempted to evaluate the usability of printed Arabic text-based CAPTCHAs.

2.1.2 Handwritten Arabic Text-based CAPTCHA

The security of handwritten text-based CAPTCHAs might be higher than that of printed text-based CAPTCHAs due to the differences that exist in terms of writing patterns between humans. In addition, handwritten CAPTCHAs provide more challenges compared to printed CAPTCHAs [20].

A study of Arabic handwritten texts by Alsuhibany and Parvez in [21] used OCR operations such as segmentation and baseline detection for distortion to generate secure handwritten Arabic CAPTCHAs.

To the best of our knowledge, no studies have evaluated the usability of handwritten Arabic text-based CAPTCHAs.

2.2 Interactive CAPTCHA

Interactive CAPTCHA is a type of CAPTCHA that requires humans to interact with the CAPTCHA by clicking or dragging and dropping any object in order to solve an offered challenge. This interaction might increase the security level, as this interaction can be relatively easy for humans but difficult for machines [22].

In 2010 and 2011, the interaction described in Lang [7] and [8] involved moving two shares, and then stacking them on each other to obtain the actual CAPTCHA. In addition, Li and Wang [23] proposed a new interactive CAPTCHA by dragging and dropping four different segments of an image and putting them in the correct place in order to form the original image. Furthermore, the interaction applied in Roshanbin [22] involves a mouse action for selecting the target character from the left-hand image and its corresponding match in the right-hand image.

In 2020, Parvez and Alsuhibany [24] proposed a new handwritten Arabic CAPTCHA scheme that involves interaction. The process involves clicking on the joining points between characters in the CAPTCHA image, either with the use of a mouse or by touching the screen.

Thus, studies in Lang [7,8] are only two studies that have designed text-based CAPTCHA based on visual cryptography. However, both studies involved using only printed English text-based CAPTCHAs. Based on the aforementioned studies, there is still no research that has applied the VC technique in the case of Arabic text-based CAPTCHA that we applied and evaluated in our paper.

When comparing our study with [7] and [8] studies, we detailed the contributions and results of each study to demonstrate our contribution clearly as shown in Tab. 1.

3 Visual Cryptography for Arabic Text-Based CAPTCHA

This section details the Arabic VC tool that was developed in Alsuhibany [10]. Specifically, it firstly presents an overview of Arabic text and Arabic text-based CAPTCHA. Then, it presents how the Arabic VC tool developed to support printed and handwritten Arabic text-based CAPTCHAs.

3.1 Arabic Text and Arabic Text-based CAPTCHA

Arabic text has several differences compared to English text. For example, Arabic text is written from right to left, whereas English is written from left to right. The second difference is that the letters in the Arabic language are connected in both handwritten and printed texts as shown in Fig. 1.

Recently, different types of CAPTCHAs have been discovered that are different based on the method used to design particular CAPTCHAs and the language used for such CAPTCHAs. The reason behind this diversity may be a desire to increase both the security and the usability aspects [13]. One of these types is Arabic text-based CAPTCHAs which we investigate in this paper.

Arabic text-based CAPTCHAs have attracted the attention of many researchers for a number of reasons. Firstly, there are more than twenty Arabic countries which reflect the need to develop CAPTCHAs based on Arabic script. Secondly, some of the online services in these countries will be designed for native Arabic speakers such as those related to government, commerce and so on. Thirdly, the nature of Arabic script makes recognition by automated code difficult; as such script contains a number of special characteristics which might increase security while maintaining the ease of use with regard to users. Furthermore,

Arabic text-based CAPTCHAs facilitate online services for Arabic users, especially if recognizing English letters leads to a usability issue [13].

Table 1: The comparison between our study and previous related works

The Study	The contribution	The results
[7]	They introduce visual encryption to encrypt English printed text-based CAPTCHA. They compared between the performance of users and CAPTCHA's breakers based on how much time is taken to solve their CAPTCHA.	The results showed that humans were able to detect images easily within 16–33 seconds, and deciphering images is almost 100%. Regarding for CAPTCHA breakers, it slowed significantly compared to users
[8]	A new scheme for English printed text-based that use VC technique to encrypt CAPTCHA image in perfectly secure way with some distortion	They conclude that the animation in their proposed CAPTCHA improves the security level against attacks, as the animation seems very difficult for software breaker. However, the animation is easy for humans, as humans are attuned to perceiving motion
Our study	A new scheme for Arabic text-based CAPTCHA using the VC technique as an extension for the tool developed in Alsuhibany [10]. Our scheme was developed for both printed and handwritten Arabic text-based CAPTCHA. Also, we evaluated its usability and security from different aspects and based on some criteria as detailed below. This evaluation to compare between the different text types involved and its effectiveness to be applied in Arabic services	The results showed that the implemented scheme offers a reasonable security and usability levels with text-based CAPTCHA itself that can be improved more to be applied in Arabic services

(a) (b)

Figure 1: (a) Example of Arabic handwritten text, (b) Example of Arabic printed text [19]

Since a number of different Arabic CAPTCHA approaches have been designed and developed, this paper focuses on employing the VC technique with Arabic text-based CAPTCHA.

3.2 The Arabic Visual Cryptography Tool

As was mentioned previously, the first employment of Arabic in terms of the VC technique was in Alsuhibany [10]. Since this study supports the incorporation of Arabic text written with the use of a keyboard (printed) only, this paper improves the situation by also supporting the incorporation of Arabic handwritten text.

The developed Arabic VC tool can be used for different purposes such as anti-phishing, CAPTCHA, watermarking, online payment systems and so on, in support of Arabic electronic services. However, in this paper we focused on applying the developed tool in an Arabic text-based CAPTCHA especially, in order to evaluate its usability and security when applied to Arabic services.

Although there are different schemes for the use of the VC technique to encrypt and decrypt an image, the tool developed in Alsuhibany [10] used a 2-out-of-2 scheme. The reason behind this is its transparency in terms of describing the idea for any person who has no technical background [10]. With regard to the technical aspect of this paper, the development of the algorithm for printed and handwritten Arabic texts based on the VC 2-out-of-2 scheme is accomplished using the Javascript programming language. This algorithm is based on some of functions that may differ slightly in printed text and in handwritten text, as we will see in the next subsections 3.2.1 and 3.2.2 Besides, Google script is utilized to enable us to publish the idea online.

3.2.1 Printed Text

Printed text is text that is generated by using the keyboard on the computer when writing any word or sentence in any language (a sample is shown in Fig. 1b). In the case of printed Arabic text, five functions have been developed for the encryption and the decryption of text using the VC 2-out-of-2 scheme. These functions are: the set-up function, the creation of the text image function, drawing the image function, the encryption and the decryption of the image functions. More details with regard to these functions can be found in Alsuhibany [10].

3.2.2 Handwritten Text

Handwritten text is text that is generated by using the hand to write on a paper, a touch screen or by moving the mouse cursor on the computer to write any word or sentence in any language by hand (a sample is shown in Fig. 1a). The use of handwritten text in a text-based CAPTCHA is important, as it seems easy for humans to recognize such text, but difficult for automated programs [22]. In the case of handwritten Arabic texts, five functions are implemented. These five functions are: the preload, the set-up, the drawing, the encryption and the decryption of the image. The only difference in handwritten Arabic text compared to printed Arabic text is that the former is created and saved as image in advance. Thus, in the creation of the text image function, there will be only a call for the encryption function without any creation of the image. Consequently, prior to the setting up function, there will be a preload function that loads the image with the handwritten text using the loadImage() function.

4 Methodology

The methodology applied in this research can be divided into two main steps: the generation and evaluation of the Arabic text-based CAPTCHA. In particular, the generation mechanism for Arabic text-based CAPTCHA is detailed in subsection 4.1. Furthermore, the steps involved in the evaluation of the generated scheme are discussed in subsection 4.2.

4.1 Generating Arabic Text-based CAPTCHA

The tool developed in Alsuhibany [10] has been improved in order to generate both printed and handwritten Arabic text-based CAPTCHAs. That is, the printed texts are generated based on the creatImage() function included in the tool with the determination of all font types, size and position. On the other hand, the handwritten text was generated by scanning several pieces of text written by single and different writers, and processing them using GIMP software [25]. They will then be added in the developed tool to directly generate handwritten Arabic text-based CAPTCHA.

4.2 Evaluating the Proposed Arabic Text-based CAPTCHA

The proposed scheme is evaluated in terms of the usability and the security aspects by conducting experimental studies, based on a number of criteria. With regard to usability, the criteria are time consumption and accuracy. As far as security is concerned, the criterion is the resistance to different forms of attack that are discussed in this section.

In order to measure the usability of the proposed scheme, an experimental study is conducted. The following describes the evaluation steps with regard to usability. (1) Gathering demographic data (i.e., age, gender, experience in using techniques, and level of education. (2) Designing and implementing a website that includes the tool developed in Alsuhibany [10]. (3) Publishing the website publicly in order to invite as many participants as possible. (4) Recording all necessary data needed for the evaluation of the extent of usability using specific functions. For example, the letters entered by the user, the state of the typed letters entered by the user as to whether they are correct or not, taking into account three reasons for consuming time: for stacking the two shares on each other, for typing the text as it appears and the total time taken to stack the shares once the text has been typed and submitted. (5) Analyzing the data obtained.

In evaluating the security aspect, this is achieved by measuring the resistance of the proposed CAPTCHA against attacks such as segmentation and recognition attacks. Finally, we analyze and discuss the results obtained from the evaluation. The methodology is summarized in Fig. 2.

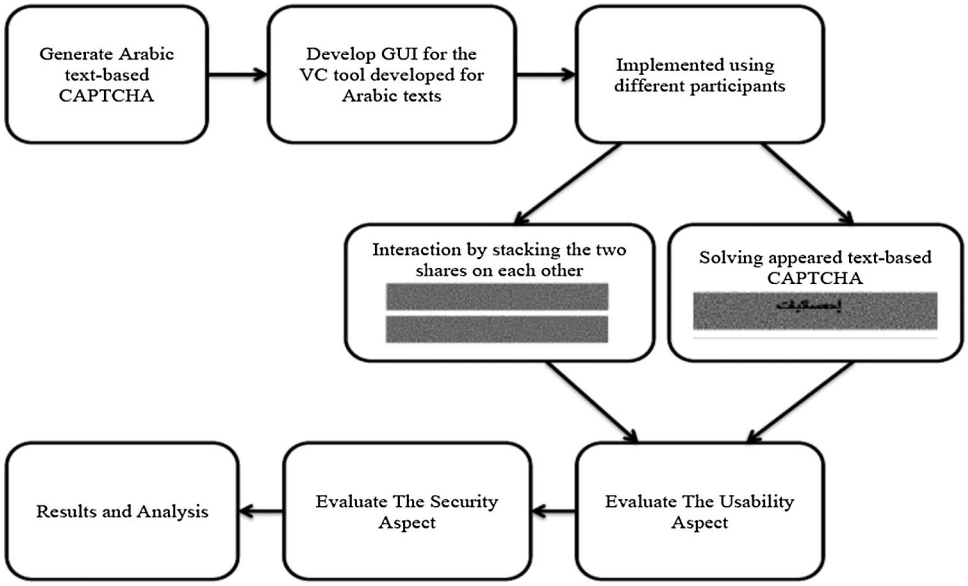

Figure 2: The proposed methodology

5 Experimental Studies

We have implemented two experimental studies; one for evaluating the usability of the proposed system with the aid of a set of participants; the other for the determining the security of the system using the developed algorithms to check their resistance against possible attacks. The following subsections detail these experiments.

5.1 Usability Evaluation

This section explains the setup and procedural steps of the usability experiment.

5.1.1 Experimental Setup

In this experiment, a number of subjects were asked to solve a set of printed and handwritten Arabic text-based CAPTCHAs using the VC technique. Thus, this section describes the experiment in more detail, including the design, the system, the participants, the environment, the text used and the survey.

5.1.1.1 Design

In our experiment, we used a within-subject online design, which means that each participant aims to solve 15 different printed and handwritten Arabic text-based CAPTCHAs. These fifteen text-based CAPTCHAs were basically five samples from each text type, as there are three different text types: meaningful printed text, handwritten text from one writer that has meaning and random handwritten texts from different writers that do not have meaning. These fifteen images were shown randomly. Hence, this type of design aims to ensure that each participant solves all text-based CAPTCHAs to determine which one of the three types of text was the easiest based on the results.

5.1.1.2 System

The graphical user interface (GUI) for our experiment was developed using HTML and JavaScript programming languages. In order to publish it online, we have used the GoogleScript editor.

5.1.1.3 Participants

There were 169 participants. They were recruited to ensure differences in gender, age, educational level and the background technology awareness. However, we excluded 28 participants who did not understand the experiment and solved the given task incorrectly.

5.1.1.4 Environment

The GUI of the experiment was published online. This makes the experiment as realistic as possible. Consequently, the results achieved reflect the usability level of the proposed idea.

5.1.1.5 Text

The Arabic texts used in this research was made up of three types: 100 printed text from a predefined list of dictionary-based words, 200 handwritten text from a single writer with each word from a predefined list of dictionary-based words (i.e., words that have meaning) and 200 handwritten text that was composed of letters from different writers for each word to form a random handwritten meaningless text as achieved in Alsuhibany [26]. This variety of text types allowed a comparison of time consumption as well as a comparison of the correctness of the results between these types.

5.1.1.6 Survey

We developed an online survey using Google Form that was shown to each participant after completion of the experiment to determine their degree of satisfaction. The survey consists of 8 questions related to the VC technique itself, the three types of text used, and our Arabic text-based CAPTCHA using the VC technique. Specifically, these questions were as follows: (1) Do you have a background or knowledge of the concept of the visual cryptography technique? (2) How easy is it for you to interact with the visual cryptography tool when it comes to inserting and placing the two shares on each other to get the actual image? (3) How easy is it to read the word that emerges after inserting the two shares on each other? (4) In the experiment what type of text did you find the easiest to deal with? (5) Which letters did you have difficulty knowing? (6) What is the level of ease of the text-based CAPTCHA used in this experiment, compared to other CAPTCHAs you have encountered in different electronic services? (7) Has the Arabic

language made it easier for you to use text-based CAPTCHA, or would it be easier to use visual cryptography if it was in English? (8) In general, after performing the experiment, which of these aspects was the most difficult for you? (Choose one or more of the following options: understanding the visual cryptography tool; inserting the two shares on each other; reading the text that appeared after stacking the shares; or nothing). Three of these questions were ranked on a 5-point Likert scale as it is a popular scale and gives the participant the chance to offer an accurate opinion. The 5-point Likert scale for two of these 3 questions is: Very Easy, Easy, Neither Easy Nor Difficult, Difficult and Very Difficult. The 5-point Likert scale for the remaining question was: Strongly Agree, Agree, Neither Agree Nor Disagree, Disagree and Strongly Disagree. Finally, we left a space for any participant who had suggestions with regard to the experiment.

5.1.2 Experimental Procedure

In this section, we explain the way that our experiment was run. This refers to the instructions given to the participants, the procedure of the usability experiment, and the collected data.

5.1.2.1 Instructions

At the beginning of the experiment, the participants were informed about the purpose of the experiment. In addition, the participants were informed that they should focus on the given task without being interrupted in any way.

5.1.2.2 The Procedure of the Usability Experiment

We considered quantitative and qualitative metrics to measure the efficiency, the effectiveness of the process and user satisfaction. The participants firstly provided personal information, and then solved all 15 Arabic text-based CAPTCHAs sequentially. Finally, the participants completed the survey. Note that after the participants had solved all the CAPTCHAs that appeared, we recorded for each CAPTCHA the information that is explained in detail in the next section.

5.1.2.3 Collected Data

The personal information of each participant was collected at the beginning of the experiment. This information related to age, gender, the technical background, and the educational level for the participant. Moreover, for each CAPTCHA sample, there were five data items that were recorded in Google Form. These were: 1) the letters typed by users for each shown sample. 2) The accuracy of the typed responses. 3) The time taken to stack the two shares on top of each other until the actual image was obtained. 4) The time taken to type the word. 5) The time taken from starting to move the two shares to the submission of the solution. This was done to measure the efficiency of our proposed CAPTCHA. The time consumption was measured by using a developed function that counts the time taken for a task.

The results recorded in terms of time consumption and the correct recognition of the letters of the word made up the quantitative usability results. However, we recorded all the survey results to collect the qualitative usability results.

5.2 Security Evaluation

For security evaluation, we applied three essential steps: preprocessing, segmentation and recognition. These related to two different procedures that we describe in detail in the following subsections.

5.2.1 Experimental Setup

In this experiment, a number of printed and handwritten Arabic text-based CAPTCHAs were used to evaluate their resistance to attacks. In this section, we describe the security experiment in more detail in terms of the system and text.

5.2.1.1 System

For the preprocessing step, we used GSA software that is designed especially for the goal of breaking CAPTCHAs [27]. Specifically, we used the remove thin objects filter with the radius in the pixel parameter equal to 2 for handwritten texts written by different writers, and the radius in the pixel parameter equal to 3 for printed and handwritten texts written by one writer.

For the segmentation step, we used Matlab software with the GUI developed in Parvez [24] that segments all types of Arabic text images.

For the recognition step, we used two different systems; one is used for first procedure and the other for the second procedure. In particular, the first one was the Google OCR service [28], in that it is a sophisticated engine that can be used by anyone on the Internet to break CAPTCHAs. We used this engine directly on all our images of different types of texts, before and after applying the preprocessing step, in order to measure the resistance of our scheme against Google OCR [28]. The second system involved using the machine-learning principle with different algorithms developed in the Python programming language to recognize characters after the preprocessing and the segmentation steps. These algorithms are: Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbors and Random Forest classifiers. The application of these different algorithms was done in order to compare them, and determine which one has the highest ability in terms of recognition. In addition, we used a deep learning principle developed with Python programming language with the Conventional Neural Network (CNN) algorithm to check their ability when it came to recognition. Choosing CNN was because many researchers have recently shown that CNN is effective in terms of character recognition [29].

5.2.1.2 Texts

The same text types that were used in the usability experiment were used in the security experiment. However, the number of samples was increased to obtain more accurate results in the segmentation and recognition attacks. That is, we used 300 meaningful printed texts, 600 meaningful handwritten texts written by one writer, and 600 meaningless handwritten texts that were composed of letters written by different writers.

5.2.2 Experimental Procedure

In this section, we explain the way that the security experiment was run in terms of the procedure and the collected data.

5.2.2.1 The Procedure of the Security Experiment

We suppose that the attacker could stack the two shares on each other automatically, then we applied two different procedures involving checking the security level of our scheme in different ways, and to determine if there was any modification that should be implemented to improve their resistance against such attacks.

First Procedure: We used Google OCR service to evaluate our scheme. That is, we took a set of images from the following different types of text: 300 meaningful printed, 600 meaningful handwritten written by one writer and 600 meaningless handwritten written by different writers. These samples involved different numbers of characters ranging from 4 to 8. Moreover, these images were fed to Google OCR before and after applying the preprocessing process.

Second Procedure: In the second procedure, we follow three consecutive steps: the preprocessing, segmentation and recognition processes.

1. Preprocessing: this is a process that is used to remove any distortion in the background of the image. This involves converting the image that we have to a binary image. In this paper, we have tried different methods in order to achieve the best results. In particular, we used Otsu thresholding [30] with the Gaussian filter using the Python programming language. Otsu thresholding is used

to automatically determine the best threshold values instead of testing different threshold values manually. The Gaussian filter was applied with different values for the kernel size to derive the values that give us the best results. Moreover, we used GSA software [27] with different filters for our images. Lastly, we selected and used the removing thin objects filter given that it was the most appropriate filter for our images. In addition, the radius in the pixel parameter for the removing thin objects filter was applied with different values to determine the best in terms of giving us the best preprocessing for our images with different text types.

Finally, we implemented the preprocessing procedure for all text types using specialized software entitled GSA CAPTCHA breaker software. This software enabled us to convert all images to black and white and remove all distortion from the background. However, although this software may affect the image negatively during the preprocessing, this can be considered as a positive aspect from the security point of view. For instance, it may remove very small dots written by the writer and this may impact the type of letter in Arabic, especially if two letters have same the shape but differ only in terms of the inclusion of dots such as in the "خ" and "ح" letters. This point may negatively affect the recognition process as will be discussed later in the paper.

2. Segmentation: once the preprocessing step had been applied correctly, the segmentation process was then applied. Segmentation is a process that determines whether or not a character is in the correct location and in the correct order, and segments each character separately. We applied the segmentation process using the segmentation algorithm developed in Parvez [24] for Arabic texts. We classified the segmentation results into four categories as shown in Tab. 2.
3. Recognition: once the segmentation step is undertaken, we run the recognition attack on all samples. The recognition attack is a process that tries to recognize letters and distinguish them from each other. Recently, the Machine Learning (ML) principle and Deep Learning, which is a subset of ML, are considered to be the best methods for recognizing text-based CAPTCHA [31,32].

   For the procedure followed with regard to recognition, we used the Python programming language to implement the selected ML algorithms and the CNN algorithm. We trained the model in each algorithm to recognize the characters as they contained 29 classes, which is equal to the number of Arabic letters in addition to "ئ" as a special character. After we applied the segmentation algorithm, we discarded any characters that are segmented incorrectly, and classified each character segmented correctly into an appropriate class. Therefore, we obtained approximately 4,100 different character images as a dataset for training and testing different models with different algorithms. Moreover, we used 75% of our dataset as training data, while 25% was used as testing data for each model. This distribution in terms of the training and testing sets was empirically selected.

   For implementation purposes, we used an open source Python ML code [33,34] with some improvements based on our particular needs.

5.2.2.2 Collected Data

We collected all the results after every step, images after applying the preprocessing process, and images after applying the segmentation process. We then sorted all the segmented images for utilization in the recognition step. For instance, all images that were segmented correctly with "أ" character were added to one set. This procedure was applied for all the Arabic characters.

6 Results

This section shows the results of evaluating the proposed scheme in terms of usability and security.

Table 2: Segmentation categories

Segmentation Category	Incorrectly Segmented	Partially Segmented	Not Segmented	Completely Segmented
Definition	All characters are segmented incorrectly or one or more characters is segmented to more than one segment	Some of the characters are segmented correctly	None of the characters are segmented	All characters are segmented correctly

6.1 Usability Results

We analyzed the collected data to measure the efficiency, effectiveness, and user satisfaction of the proposed scheme. The details are as shown in the following sections.

6.1.1 The Results in Term of Time Consumption (Efficiency):

The collected times taken were of three types: the time taken to stack the two shares on each other; the time taken to type the word that appeared after stacking the two shares on each other; and the total time taken from stacking the two shares on each other until the submission of the result.

Tab. 3 shows the results with regard to stacking the two shares until the actual image appears. On the other hand, Tab. 4 shows the results in terms of the average time taken by each participant to type the word that appears.

Table 3: The average time taken to move the two shares for all text types

Type of Text	Printed Text	Meaningful Handwritten text from one writer	Meaningless Handwritten t ext from different writers
Average (Seconds)	18.22	14.01	32.45

Table 4: The average time taken to type all the text types

Type of Text	Printed Text	Meaningful Handwritten text from one writer	Meaningless Handwritten text from different writers
Average (Seconds)	7.04	6.54	21.01

We then calculated the average total time for each text type. The comparison between each texts type is shown in Tab. 5.

Table 5: The average of the total time for all text types

Type of Text	Printed Text	Meaningful Handwritten text from one writer	Meaningless Handwritten text from different writers
Average (Seconds)	25.26	20.56	53.46

6.1.2 The Results in Terms of Accuracy (Effectiveness):

Accuracy means the correctness in solving the proposed scheme. We calculated the average of the accuracy for all participants with regard to each text type. The results are as shown in Tab. 6.

Table 6: The average accuracy for all text types

The average accuracy		
Printed text	Meaningful Handwritten text (one writer)	Meaningless Handwritten text (different writers)
83%	91%	17%

6.1.3 The Results of the Survey (Satisfaction):

As we explained previously in Section 5.1.1.6, the participants responded to 8 questions. The results are as shown in Fig. 3.

6.2 Security Results

This section presents the results of evaluating the security aspect of the proposed approach.

6.2.1 First Procedure (Google API):

After we evaluated our samples using the Google API service, we achieved interesting results. In particular, we observed that Google API can recognize all the characters correctly, partially recognize them or cannot recognize them. Moreover, before applying the preprocessing step, we observed that the recognition of the meaningless handwritten texts was ineffective, with a recognition rate of 0.5% as shown in Tab. 7. However, the meaningful printed texts and the handwritten texts written by one writer were somewhat recognizable, with a recognition rate of 9% for the printed texts and 10.33% for the handwritten texts as shown in Tab. 7. Furthermore, in terms of recognition of characters after applying the preprocessing step, we observed that the recognition rate increased significantly (64%) for the printed texts and 32.5% for the meaningful handwritten texts written by one writer as shown in Tab. 8. However, the recognition rate of the meaningless handwritten texts that were written by different writers was still poor (2.7%) as shown in Tab. 8.

6.2.2 Second Procedure:

Since we have applied three sequential steps, this section discusses the results of each step. However, the results of the preprocessing were as an initial step before the segmentation and the recognition steps in order to remove the distortion in the background of the image.

6.2.2.1 Segmentation

We segmented 1,500 different Arabic printed and handwritten text-based CAPTCHAs, and the results are as follows. Firstly, during the segmentation of each sample, if there was an error in segmentation and the segmentation process was stopped, we excluded this image from our results and classified it as an outlier. This outlier was due the effect of the segmentation position as shown in Fig. 4. Consequently, we excluded 35 samples as outliers in total. All of these outliers were from handwritten samples that were either from one writer (20 samples) or from different writers (15 samples). Interestingly, there none of the printed samples were classified as outliers.

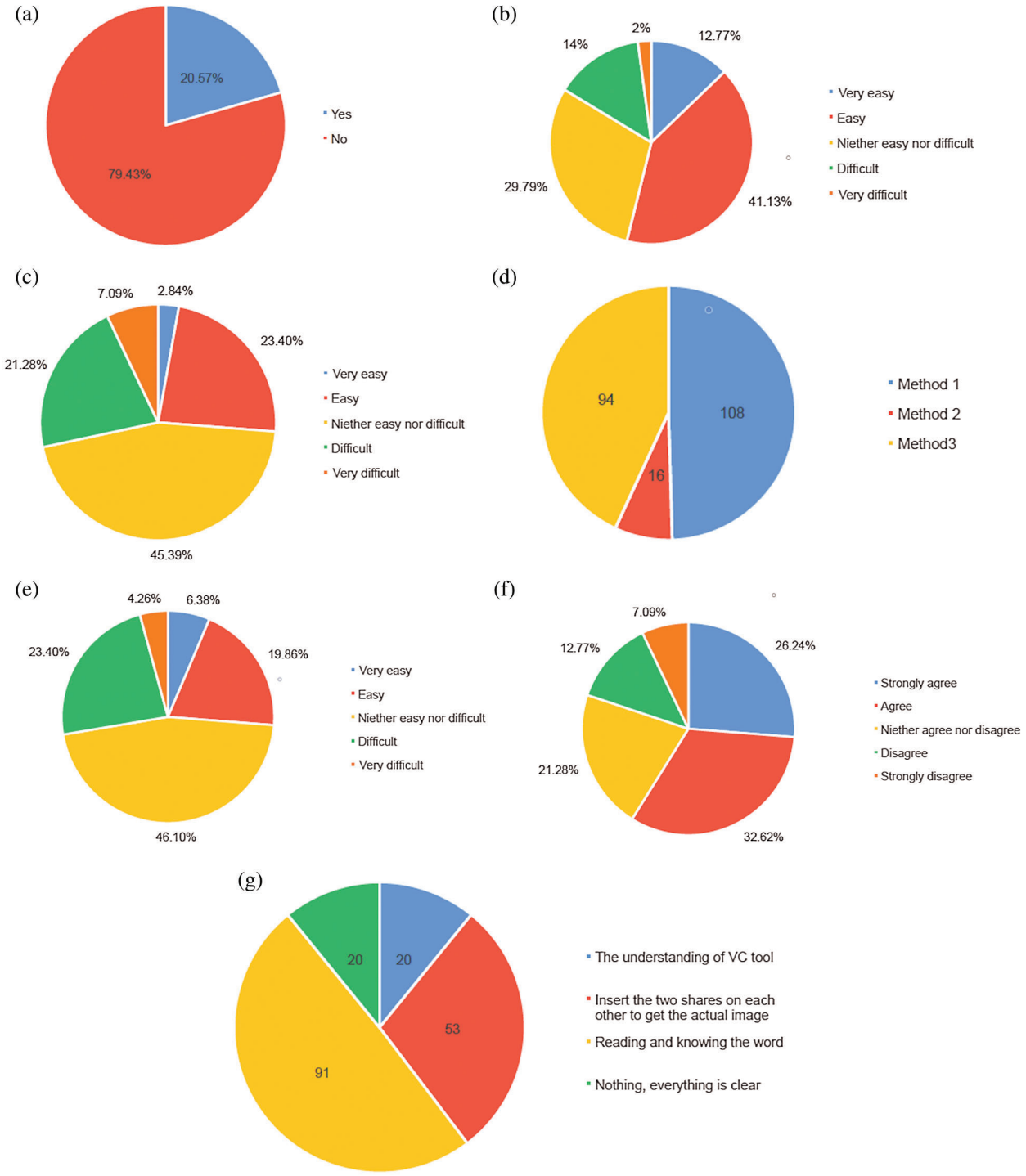

Figure 3: The results of the survey questions: (a) the result of the first question, (b) the result of the second question, (c) the result of the third question, (d) the result of the fourth question, (e) the result of the sixth question, (f) the result of the seventh question and (g) the result of the eighth question

Table 7: The recognition results using Google API before applying preprocessing for Arabic text-based CAPTCHA

Text Type		Number of characters in the image					Percentage of words that are recognized correctly for each text type
		4	5	6	7	8	
Handwritten by different writers	Number of words all of whose characters were recognized correctly	2	1	0	0	0	3 = 0.5%
Handwritten by one writer		7	11	13	12	19	62 = 10.33%
Printed		4	8	6	4	5	27 = 9%

Table 8: The recognition results using Google API after applying preprocessing for Arabic text-based CAPTCHA

Text Type		Number of characters in the image					Percentage of words that are recognized correctly for each text type
		4	5	6	7	8	
Handwritten by different writers	Number of words all of whose characters were recognized correctly	6	5	2	1	2	16 = 2.7%
Handwritten by one writer		47	30	42	38	38	195 = 32.5%
Printed		32	34	41	43	42	192 = 64%

Figure 4: Example of a text-based CAPTCHA that was classified as an outlier

Secondly, if the CAPTCHA image was segmented completely, then it was classified into one of four categories based on how it was segmented. As was explained previously in Tab. 2, these categories are: completely segmented, partially segmented, incorrectly segmented and not segmented.

After classifying the results, we calculated the total and the percentage for each category of the segmentation for each type of text as shown in Tab. 9. This allowed a comparison between text types in terms of resistance against segmentation attack.

6.2.2.2 Recognition

For the ML algorithms, we applied different experiments with different number of training and testing sets, and then selected the best one. However, in the case of the CNN algorithm, we applied different experiments with a different number of training and testing sets, iterations, layers and neurons in each

layer, and then selected the best one. After the implementation of these algorithms, we obtained interesting results as shown in Tab. 10. The accuracy of the character classification determined the ability of each algorithm to recognize and classify each character correctly.

Table 9: The total and percentage of segmenting each text type

Text Type	Calculation	Category of the segmentation			
		Completely Segmented	Partially Segmented	Incorrectly Segmented	Not Segmented
Handwritten from different writers	The total and percent for each type of segmentation	18 = 3%	552 = 92%	15 = 2.5%	0 = 0%
Handwritten from one writer		20 = 3.33%	510 = 85%	49 = 8.17%	1 = 0.17%
Printed		15 = 5%	250 = 83.33%	33 = 11%	2 = 0.67%

Table 10: Results of the classification accuracy of each algorithm

Algorithm	SVM	Naïve Bayes	K-Nearest Neighbors	Random Forest	CNN
Accuracy	47%	0.06%	51%	50%	56%

7 Discussion

This section discusses the results in terms of the usability and security experiments.

7.1 Usability

In general, we observed that the meaningless handwritten text from different writers had the highest time consumption with a big difference compared to the other two types. However, the three consumption times that we calculated for the other text types (i.e., printed and meaningful handwritten text) were very close to each other in terms of consumption time, and better than the meaningless handwritten text from different writers. This may demonstrate that the texts that have meaning were the easiest in general, whether they were printed or handwritten, compared to meaningless handwritten words.

In terms of accuracy, we observed that printed text and handwritten text from one writer were close to each other (i.e., 83% and 91%) compared to meaningless handwritten texts written by different writers which had a low accuracy rate (i.e., 17%). This indicates that handwritten text from one writer is easier to recognize correctly than meaningless handwritten texts written by different writers.

Based on the results of the consumption times shown in Tabs. 3, 4, 5 and the results in terms of accuracy shown in Tab. 6, it is clear that printed text and handwritten text from one writer could make our scheme more usable. With regard to meaningless handwritten texts from different writers, the aim will be to improve the results in our future works.

Based on the results of the survey, we observed that most of the participants did not have any background in VC technology. Thus, we perhaps need to introduce this technology as a game that users can enjoy. Although most of participants did not have a background in VC, most of them were able to interact successfully with the proposed scheme. This may reflect a good usability level in terms of the scheme.

Moreover, in considering which text type was easiest for the participants to use, we observed that more than half of the participants (i.e., 108) stated that the printed text was easy to use. In addition, 94 said that the handwritten texts from one writer were easy to use. This is may be because both of these types of text have meaning and were not made up of meaningless words.

In terms of the level of ease of use of our scheme compared to other schemes, we observed that almost half of the participants (46.1%) said that our scheme was neither easy nor difficult. This motivates us to improve our scheme in terms of its usability level.

With regard to the evaluation of which part of our scheme was the hardest, we noted that 91 participants said that the reading and knowing the words was the most difficult part. This may be due to the use of meaningless handwritten texts from different writers. However, 53 of the participants said that stacking the two shares on each other was the most difficult part. The reason behind this might be their lack of experience, as they may need more time to adopt themselves to the proposed scheme. In terms of understanding the mechanism of the VC tool, only 20 participants said that understanding it was difficult. This represents about 14% of the participants, which means that most of the participants were able to understand the mechanism though this was the first time they had used this tool.

Finally, from the results with regard to question 5 which was about the letters that the participants had difficulty recognizing, we observed that most faced difficulty in dealing with characters that have the similar shape and only differ in terms of the number of dots or the position of the dots. The characters that have a similar shape and cause such confusion were: {ز، ر، د، ذ}, {ط، ظ}, {ع، غ}, {ص، ض}, {ح، ج، خ}, {ب، ت، ث، ن، ي}, {ف، ق}, {و،}. This result confirms the findings of [19]. In addition, the participants found some difficulty in dealing with some of the characters that do not have the exact similar shape, but due to their position in the word and the effect of the writer's handwriting. For instance, {ف} can be confused with {غ}, and {ح} can be confused with {ع} if it comes in the middle of the word and is handwritten such as {ـفـ ، ـغـ} and {ـعـ ، ـحـ}. In addition, characters that have dots may be confused with the background, especially if the dots are written in a thin font. In general, most of the participants said that handwritten words might cause some confusion between different characters due the impact of the writer's handwriting.

7.2 Security

For the first procedure, before applying the preprocessing, the recognition of the characters using Google API was found to be difficult with regard to recognize all text types (i.e., the recognition rate was 10.33%). This may be due to the effect of the distorted background due to stacking the two shares on each other. Unfortunately, after preprocessing, the recognition of the characters using Google API was high for printed texts and for meaningful handwritten texts. As the Google API used ML principles and we have used only one type of font which happened to be the simplest one, this may allow Google API to learn our printed texts easily. However, in the case of meaningless handwritten texts that were written by different writers, the recognition rate was very low even after applying the preprocessing. This reflects the high level of security provided by this type of text.

In general, recognition using Google API is still a challenge, especially for samples prior to preprocessing. Nevertheless, based on our observations, some modifications such as adding different fonts and sizes with regard to the printed texts might improve the security of the proposed scheme, especially for text that has meaning. In terms of meaningful handwritten text written from a single writer, we may need to decrease the size of the word. In addition, altering how some letters are joined up in the writing would mean that part of characters disappears compared with the original as shown in Fig. 5 to make recognition more difficult as understanding how letters are joined up is difficult for the recognizer.

Figure 5: Handwritten text with how the Meem "م" character is part of the words

For the second procedure, based on the results achieved with regard to segmentation, we noticed that every text type was partially segmented in most of the CAPTCHA images. However, we can state that the printed texts were more secure than the two other text types (i.e., handwritten texts), as the former has the highest incorrect segmentation rate (11%) for all characters in the image. In addition, we noticed that all text types are very close to each other in terms of considering the number of Arabic text-based CAPTCHAs that were completely segmented for all characters in the image. This may be due to the effect of the VC technique used, and the application of the preprocessing step on the images. Another reason could be because of the written nature of the handwritten texts and the font type used for the printed texts. However, in the case of the images that were not segmented as a whole, since the number of samples in each text type were very few (i.e., 3 samples), this indicates that our scheme can prevent segmentation attacks.

Finally, in general, in terms of all the segmentation results, we noticed that our scheme is reasonably secure against segmentation attacks as only a very small number of CAPTCHA images were segmented completely (only 53 CAPTCHA samples, i.e., 3.5%). However, our scheme may need improvement in order to increase its resistance to segmentation attacks.

In terms of recognition results, after conducting several experiments, we found that the results of using different algorithms for character classification are close to each other with the exception of the Naive Bayes algorithm that was very small (i.e., 0.06%). However, we observed that the CNN deep learning algorithm had the highest accuracy rate with 56%. This means that the CNN algorithm has a reasonable capacity for recognition but still not very high. This may be due to there being some confusion between some of the characters that have the same shapes and differ only in the number or the position of dots.

Finally, after we obtained these results with regard to character classification and the prediction of the characters, we noted that the accuracy percentage in terms of character classification is not small. This percentage may be because the text-based CAPTCHAs used in our scheme were being completely processed. Thus, our scheme needs some improvements in terms of its ability to resist recognition attacks. For instance, we could apply some rotations, distortions and different fonts for the characters.

It is remarkable that our scheme has the challenge of the interaction by stacking two shares on each other. This gives our scheme a greater degree of security against any attack, as this has not yet been accomplished automatically. Throughout however, we assumed that the attacker would be able to stack the two shares automatically.

8 Conclusion and Future Works

Although there are many Latin text-based CAPTCHAs currently in use, Arabic text-based CAPTCHA has become one of the most important types due to the widespread use of Arabic online services. This paper thus investigated the usability and the security of a new scheme for Arabic text-based CAPTCHA using the VC tool. The proposed scheme has two layers of security: interaction by stacking two shares of a CAPTCHA image to acquire the original CAPTCHA, and the nature of the proposed scheme involving distortion using the VC tool. Experimental studies were conducted to evaluate the usability and security levels of the

proposed scheme. The results show that the proposed scheme has a practical level of usability and security against such attacks.

In terms of future work, our proposed scheme may need some improvements in order to increase its usability and security levels. For example, using filters may improve the movement of the shares. In addition, developing a generator to produce meaningless handwritten texts from different writers may enhance its usability. Moreover, applying different features to protect against different attacks such as segmentation and recognition attacks, might improve the security level.

Acknowledgement: Authors would like to thank Qassim University for supporting this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Von Ahn, M. Blum and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, no. 2, pp. 56–60, 2004.

[2] M. Moradi and M. Keyvanpour, "CAPTCHA and its alternatives: A review," *Security and Communication Networks*, vol. 8, no. 12, pp. 2135–2156, 2015.

[3] C. Obimbo, A. Halligan and P. De Freitas, "CaptchAll: An improvement on the modern text-based CAPTCHA," *Procedia Computer Science*, vol. 20, pp. 496–501, 2013.

[4] C. A. Fidas, N. M. Avouris and A. G. Voyiatzis, "On the necessity of user-friendly CAPTCHA," in *Conference on Human Factors in Computing Systems - Proceedings*, Vancouver, BC, Canada, pp. 2623–2626, 2011.

[5] L. Zheng, A. H. Hassin and X. Tang, "A new algorithm for machine printed Arabic character segmentation," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1723–1729, 2004.

[6] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "Persian/Arabic baffletext CAPTCHA," *Journal of Universal Computer Science*, vol. 12, no. 12, pp. 1783–1796, 2006.

[7] S. R. Lang and N. Williams, "Impeding CAPTCHA breakers with visual decryption," in *Proc. of the Eighth Australasian Conference on Information Security*, Australian Computer Society, Australia, vol. 105, pp. 39–46, 2010.

[8] A. E. Ali, N. F. Hassan and M. E. E. D. Abdulmunim, "Generate animated CAPTCHA based on visual cryptography concept," *Engineering and Technology Journal*, vol. 29, no. 16, pp. 3405–3416, 2011.

[9] M. Naor and A. Shamir, "Visual cryptography BT-Advances in Cryptology — EUROCRYPT'94," Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, pp. 1–12, 1995.

[10] S. A. Alsuhibany, "Developing a visual cryptography tool for Arabic text," *IEEE Access*, vol. 7, pp. 76573–76579, 2019.

[11] M. S. Shahreza, "Verifying spam SMS By Arabic CAPTCHA," in *2006 2nd Int. Conf. on Information & Communication Technologies*, Damascus, Syria, vol. 1, pp. 78–83, 2006.

[12] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "Multilingual CAPTCHA," in *ICCC 2007 - 5th IEEE Int. Conf. on Computational Cybernetics*, Gammarth, Tunisia, pp. 135–139, 2007.

[13] B. Khan, K. Alghathbar, M. K. Khan, A. M. AlKelabi and A. Alajaji, "Cyber security using arabic captcha scheme," *International Arab Journal of Information Technology*, vol. 10, no. 1, pp. 76–84, 2013.

[14] M. Bakry, M. Khamis and S. Abdennadher, "AreCAPTCHA: Outsourcing Arabic Text Digitization to Native Speakers," in *2014 11th IAPR International Workshop on Document Analysis Systems, IEEE*, Tours, France, pp. 304–308, 2014.

[15] G. Akila, M. El-Menisy, O. Khaled, N. Sharaf, N. Tarhony *et al.,* "Kalema: Digitizing arabic content for accessibility purposes using crowdsourcing," in *Int Conf. on Intelligent Text Processing and Computational Linguistics*, Cairo, Egypt, pp. 655–662, 2015.

[16] H. Abubaker, K. Salah, H. Al-Muhairi and A. Bentiba, "Cloud-based Arabic reCAPTCHA service: Design and architecture," in *2015 IEEE/ACS 12th Int. Conf. of Computer Systems and Applications (AICCSA)*, Marrakech, Morocco, pp. 1–6, 2015.

[17] H. Abubaker, K. Salah, H. Al-Muhairi and A. Bentiba, "Architectural design of a cloud-based reCAPTCHA service," in *2016 12th Int. Conf. on Innovations in Information Technology (IIT), Innovations in Information Technology (IIT), 2016 12th Int. Conf. on. IEEE*, Al Ain, Abu Dhabi, United Arab Emirates, pp. 1–6, 2016.

[18] H. Abubaker, K. Salah, H. Al-Muhairi and A. Bentiba, "Arabic reCAPTCHA service for enhancing digitization of Arabic manuscripts," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3391–3408, 2017.

[19] S. A. Alsuhibany, M. T. Parvez, N. Alrobah, F. Almohaimeed and S. Alduayji, "Evaluating robustness of Arabic CAPTCHAs," in *2017 2nd Int. Conf. on Anti-Cyber Crimes (ICACC)*, Abha, Saudi Arabia, pp. 81–86, 2017.

[20] A. Rusu and V. Govindaraju, "Handwritten CAPTCHA: using the difference in the abilities of humans and machines in reading handwritten words," in *Ninth International Workshop on Frontiers in Handwriting Recognition IEEE*, Kokubunji, Japan, pp. 226–231, 2004.

[21] S. A. Alsuhibany and M. T. Parvez, "Secure Arabic handwritten CAPTCHA generation using OCR operations," in *2016 15th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). IEEE*, Shenzhen, China, pp. 126–131, 2016.

[22] N. Roshanbin, "Interweaving unicode, color, and human interactions to enhance CAPTCHA security," 2014.

[23] X. Li and X. Wang, "Research on interactive CAPTCHA mechanism based on RIA," in *2011 Int. Conf. on Multimedia Technology IEEE*, Hangzhou, China, pp. 679–682, 2011.

[24] M. T. Parvez and S. A. Alsuhibany, "Segmentation-validation based handwritten Arabic CAPTCHA generation," *Computers & Security*, vol. 95, pp. 1–12, 2020.

[25] "GIMP - GNU Image Manipulation Program." https://www.gimp.org/ (accessed Nov. 27, 2020).

[26] S. A. Alsuhibany, F. Almohaimeed and N. Alrobah, "Synthetic Arabic handwritten CAPTCHA," *International Journal of Information and Computer Security*, (in press), 2021.

[27] "Captcha Breaker, Breaks any Captcha - Works with any Software." https://www.gsa-online.de/product/captcha_breaker/ (accessed Oct. 24, 2020).

[28] "Vision AI | Derive Image Insights via ML | Cloud Vision API." https://cloud.google.com/vision (accessed Oct. 24, 2020).

[29] D. Lin, F. Lin, Y. Lv, F. Cai and D. Cao, "Chinese character CAPTCHA recognition and performance estimation via deep neural network," *Neurocomputing*, vol. 288, pp. 11–19, 2018.

[30] "OpenCV: Image Thresholding." https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html (accessed Oct. 24, 2020).

[31] O. Bostik and J. Klecka, "Recognition of CAPTCHA characters by supervised machine learning algorithms," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 208–213, 2018.

[32] M. Yousef, K. F. Hussain and U. S. Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognition*, vol. 108, pp. 107482, 2020.

[33] "Introduction-to-Machine-Learning/Building a Digit Recognizer at master · codeheroku/Introduction-to-Machine-Learning · GitHub." https://github.com/codeheroku/Introduction-to-Machine-Learning/tree/master/Building a Digit Recognizer (accessed Nov. 01, 2020).

[34] "Classify Images Using Convolutional Neural Networks & Python | by randerson112358 | Medium." 2019. https://randerson112358.medium.com/classify-images-using-convolutional-neural-networks-python-a89cecc8c679 (accessed Nov. 01, 2020).