IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 28, 2021, accepted October 5, 2021, date of publication October 13, 2021, date of current version October 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3119571

Interactive Handwritten and Text-Based Handwritten Arabic CAPTCHA Schemes for Mobile Devices: A Comparative Study

SULIMAN A. ALSUHIBANY AND AYSHAH A. ALNOSHAN
Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia
Corresponding author: Suliman A. Alsuhibany (salsuhibany@qu.edu.sa)

The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

ABSTRACT CAPTCHA tests (Completely Automated Public Turing test to tell Computer and Humans Apart) are used by many services and websites. Recently, researchers have proposed interactive handwritten and text-based handwritten Arabic CAPTCHA schemes. The former scheme presents a handwritten CAPTCHA image, then requests users to select the joints between Arabic letters. In the latter scheme, a new generator of Arabic handwritten CAPTCHA images is developed, once the image is generated, the user is asked to type the letters shown in the image. Although both of them have shown promising results, this experimental study compares them in terms of security and usability for mobile device applications. The results demonstrated that the interactive scheme performs better than the text-based handwritten scheme in both usability and security.

INDEX TERMS Information security, authentication, handwriting synthesis, Arabic CAPTCHA, interactive CAPTCHA.

I. INTRODUCTION

Due to their popularity, many electronic services (e-services) have been targeted by denial of service attacks using automated codes and robots. Thus, it has become important to maintain the availability of these e-services and ensure that all users are real human beings. For this reason, CAPTCHA tests (Completely Automated Public Turing test to Tell Computer and Humans Apart) were created in [1] to differentiate between real users and automated codes. In particular, many websites use CAPTCHA tests to prevent and protect their services from malicious users and automated spammers. Several global companies use them as well, such as Google and Microsoft. Moreover, CAPTCHA tests come in many varieties, such as text-based CAPTCHA, audio-based CAPTCHA, image-based CAPTCHA and interactive CAPTCHA.

Most researchers have developed and improved text CAPTCHA schemes based on the Latin script. Although an English CAPTCHA scheme has several benefits, it also has several vulnerabilities and weaknesses, as demonstrated in [6], [16]. A user who does not know English might also find these schemes difficult to use [17]. One solution is to use alternatives to the Latin script, such as the Arabic script [18]. According to Internet World Stats [19], the internet usage in Arabic countries has increased as their populations have grown, with 180 million individuals in 2020 living in Arabic-speaking countries. Additionally, a number of websites already provide services in Arabic. Moreover, it is not only Arabic that uses the Arabic script, but also other languages, including Persian, Malay and Urdu. Thus, utilising the Arabic script allows more users to interact with CAPTCHA schemes [18]. From a security point of view, the recognition of the Arabic script is more difficult, as it supports fonts that differ in the design of the letters' strokes and ligatures. Despite this advantage, Arabic-language CAPTCHA schemes may be the target of various types of attack, such as those from artificial intelligence [16].

Recently, researchers have proposed two novel Arabic CAPTCHA schemes: an interactive handwritten Arabic CAPTCHA [3] and a text-based handwritten Arabic CAPTCHA [2]. The former scheme presents a handwritten CAPTCHA image, then requests users to select the joints between Arabic letters. In the latter scheme, the researchers created a new generator of Arabic handwritten CAPTCHA images that may feature letters written by one or different

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Liu.

writers. Once the image is generated, the user is asked to type the letters shown in the image.

In this study, we regenerated the interactive [3] and text-based Arabic schemes [2] in order to compare their usability and security aspects for mobile device, since the expectation that the efficiency might increase further due to the way people hold mobile device. For usability, we conducted an experimental study on mobile devices to measure two performance parameters: efficiency (i.e. the time to solve a CAPTCHA) and effectiveness (i.e. the correctness of typing the characters shown in the text-based scheme, or the correctness of indicating the letter joints in the interactive scheme). We also conducted an experimental security evaluation to measure the resistance of each scheme against select attacks. The results of our comparative study showed that the interactive scheme is better suited to mobile device use than the text-based scheme. In addition, the interactive scheme is more resistant against attacks.

This paper is organized as follows: We discuss related works in Section 2. Section 3 gives an overview of the targeted schemes. Section 4 explains our method. We show the results in Section 5 and discuss them in Section 6, and finally, we conclude the paper in Section 7.

II. RELATED WORK

This section discusses general Arabic text-based CAPTCHA schemes, interactive handwritten CAPTCHA schemes and the use of CAPTCHA on mobile devices.

A. TEXT-BASED HANDWRITTEN ARABIC CAPTCHA

Alsuhibany and Parvez [7] proposed a method to secure handwritten Arabic CAPTCHA tests based on the KFUPM Handwritten Arabic TexT (KAHTT) database [10]. The first step is to carry out the CAPTCHA tests as PAW (Part of Arabic Word), then extract the PAW main body and segment it via a PAW segmentation algorithm. After that, the segmented characters in the PAW scheme are displaced from their position. Finally, random colouring and noise are added to the image. Another method to generate an Arabic CAPTCHA image is to use multilevel difficulty by applying vertical or horizontal PAW image displacement, as well as random rotation. This method's robustness was tested by using segmentation and recognition techniques. Additionally, the researchers evaluated usability for two aspects: response time and accuracy. The results of both tests gave a good indication of applying handwritten Arabic language compared with other works. Although of this indication, this method is limited in the sense that a finite number of Arabic words from a pre-collected database were used.

Aldosari and Al-Daraiseh [4] also presented a new advanced CAPTCHA technique to differentiate between humans and bots. This technique utilises handwritten CAPTCHA images with unique features to separate handwritten characters. This CAPTCHA scheme can be combined with different languages besides English (the default). The authors used six different optical character recognition (OCR) readers to test the technique's robustness. This showed a good result in terms of the usability as the percentage of correctly recognised CAPTCHA images was 92%. However, there is a lack of the robustness evaluation in which the OCRs are only used and more sophisticated methods can be used such as an automated segmentation algorithm attack and machine learning approaches.

Alsuhibany *et al.* [2] offered a new generator of handwritten Arabic CAPTCHA images as well based on different writers. In particular, the generator randomly creates a CAPTCHA image by selecting a number of characters to appear in the image. This image may contain letters from one or different writers. It is important to note that one writer means that all selected letters have been written by one writer, whereas different writers mean that the selected letters have all been written by different writers. The generator also distorts, rotates and flips the Arabic letters. This generator, however, has not been tested on smartphones, as we aim in this paper.

B. INTERACTIVE ARABIC CAPTCHA

In contrast to a text-based approach, Parvez and Alsuhibany [3] developed an interactive handwritten Arabic CAPTCHA scheme. This scheme generates a handwritten Arabic CAPTCHA image, and the user is then requested to select a joint between the Arabic letters that appear in the image. The generation of this CAPTCHA image stems from synthesised Arabic PAWs. This scheme has been evaluated for both usability and security, with the results showing good usability and security levels. This scheme, however, has not been tested on smartphones, as we aim in this paper.

C. CAPTCHA TESTS ON MOBILE DEVICES

Kulkarni and Fadewar [8] created a new CAPTCHA scheme specifically for mobile devices. The proposed pedometric CAPTCHA scheme attends to users' abilities while walking or moving with the mobile device. Meanwhile, Guerar *et al.* [9] proposed a physical CAPTCHA method for mobile devices. This method requires users to move the mobile device at a specific angle, as well as enter a PIN. Moreover, Saxena *et al.* [24] proposed a new CAPTCHA scheme that depends on a cloud data and test storage. Author proposes more than one method of CAPTCHA test, depends on request from users to select a specific country location, specific color, and drag until the end of the test. In addition, Aburada *et al.* [28] proposed a new CAPTCHA suitable for mobile devices and discussed its practicality. Although these studies proposed CAPTCHA schemes for mobile devices, but their formulations do not fit exactly with our approach in terms of handwritten Arabic CAPTCHA. We refer to the next section for a further discussion.

Guerar *et al.* [26] introduced Invisible CAPPCHA approach that uses a trusted sensor embedded in a secure element located on a smartphone. This approach is completely transparent to users in terms of distinguishing between human and computers. Nevertheless, this approach has a low level of

TABLE 1. Limitations of related works.

Study	Limitation
[7]	This study proposed a method that is limited in the sense that a finite number of Arabic words from a pre-collected database were used
[4]	There is a lack of the robustness evaluation in this study in which the OCRs are only used and more sophisticated methods can be used such as an automated segmentation algorithm attack and machine learning approaches
[2]	The proposed generator in this study has not been tested on smartphones, as we aim in this paper
[3]	The proposed scheme in this study has not been tested on smartphones, as we aim in this paper
[8], [9], [24], [28]	Although these studies proposed CAPTCHA schemes for mobile devices, but their formulations do not fit exactly with our approach in terms of handwritten Arabic CAPTCHA.
[26]	This study has a low level of accuracy in the detection of the tap event, and there was no detail for the usability test.
[27]	Although this study showed some schemes' performance better than others, all of them have such usability issues like the ambiguity of some CAPTCHA images. Moreover, the samples size used in the experiment was too small that would reduce the power of the study and increase the margin of error.

accuracy in the detection of the tap event. Also, there was no detail for the usability test.

Jiang *et al.* [27] presented an exploratory study that aims to develop a more holistic view of usability issues in mobile friendly CAPTCHA. In particular, the performance of seven different CAPTCHA schemes was examined. Although some schemes showed performance better than others, all of them have such usability issues like the ambiguity of some CAPTCHA images as participants were zooming in and out to inspect the detail, which may lead to tap on an image by accident since the images in such tests occupied the whole screen. Moreover, the samples size used in the experiment (i.e. 20 participants) is too small that would reduce the power of the study and increase the margin of error, which can render the study meaningless.

Table 1 compares aforementioned studies' limitations.

III. TARGETED SCHEMES: AN OVERVIEW

This section explains the Arabic script, the interactive handwritten Arabic CAPTCHA scheme, and text-based handwritten Arabic CAPTCHA scheme.

جامعة القصيم

(a) a handwritten Arabic text

(b) a printed Arabic text

FIGURE 1. (a) A handwritten Arabic text sample and (b) a printed Arabic text sample.

A. ARABIC SCRIPT

Since the targeted schemes are based on Arabic script, this section explains briefly the characteristics of the Arabic language in terms of writing direction, shapes, and recognition. In particular, the Arabic language has 28 basic letters that can be described with 15 primary strokes, and they only differ in the number or position of letters' dots. Arabic letters are written from right to left, and they are connected during writing, both in printed and handwritten texts. Table 2 shows the Arabic letters and their contextual forms.

Generally speaking, Arabic letters are context-sensitive—a single letter can be written in up to four different contextual shapes depending on its position in a word. For instance, as shown in Table 2, the form of the letter *meem* can be either "ـمـ," "ـم," "م," or "مـ," where it can be a single letter, at the end of a word, between two letters, or at the beginning of a word. Moreover, several Arabic characters have similar shapes, for example, ب ت ث ن, ج ح خ, ص ض, ع غ, ط ظ, د ذ ر ز, و, ف ق. As stated in [5], [20], this similarity makes it difficult for OCR to recognize characters correctly.

In contrast to Latin script, there are various features of Arabic scripts that make the recognition process relatively more difficult. In Arabic writing, the lack of space between characters is one of these features, making the recognition process and the segmentation phase in both printed and handwritten Arabic text harder [21]. When typing Arabic text, there can be an overlapping between characters in terms of space (e.g., "وا" in which "ا" overlaps with "و"). This overlapping feature makes both the recognition and segmentation processes difficult, as demonstrated in [22]. Arabic OCRs are mostly developed based on a few font types. When a text is written in a different font type, it is unrecognizable [23].

Despite extensive research in handwriting recognition over the past several decades, the recognition results for handwritten text are far behind those obtained for printed text. The regularities present in printed text are not available in unconstrained handwriting. Thus, the recognition of handwritten text remains a challenging task. Figure 1 shows samples written in both printed and handwritten texts.

B. THE INTERACTIVE HANDWRITTEN ARABIC CAPTCHA SCHEME

The interactive handwritten Arabic CAPTCHA scheme generates a CAPTCHA image based on synthesised PAWs. This generation depends on four levels of distortion, listed as follows. The Level 0 represents an original image generated

TABLE 2. Arabic letters and their contextual forms.

Transcription	General	Contextual forms			
		Isolated	End	Middle	Beginning
ALEF	ا	ا	ـا	-	-
BAA	ب	ب	ـب	ـبـ	بـ
TAA	ت	ت	ـت	ـتـ	تـ
THA	ث	ث	ـث	ـثـ	ثـ
JEEM	ج	ج	ـج	ـجـ	جـ
HAA	ح	ح	ـح	ـحـ	حـ
KHAA	خ	خ	ـخ	ـخـ	خـ
DAL	د	د	ـد	-	-
THAL	ذ	ذ	ـذ	-	-
RAA	ر	ر	ـر	-	-
ZAIN	ز	ز	ـز	-	-
SEEN	س	س	ـس	ـسـ	سـ
SHEEN	ش	ش	ـش	ـشـ	شـ
SAAD	ص	ص	ـص	ـصـ	صـ
DHAD	ض	ض	ـض	ـضـ	ضـ
TTAA	ط	ط	ـط	ـطـ	طـ
TTHAA	ظ	ظ	ـظ	ـظـ	ظـ
AIN	ع	ع	ـع	ـعـ	عـ
GHAIN	غ	غ	ـغ	ـغـ	غـ
FAA	ف	ف	ـف	ـفـ	فـ
QAAF	ق	ق	ـق	ـقـ	قـ
KAAF	ك	ك	ـك	ـكـ	كـ
LAM	ل	ل	ـل	ـلـ	لـ
MEEM	م	م	ـم	ـمـ	مـ
NOON	ن	ن	ـن	ـنـ	نـ
HHAA	هـ	ه	ـه	ـهـ	هـ
WAW	و	و	ـو	ـو	و
YAA	ي	ي	ـي	ـيـ	يـ
TAAM	-	ـة	-	-	-
WAW with hamazah	-	ؤ	ـؤ	-	-

without any distortion, as shown in Figure 2(a). The Level 1 represents a CAPTCHA image with the enclosed space of each character filled in with random background colour (Figure 2(b)). The Level 2 features constituents (e.g. random colours and dots) placed above and below each letter (Figure 2(c)). The fourth and final level represents the same distortion as level 2, but adds three horizontal lines, one of them dotted, across the full word, as shown in Figure 2(d).

As shown in Figure 2, the interactive CAPTCHA scheme uses synthesized handwritten Arabic words to generate CAPTCHAs. To solve this scheme, users are asked to find the segmentation points in the cursive Arabic words as shown in Figure 3.

This scheme is evaluated in terms of security and usability aspects. In particular, a controlled laboratory experiment was conducted to evaluate the usability of this scheme in two modes: *touching mode* and *clicking mode*. In touching mode, the user was asked to select the joining points in the CAPTCHA image by touching the screen with his/her finger,

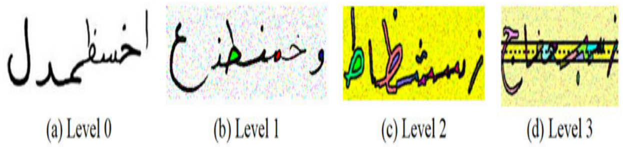


FIGURE 2. Different levels of distortion in the interactive CAPTCHA scheme [3].

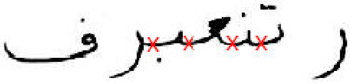


FIGURE 3. Segmentation points are marked by 'x' [3].

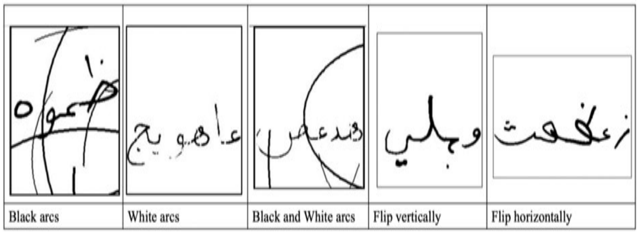


FIGURE 4. Different types of distortion in the text-based CAPTCHA scheme [2].

while in clicking mode, the user was asked to select the joining points in the CAPTCHA image using a mouse. The results of this evaluation showed that the click mode was generally easier than touch mode. Moreover, an automated segmentation algorithm attack and a number of OCR attacks were used to evaluate the security of this scheme. The results showed interesting level of resistance against these attacks, although the automatic segmentation algorithm attack poses a security threat for only 4% to 6% of CAPTCHA samples with distortion levels 0 or 1.

C. THE TEXT-BASED HANDWRITTEN ARABIC CAPTCHA SCHEME

The text-based handwritten Arabic CAPTCHA scheme generates a CAPTCHA image by applying different distortions and rotations, such as horizontal and vertical flips, of some or all Arabic letters. Then, the user is asked to type the displayed letters. Figure 4 shows examples of the different distortion types. For details, refer to the reference [2].

The usability and robustness aspects of this scheme were evaluated in this section. An experimental study was conducted to collect data on user performance in a laboratory environment. In this study, two metrics were measured: the correctness of the solutions entered of a given CAPTCHA by the user and the average time in seconds taken by the user to solve a given CAPTCHA. Furthermore, the security evaluation include different attacks that are usually used to break text-based CAPTCHAs [e.g. 25] were utilized. The results of the evaluation showed a good success rate in terms of both security and usability aspects.

IV. EXPERIMENTAL STUDY

This paper aims to empirically investigate the practicality of interactive handwritten and text-based handwritten Arabic CAPTCHA schemes for mobile devices. Thus, both schemes implemented in order to be adopted for mobile devices. As our methodology focuses on the quantitative performance measures, an experimental study conducted to evaluate the security performance measures and the usability performance measures. For the security performance measures, we analyzed the segmentation and recognition accuracies. For the usability performance measures, we analyzed the efficiency and effectiveness.

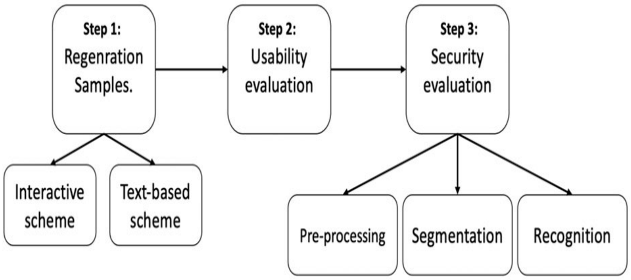


FIGURE 5. Steps of our utilised experimental study.

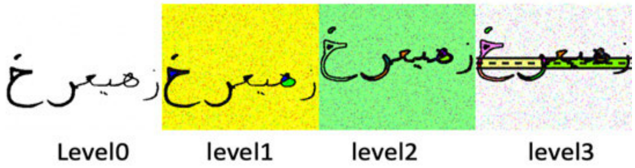


FIGURE 6. Different distortion levels in the interactive scheme samples.

The reason behind choosing the quantitative approach is to clearly determine the more practical scheme for mobile devices. Moreover, it would help to compare not only them together, but also with others. Therefore, the results of both schemes are compared with others as will be shown in Section 6. More details are discussed in the following section.

Our experimental study is divided into three main steps, shown in Figure 5. We discuss these steps in detail in the following sections.

A. SAMPLE GENERATION

The first step of our experiment was to generate new samples from both the interactive and text-based handwritten Arabic CAPTCHA schemes. These samples were used in the second and third steps. For the interactive scheme [3], we generated 2,000 samples, with 500 from each level of distortion. Figure 6 shows examples of the interactive scheme's different distortion levels.

For the text-based handwritten Arabic CAPTCHA scheme [2], we generated 5,000 samples, with 500 for each type of distortion. These distortions were black arcs, white arcs, black and white (B&W) arcs, horizontal flips and vertical flips for one writer and different writers (i.e. 2,500 samples for one writer and 2,500 samples for different writers). Figure 7 shows samples of each arc type and Figure 8 shows examples of horizontal and vertical flips.

B. USABILITY EVALUATION

This section describes the developed application for the experimental study, the design of the experiment, participants and collected data.

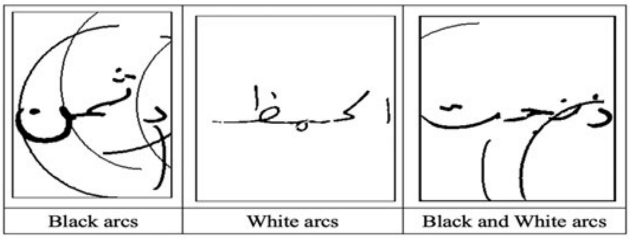


FIGURE 7. Different arc distortions in the text-based scheme.

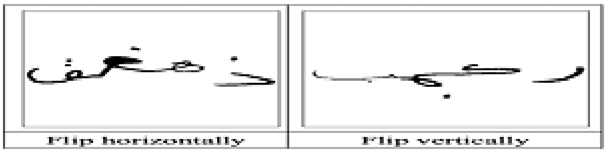


FIGURE 8. Different flip modes in the text-based scheme.

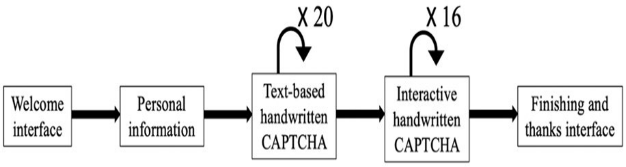

Welcome interface
Personal information
X 20
Text-based handwritten CAPTCHA
X 16
Interactive handwritten CAPTCHA
Finishing and thanks interface

FIGURE 9. Interfaces of the developed application.

1) USABILITY EVALUATION APPLICATION

In the usability study, we first conducted a pilot study; based on the feedback from this study, we conducted the real experiment. In particular, we designed an application for the Android system using Android Studio IDE. This application has several interfaces, as shown in Figure 9, with each interface explained below.

The initial application interface is the welcoming interface contained the research title. After this interface, the user personal information interface appeared, which asked for the users' gender, age, personal phone system and technical background to help us in analysing the data. After users entered this personal information, the third interface asked them to start recognising the text-based CAPTCHA images. There were 20 interfaces containing samples of this scheme. For this, we randomly selected these samples from the different types of distortion. Furthermore, these samples contained different numbers of letters, ranging from 4 to 8. We generated these samples using the same generator as [2] to create meaningless words. Figure 10 shows a sample of this scheme as used in our study.

After completing the text-based CAPTCHA scheme task, the fourth interface prompted the users to start the interactive scheme test. For this, we randomly selected 16 CAPTCHA samples, with 4 images from each level of the interactive CAPTCHA scheme as discussed in Section 3. Figure 11 shows a sample of this scheme as used in our study.

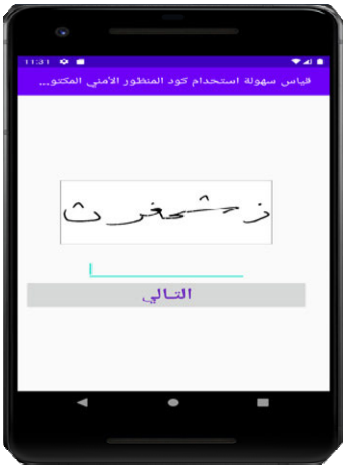


FIGURE 10. A sample text-based Handwritten Arabic CAPTCHA image shown in the developed application's interface.

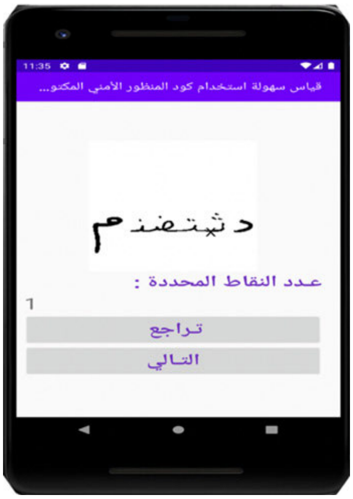


FIGURE 11. A sample interactive CAPTCHA image shown in the interface.

It is important noting that the aforementioned process should be accomplished sequentially.

2) DESIGN OF THE USABILITY EXPERIMENT

Due to Covid-19 restrictions, we could not conduct a controlled usability experiment. Therefore, we conducted the usability evaluation for both schemes in an uncontrolled environment meant to mimic real-world conditions when solving CAPTCHA challenges.

This study's experimental design is within-subjects, which means that all the participants were asked to solve twenty samples for the text-based handwritten CAPTCHA scheme. Then, the participants were asked to solve sixteen samples from each of the four distortion levels developed (as explained above) for the interactive handwritten CAPTCHA, as shown in Figure 9. This ensured that the same number of CAPTCHAs were solved for each scheme, and that there were no confounding factors causing bias in the results.

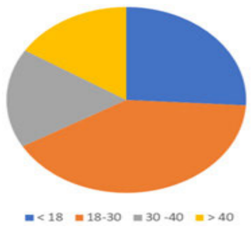


FIGURE 12. Participants' age.

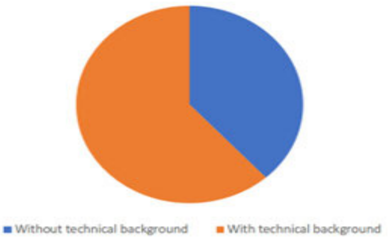


FIGURE 13. Participants' background.

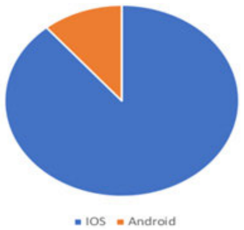


FIGURE 14. Participants' personal mobile platform.

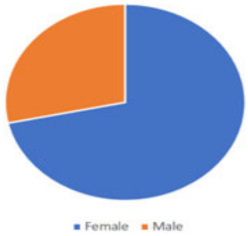


FIGURE 15. Participants' gender.

3) PARTICIPANTS

The number of participants who successfully completed the task amounted to 80 volunteers. Figures 12–15 show the participants' characteristics. In particular, most of the participants' ages (41%) ranged from 18 to 30, have technical background (62%), using IOS operating system (89%), and female (72%).

4) APPARATUS

We developed and implemented an Android application for evaluating the usability aspect as we explained previously. This application then was installed on Android OS smart phone which is: *Samsung Galaxy A10.*

5) COLLECTED DATA

We assessed both schemes' usability by collecting quantitative data to measure the satisfaction and human performance.

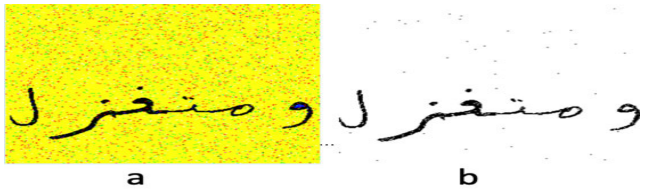


FIGURE 16. An interactive CAPTCHA sample before and after pre-processing.

Accordingly, we recorded two parameters in our system's database:

- The user input (i.e. the typed letters for the text-based scheme and the correctness of indicating the joints of the displayed characters for the interactive scheme).
- The response time (i.e. the time users took to solve the CAPTCHA challenges, in seconds).

C. SECURITY EVALUATION

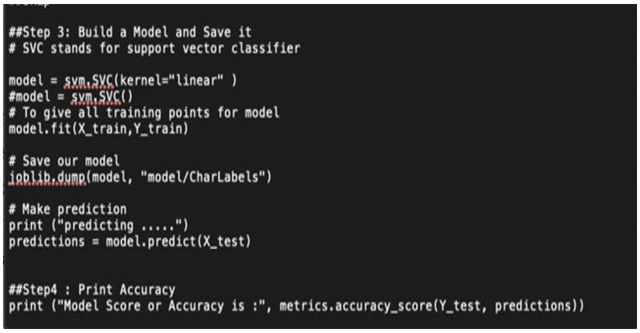
For the security evaluation, there were three processes that we applied to test the robustness of each CAPTCHA: pre-processing, segmentation and recognition. We applied pre-processing on both schemes. We also applied both segmentation and recognition processes in the text-based handwritten CAPTCHA scheme, though only the segmentation process in the interactive handwritten CAPTCHA scheme, as the recognition process is useless for this scheme. Specifically, once the sample's letters in the interactive handwritten CAPTCHA scheme are segmented, users pinpoint (i.e. recognize) the letter joints, which is the key purpose of this scheme. Each security evaluation process (i.e. pre-processing, segmentation and recognition) is explained in the next sections.

1) PRE-PROCESSING

Pre-processing converts a CAPTCHA image to black and white by removing as much noise from the image as possible. In this study, we used a GSA CAPTCHA Breaker [11] for pre-processing in both the interactive and text-based CAPTCHA schemes. For the interactive scheme, we generated 500 samples from each level of distortion, yielding 2,000 total samples. For the text-based scheme, there were 500 different samples from each distortion type for 5,000 total samples for both one and different writers. Figure 16(a) shows an interactive CAPTCHA sample before pre-processing, while Figure 16(b) shows the results after pre-processing.

2) SEGMENTATION

After the pre-processing step, the segmentation process is occurred. The segmentation process divides a word in the CAPTCHA image into characters. This process can be accomplished through different methods and algorithms (e.g. [3], [7]). In this paper, we used a segmentation algorithm specially designed for Arabic connected characters using MATLAB software for both schemes' samples. In particular,



```
##Step 3: Build a Model and Save it
# SVC stands for support vector classifier

model = svm.SVC(kernel="linear" )
#model = svm.SVC()
# To give all training points for model
model.fit(X_train,Y_train)

# Save our model
joblib.dump(model, "model/CharLabels")

# Make prediction
print ("predicting .....")
predictions = model.predict(X_test)


##Step4 : Print Accuracy
print ("Model Score or Accuracy is :", metrics.accuracy_score(Y_test, predictions))
```

FIGURE 17. A sample Python code to run the SVM algorithm.

we applied it to 2,000 samples from the interactive scheme and 5,000 samples from the text-based scheme.

3) RECOGNITION

The recognition process aims to identify the characters in a CAPTCHA image. For the interactive CAPTCHA scheme, we did not use this process, as we explained previously. Meanwhile, for the text-based CAPTCHA scheme, we used a Google API [12] as a new and highly sophisticated OCR engine to recognize the CAPTCHA images' characters. All 5,000 text-based CAPTCHA images were subsequently fed to this OCR.

To support the recognition process, we applied several machine learning (ML) algorithms to measure the text-based CAPTCHA scheme's ability to resist against the characters recognition. These algorithms were both linear and non-linear and included Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbours, Classification and Regression Trees, Gaussian Naive Bayes and Support Vector Machines (SVM). We selected these algorithms due to their high performance and encouraging results in many studies [13], [14].

Specifically, we arranged the outputs of the segmentation process, or the characters from each sample, in folders based on character. After that, we created our dataset by converting all character images into binary data, labelling them per character and storing them as dataset files. Afterwards, we divided our dataset into two sets: a training set that included 80% of the dataset, and a test set that included the remaining 20%. Based on this, we ran the ML algorithms on the dataset. Then, we created a model for each algorithm and ran them on the test dataset to measure learning accuracy. In this part of the recognition, we used the Python programming language. Figure 17 demonstrates a code used to measure the accuracy of the SVM algorithm.

V. RESULTS

This section presents the results of the evaluated target schemes in terms of usability and security while using mobile device applications.

A. USABILITY RESULTS

To evaluate human performance, we measured the following metrics:

TABLE 3. Usability results.

Measurement (average)	Interactive scheme	Handwritten scheme
Efficiency (in seconds)	5.33	13.35
Effectiveness	96.19%	52.74%

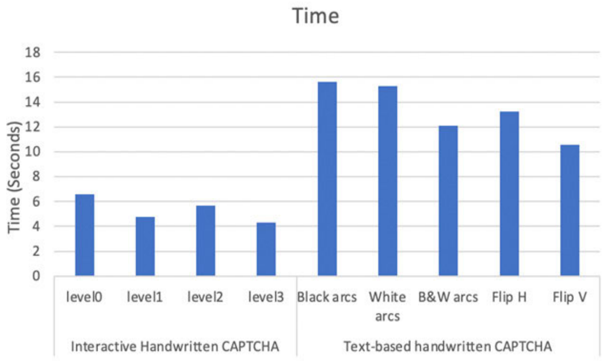


FIGURE 18. Efficiency of both CAPTCHA schemes.

- **Efficiency:** The time (in seconds) that elapses between the moment a CAPTCHA is shown to the user and the moment when the user clicks the "Next" button on the developed interface.
- **Effectiveness:** The correctness of typing the shown characters for the text-based handwritten CAPTCHA scheme, or the degree of conformity and correctness of indicating the joints between letters in the displayed characters in the interactive CAPTCHA scheme.

Table 3 shows the efficiency and effectiveness results for both schemes. From the table, we can infer that there is a significant difference between the schemes per these measures. The usability results reveal the low average time consumption to solve the interactive CAPTCHA scheme tests at 5.33 seconds, compared to 13.35 seconds for the text-based CAPTCHA scheme tests. Additionally, the interactive scheme's average effectiveness was 96%, while the text-based scheme's was 52.74%.

The detailed efficiency results for both schemes are shown in Figure 18, and their detailed effectiveness results are shown in Figure 19.

B. SECURITY RESULTS

This section presents the security results, explaining in particular the segmentation and recognition processes.

1) SEGMENTATION RESULTS

The result of the segmentation process could fall under one of the following categories:

- **Not segmented:** The segmentation algorithm does not find any joint.

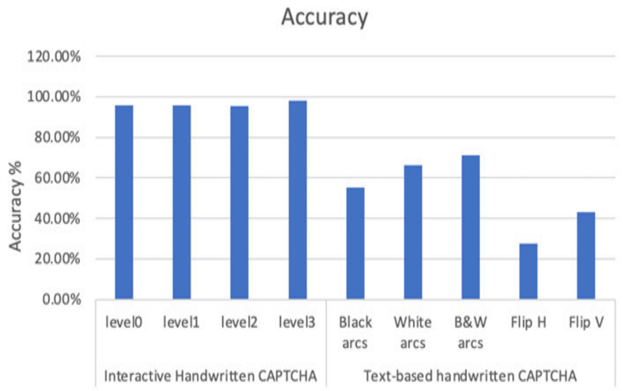


FIGURE 19. Effectiveness of both CAPTCHA schemes.

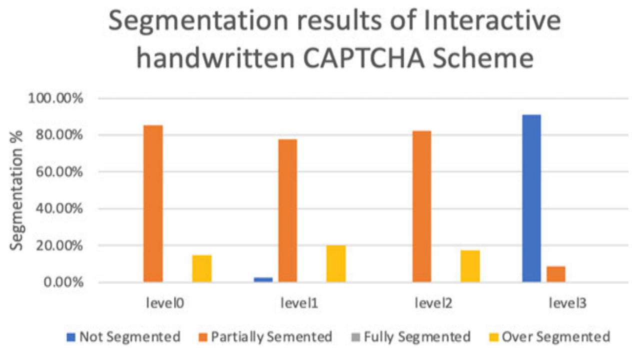


FIGURE 20. Segmentation results for the interactive handwritten Arabic CAPTCHA scheme.

- **Partially segmented:** The segmentation algorithm finds one or more joints, but not all of them.
- **Fully segmented:** The segmentation algorithm finds all joints.
- **Over-segmented:** The segmentation algorithm finds more than the actual joints.

The results of both schemes' segmentation processes are explained based on these categories as follows. Figure 20 summarises the interactive scheme's the average segmentation results. In particular, Level 3 distortion performs the best in terms of segmentation resistance, as it had the lowest not-segmented samples at 91%. Furthermore, Level 0 was 85% partially segmented, Level 1 was 77% partially segmented and Level 2 was 82% partially segmented.

Figures 21 and 22 summarize the segmentation results for the text-based scheme. Specifically, the segmentation results for the black arc distortion type had 21% not-segmented, 69% partially segmented and 9% over-segmented samples. The segmentation results for the white arcs were 12% partially segmented, 0.20% fully segmented and 87% over-segmented. Further, the results of segmenting samples with both black and white arcs were 3% not segmented, 51% partially segmented and 45% over-segmented. Additionally, the results of segmenting horizontal and vertical flips were 81% and 82%, respectively for partially segmented, and 18% and 17% for over-segmented. Interestingly, the segmentation results show no significant difference between one and different writers.

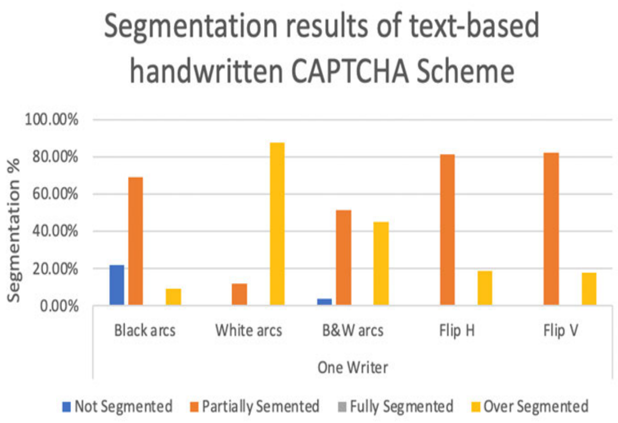


FIGURE 21. Segmentation results for the text-based handwritten Arabic CAPTCHA scheme – One writer.

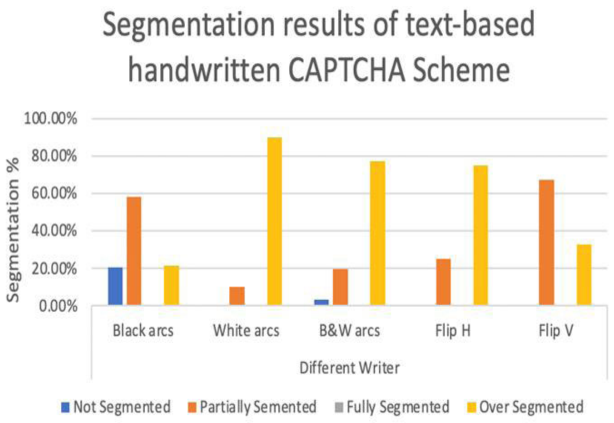


FIGURE 22. Segmentation results for the text-based handwritten Arabic CAPTCHA scheme – Different writers.

2) RECOGNITION

The recognition process results are based on the Google API recognition engine and the set of ML algorithms. The results of the Google API recognition are divided into three groups:

- **Not recognized:** The letters are not recognized.
- **Partially recognised:** One or more letters are recognised, but not all of them.
- **Fully recognized:** All letters are fully recognized.

The results of the recognition using the Google API based on these groups are summarized in Figure 23.

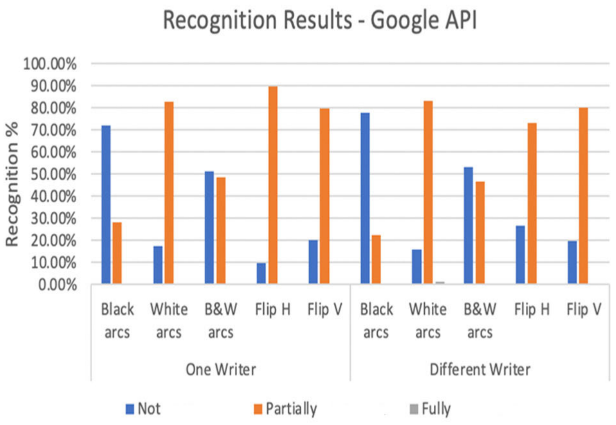


FIGURE 23. Google API recognition results.

TABLE 4. ML algorithm recognition results.

Algorithm	Results
Logistic Regression	0.68
Linear Discriminant Analysis	0.62
K-Nearest Neighbours	0.66
Classification and Regression Trees	0.68
Gaussian Naive Bayes	0.18
Support Vector Machines	0.44

In particular, the results of recognizing black arcs on letters written by one writer were 72% not recognized and 28% partially recognized. For white arcs, the results were 17% not recognized, 82% partially recognized and 0.4% fully recognized. For black and white arcs, the results were 51% not recognized and 48% fully recognized. For the horizontal and vertical flips, the results were 9% and 20% not recognized, respectively. Moreover, the horizontal and vertical flips were each partially recognized at 89% and 79%.

The recognition results for samples written by different writers are very close to the results by one writer, as shown in Figure 22.

The second part of the recognition process utilised the ML algorithms. The results of this step are shown in Table 4, with Logistic Regression and Classification and Regression Tree algorithms performing the best at 68% each. Surprisingly, the Gaussian Naive Bayes algorithm performed the worst at 18%, as it was expected to be better than this performance.

VI. DISCUSSION

The evaluation results demonstrated that the interactive handwritten Arabic CAPTCHA scheme performed better than the text-based handwritten Arabic CAPTCHA scheme in both usability and security. Specifically, the interactive scheme's effectiveness was 96% and its efficiency was less than 6 seconds. On the other hand, the effectiveness of the text-based scheme was 52% and its efficiency more than 13 seconds.

TABLE 5. Summary of the comparison results between our results and other schemes' result.

Study	Scheme	Accuracy	Response time
Our study	Interactive	**96%**	**5 seconds**
	Handwritten	52%	13 seconds
[3]	Interactive	60%	8 seconds
[2]	Handwritten	73%	14 seconds
[27]	FunCaptcha	81.6%	8.2 seconds
[27]	KeyCAPTCHA	84.10%	8.8 seconds
[27]	ReCAPTCHA v2	95%	5.9 seconds
[27]	sweetCaptcha	96%	5.5 seconds
[27]	TapCHA v2	97%	4.95 seconds
[27]	visualCAPTCHA	96%	6 seconds

TABLE 6. Summary of the recognition results.

Study	OCR engine	Recognition result
Our study	Google API [12]	63%
[2]	ABBYY [15]	0%

When comparing these results with the schemes' original results, the text-based scheme's effectiveness using horizontal flips was 27% in our study but 72% in [2]. Additionally, the interactive scheme's effectiveness was 60% in [3] but 96% in this study. For efficiency, the text-based scheme's average time was 13 seconds in this study but 14 seconds in [2]. Meanwhile, for the interactive scheme's efficiency results, the average time was 5 seconds in our study but 8 seconds in [3]. Thus, the results of the interactive scheme in our study showed a promising result (Table 5).

It is interesting to note that our results are benchmarked against the results given in [27] as shown in Table 5. Although the performance results of TapCHA v2 scheme seem competitive to our Interactive scheme, the samples size used in [27] was too small that would reduce the power of the study and increase the margin of error.

Furthermore, the text-based scheme's recognition results were greater in our study compared to its original study results [2]. In particular, using the Google API in our study enhanced recognition by 63%, as shown in Table 6.

VII. CONCLUSION AND FUTURE WORKS

In this experimental study, we regenerated the interactive and text-based handwritten Arabic CAPTCHA schemes for mobile device applications to evaluate their usability and security. The usability results showed that the effectiveness and efficiency of the interactive scheme are better than those of the text-based scheme. Not only that, but also the interactive scheme is more resistant to attacks. Interestingly, the results of recognizing the text-based scheme's images were enhanced in our study. Overall, though, the interactive scheme seems more suitable for mobile device applications.

Our on-going research can help improve segmentation algorithms. In addition, we would like to extend our usability study to involve more participants. This study could also be applied to research on different mobile devices.

REFERENCES

[1] A. L. Von, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Commun. ACM*, vol. 47, no. 2, pp. 56–60, 2004, doi: 10.1145/966389.966390.

[2] S. A. Alsuhibany, F. N. Almohaimeed, and N. A. Alrobah, "Synthetic Arabic handwritten CAPTCHA," *Int. J. Inf. Comput. Secur.*, vol. 1, no. 1, p. 1, 2021.

[3] M. T. Parvez and S. A. Alsuhibany, "Segmentation-validation based handwritten Arabic CAPTCHA generation," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101829.

[4] M. H. Aldosari and A. A. Al-Daraiseh, "Strong multilingual CAPTCHA based on handwritten characters," in *Proc. 7th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2016, pp. 239–245, doi: 10.1109/IACS.2016.7476118.

[5] C. Tangmanee, "Effects of text rotation, string length, and letter format on text-based CAPTCHA robustness," *J. Appl. Secur. Res.*, vol. 11, no. 3, pp. 349–361, Jul. 2016, doi: 10.1080/19361610.2016.1178553.

[6] S. A. Alsuhibany, "Optimising CAPTCHA generation," in *Proc. 6th Int. Conf. Availability, Rel. Secur.*, Aug. 2011, pp. 740–745, doi: 10.1109/ARES.2011.114.

[7] S. A. Alsuhibany and M. T. Parvez, "Secure Arabic handwritten CAPTCHA generation using OCR operations," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 126–131, doi: 10.1109/ICFHR.2016.0035.

[8] S. Kulkarni and H. S. Fadewar, "Pedometric CAPTCHA for mobile internet users," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 600–604, doi: 10.1109/RTEICT.2017.8256667.

[9] M. Guerar, A. Merlo, and M. Migliardi, "Completely automated public physical test to tell computers and humans apart: A usability study on mobile devices," *Future Gener. Comput. Syst.*, vol. 82, pp. 617–630, May 2018, doi: 10.1016/j.future.2017.03.012.

[10] Khatt.ideas2serve.net. (2020). *KAHTT Database*. Accessed: Mar. 14, 2020. [Online]. Available: http://khatt.ideas2serve.net/

[11] GmbH GSA, GSA Captcha Breaker. *Breaks Any Captcha—Works With Any Software*. Accessed: Jul. 18, 2020. [Online]. Available: https://www.gsa-online.de/product/captcha_breaker/

[12] (2021). *Vision AI | Derive Image Insights Via ML| Cloud Vision API*. Google Cloud. Accessed: Jan. 13, 2021. [Online]. Available: https://cloud.google.com/vision

[13] H. Kusetogullari, A. Yavariabdi, A. Cheddad, H. Grahn, and J. Hall, "ARDIS: A Swedish historical handwritten digit dataset," *Neural Comput. Appl.*, vol. 32, no. 21, pp. 16505–16518, Nov. 2020, doi: 10.1007/s00521-019-04163-3.

[14] R. Karakaya and S. Kazan, "Handwritten digit recognition using machine learning," *Sakarya Univ. J. Sci.*, vol. 25, no. 5.

[15] *ABBYY FineReader 14, PDF Software with Text Recognition—ABBYY FineReader 14 OCR*. Accessed: Feb. 22, 2019. [Online]. Available: https://www.abbyy.com/en-me/finereader/

[16] M. Moradi and M. Keyvanpour, "CAPTCHA and its alternatives: A review," *Secur. Commun. Netw.*, vol. 8, no. 12, pp. 2135–2156, 2015.

[17] W. K. Abdullah Hasan, "A survey of current research on CAPTCHA," *Int. J. Comput. Sci. Eng. Surv.*, vol. 7, no. 3, pp. 1–21, Jun. 2016.

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

[19] (2019). *Internet World Stats*. Internet Usage in the Middle East. [Online]. Available: https://web.archive.org/web/20190516191014/ and [Online]. Available: https://www.internetworldstats.com/stats5.htm

[20] S. A. Sattar, S. Haque, M. K. Pathan, and Q. Gee, "Implementation challenges for nastaliq character recognition," in *Proc. Int. Multi Topic Conf.* Berlin, Germany: Springer, 2008, pp. 279–285.

[21] A. AL-Shatnawi and K. Omar, "A comparative study between methods of Arabic baseline detection," in *Proc. Int. Conf. Electr. Eng. Informat.*, Aug. 2009, pp. 73–77.

[22] A. Khalil, S. Abdallah, S. Ahmed, and H. Hajjdiab, "Script familiarity and its effect on CAPTCHA usability," *Int. J. Web Portals*, vol. 4, no. 2, pp. 74–87, Apr. 2012, doi: 10.4018/jwp.2012040105.

[23] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "Persian/Arabic baffletext CAPTCHA," *J. Universal Comput. Sci.*, vol. 12, no. 12, pp. 1783–1796, 2006.

[24] A. Saxena, N. S. Chauhan, S. K. R., A. S. Vangal, and D. P. Rodrguez, "A new scheme for mobile based CAPTCHA service on cloud," in *Proc. IEEE Int. Conf. Cloud Comput. Emerg. Markets (CCEM)*, Oct. 2012, pp. 17–22, doi: 10.1109/CCEM.2012.6354589.

[25] E. Bursztein, M. Martin, and J. Mitchell, "Text-based CAPTCHA strengths and weaknesses," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, 2011, pp. 125–138.

[26] M. Guerar, A. Merlo, M. Migliardi, and F. Palmieri, "Invisible CAPPCHA: A usable mechanism to distinguish between malware and humans on the mobile IoT," *Comput. Secur.*, vol. 78, pp. 255–266, Sep. 2018.

[27] N. Jiang, H. Z. Dogan, and F. Tian, "Designing mobile friendly CAPTCHAs: An exploratory study," in *Proc. 31st Int. BCS Hum. Comput. Interact. Conf. (HCI)*, Jul. 2017, pp. 1–7.

[28] K. Aburada, S. Usuzaki, H. Yamaba, T. Katayama, M. Mukunoki, M. Park, and N. Okazaki, "Implementation of CAPTCHA suitable for mobile devices," *IEICE Commun. Exp.*, vol. 8, no. 12, pp. 601–605, 2019, doi: 10.1587/comex.2019gcl0060.

SULIMAN A. ALSUHIBANY received the M.Sc. degree in computer security and resilience and the Ph.D. degree in information security from Newcastle University, U.K. He is currently an Associate Professor with the Department of Computer Science, College of Computer, Qassim University, Saudi Arabia. He has published in some of the most reputed journals and conferences. His research interests include human aspects of security (e.g. the so-called "usable security"), CAPTCHAs, spam-filter, keystroke dynamics, and information security.

AYSHAH A. ALNOSHAN received the B.Sc. degree in information technology from Qassim University, Saudi Arabia, where she is currently pursuing the M.Sc. degree with the Department of Computer Science, College of Computer. Her research interest includes information security.