# A Proposal of CAPTCHA Using the Combination of Latin and Jawi Script

# A Proposal of CAPTCHA Using the Combination of Latin and Jawi Script

Hisaaki YAMABA[a], Ahmad Saiful Aqmal Bin Ahmad Sohaimi[b],
Kentaro ABURADA[c], Naonobu OKAZAKI[d]

## Abstract

This paper proposes a new type text-based CAPTCHA using both Jawi script and Latin script which are used to describe Maley language. Many web sites come to provide CAPTCHA to prevent bots and other automated programs from signing up for email, spamming comments, violating privacy, and preventing brute force login attacks on user accounts. Text-Based CAPTCHAs are the most common and earliest CAPTCHA. But as optical character recognition (OCR) technology has improved, the intensity of distortions that must be applied to CAPTCHA for them to remain unrecognizable by OCR has increased. This has reached a point where humans are having difficulty in solving the CAPTCHA. The idea of the proposed CAPTCHA is to generate one same string written in Latin scripts and Jawi scripts. Since some of the string' s characters are hidden by the interference, users need to combine both strings to solve this CAPTCHA. This idea is uses that the fact that almost Jawi characters are one-to-one correspondence with Latin characters, A series of experiments was carried out to evaluated the performance of the proposed CAPTCHA. First, some OCR softweres were prepared and it was clarified that they could not read the partially hidden Latin and Jawi strings. Next, A computer program was developed using computational technique such as PHP, JavaScript, and so on for the usability evaluation. 13 experimental subjects participated the experimetns. The results of the experiments showed that the averate time to solve the CAPTCHA and the accuracy rates were acceptable compared with the indices shown in the existing research.

*Keywords:* CAPTCHA, Maley language, Jawi script, Latin script

## 1. INTRODUCTION

CAPTCHA — Complete Automated Public Turing Test to Tell Computers and Humans Apart — is a type of challenge-response test used to distinguish between human users and automated programs. CAPTCHA has become quite common on websites and applications. They are used to prevent bots and other automated programs from signing up for email, spamming comments, violating privacy, and preventing brute force login attacks on user accounts.

A good quality CAPTCHA test should have the following characteristics:

1. Easy for humans to solve.
2. Hard for a computer bot to solve.

CAPTCHA must be highly secure and easy to use. The previous works on CAPTCHA discuss that many versions of CAPTCHA have been proposed, developed, and should be not only difficult to solve by computer programs but also easy for humans to solve.

Text-Based CAPTCHAs are the most common and earliest CAPTCHA. They request users to enter the string of characters that appears in a distorted form on the screen.

However, there is a need to develop a new text-based CAPTCHA that is easy for humans to solve despite increasing the distortion intensity. As optical character recognition (OCR) technology has improved, the intensity of distortions that must be applied to CAPTCHA for them to remain unrecognizable by OCR has increased. This has reached a point where humans are having difficulty in solving the CAPTCHA.

Based on the identified problem, this paper propose a new CAPTCHA that combine two different characters' scripts, which are Latin scripts and Jawi script, into one CAPTCHA.

## 2. RELATED WORK

This section highlights studies of both text-based Latin CAPTCHAs and Arabic CAPTCHAs schemes.

### 2.1 Latin CAPTCHA Scheme

Latin CAPTCHA or CAPTCHA that consists of English letters, are the earliest and most deployed CAPTCHA, typically asking users to recognize the distorted words correctly. The CAPTCHA idea was first implemented by Alta Vista to prevent automated-bots from automatically registering the web sites [5)]. The mechanism behind the CAPTCHA idea was to generate slightly distorted characters and present it to the users. There are many ex-

[a] Assistant Professor, Dept. of Comp. Sci. & Sys. Eng.
[b] Student of Dept. of Comp. Sci. & Sys. Eng.
[c] Associate Professor, Dept. of Comp. Sci. & Sys. Eng.
[d] Professor, Dept. of Comp. Sci. & Sys. Eng.

Table 1 Additional of 6 Jawi characters

| Character | Final | Medial | Initial | Isolated |
|---|---|---|---|---|
| چ | ـچ | ـچـ | چـ | چ |
| ڠ | ـڠ | ـڠـ | ڠـ | ڠ |
| ڤ | ـڤ | ـڤـ | ڤـ | ڤ |
| ݢ | ـݢ | ـݢـ | ݢـ | ݢ |
| ۏ | ـۏ |  |  | ۏ |
| ڽ | ـڽ | ـڽـ | ڽـ | ڽ |

amples of using Latin script in a CAPTCHA, such as Gimpy CAPTCHA, Ez-Gimpy CAPTCHA, and Baffle-Text CAPTCHA. Gimpy CAPTCHA selects several words from the dictionary and displays all the distorted words to the users. While Ez-gimpy only displays one distorted word. Baffle-Text CAPTCHA [6] is a modified version of Gimpy that generates a random meaningless word as a CAPTCHA.

### 2.2 Arabic CAPTCHA Scheme

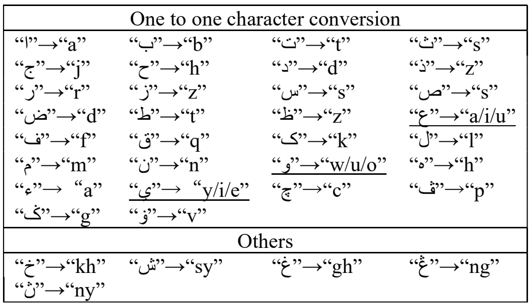
So far, no CAPTCHA has been implemented using Jawi script, but many studies have been done on Arabic CAPTCHAs. Since Jawi script is almost similar to the Arabic script, the author explain some Arabic CAPTCHA schemes. The first work that employing Arabic script in the CAPTCHA field is present in [1] that generates random meaningless Arabic words as CAPTCHA. In particular, the work reported in [1] presents an application of Persian/Arabic CAPTCHA, while the work in [2] applies Arabic CAPTCHA for verifying spam SMS. Khan et al. [3] improved the previous work of typed-text Arabic CAPTCHA. Specifically, they exploited Arabic OCRs' limitations in reading Arabic text by adding background noise and using specific Arabic font types in CAPTCHA generation. The study in [4] proposed advanced Nastaliq CAPTCHA that provides essentially random meaningless Persian words that are close to Arabic words in terms of script.

## 3. JAWI SCRIPT : AN OVERVIEW

This section explains Jawi script's characteristics in terms of origin, writing direction, and shapes.

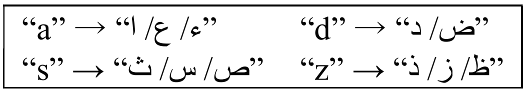
In the previous years, Jawi script became one of the first script used among Malaysian, Indonesian, and Bruneian. Nowadays, Jawi script is still included in the educational module, specifically in the Asian countries such as Malaysia, Indonesia, and Brunei. Jawi script is almost similar to the Arabic script except for 6 letters that are shown in Table 1. Jawi script contains 35 letters, and they are written from right to left like Arabic. Jawi scripts have different shapes depending on their position in the word, i.e., initial, middle, final, or isolated. In contrast to Arabic scripts, Jawi scripts could be written as in the Malay language.

In writing Malay Language, almost Jawi characters are one-to-one correspondence with Latin characters as shown in Fig. 1. The underlined characters in Fig. 1 are the characters in Jawi script that represent more than one character in Latin script depending on its context. There are also several Jawi characters that represent the same Latin characters based on the word used as shown in Fig. 2.



| One to one character conversion | | | |
|---|---|---|---|
| "ا"→"a" | "ب"→"b" | "ت"→"t" | "ث"→"s" |
| "ج"→"j" | "ح"→"h" | "د"→"d" | "ذ"→"z" |
| "ر"→"r" | "ز"→"z" | "س"→"s" | "ص"→"s" |
| "ض"→"d" | "ط"→"t" | "ظ"→"z" | "ع"→"a/i/u" |
| "ف"→"f" | "ق"→"q" | "ک"→"k" | "ل"→"l" |
| "م"→"m" | "ن"→"n" | "و"→"w/u/o" | "ه"→"h" |
| "ء"→ "a" | "ي"→ "y/i/e" | "چ"→"c" | "ڤ"→"p" |
| "ݢ"→"g" | "ۏ"→"v" | | |
| **Others** | | | |
| "خ"→"kh" | "ش"→"sy" | "غ"→"gh" | "ڠ"→"ng" |
| "ڽ"→"ny" | | | |

Fig. 1 Conversion of Jawi script to Latin script



"a" → "ا / ع / ء"  "d" → "د / ض"

"s" → "ث / س / ص"  "z" → "ذ / ز / ظ"

Fig. 2 Multiple conversion of Latin characters

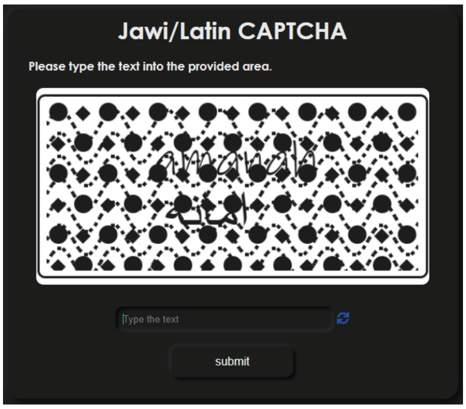


Fig. 3 Proposed CAPTCHA

Jawi script are expected to provide better security against OCR software, as many Jawi characters share same main body and differ only in the number of dots. Thus, this paper proposes an alternative solution by providing another text-based CAPTCHA using the combination of Jawi script and Latin script.

## 4. PROPOSED SCHEME

This section describes the basic idea of the proposed CAPTCHA, the text generation process of the CAPTCHA, and the distortion patterns of the CAPTCHA.

This paper proposes a new CAPTCHA combining two different characters' scripts: Latin scripts and Jawi script, into one CAPTCHA, as shown in Fig. 3. The idea of this CAPTCHA is to generate one same string written in Latin scripts and Jawi scripts. To solve this CAPTCHA, the users need to combine both strings written in Latin script

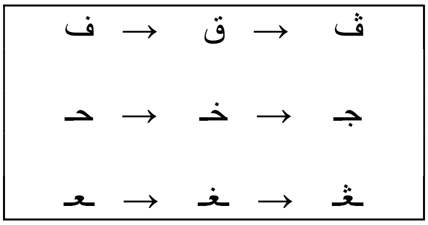


Fig. 4 Jawi characters differ in the number of dots

and Jawi script as the string's characters are hidden by the interference.

For text generation, the proposed scheme consists of two types, which are (1) Malay word that is randomly selected from the dictionary or (2) a random meaningless Latin/Jawi string. For type (1), one word is randomly selected from a database and its two spellings, one is spelled in Latin script and the other is spelled in Jawi script, are given. Then, the selected word is developed into an image. For type (2), a string is randomly generated using Latin script with 4 to 8 characters length. Then, the generated Latin string is converted to a Jawi string. Lastly, both generated strings are developed into an image.

In this CAPTCHA scheme, two distortion patterns are both introduced: pattern 1 and pattern 2 as shown in Fig. 5. The distortion pattern's primary purpose is that the generated CAPTCHA should survive OCR attacks while being readable to a human.

Pattern 1 uses dotted sine waves as the background because Jawi characters have a cursive shape. In addition, black dots are added to the CAPTCHA image as many Jawi characters share the same main body and differ only in the number of dots, as shown in Fig. 4. This kind of background can confuse the OCR software to recognize the characters. The black circles are used to hide the character in Jawi and Latin words while ensuring that the same character hidden in Latin word is not hidden in Jawi word, as shown in Fig. 5 (a).

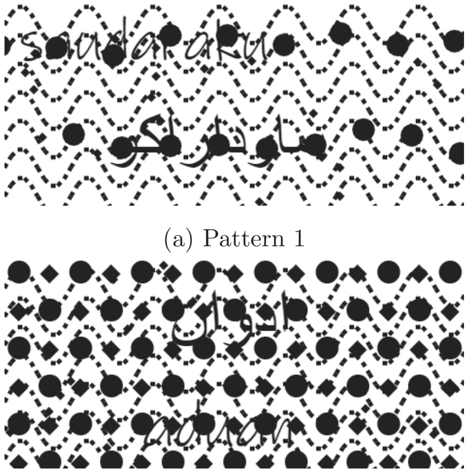
Pattern 2 also uses dotted sine waves and black dots as a background, plus the black circles and black squares are being used as the distortion, as shown in Fig. 5 (b).

## 5. SECURITY EVALUATION

This section describes the security evaluation experiment of the proposed method and the results of the experiment.

### 5.1 Purpose and Conditions

The purpose of security evaluation is to find how secure the method is against bot attacks. To prove that the proposed CAPTCHA is secured, a security experiment was conducted using modern OCR software: Tesseract and ABBYY FineReader. This software comes with a Malay dictionary that allows it to detect Malay words. The CAPTCHA samples for each type of string and pattern are generated and tested, as shown in Table 2.



(a) Pattern 1

(b) Pattern 2

Fig. 5 Proposed CAPTCHA patterns

Table 2 CAPTCHA samples for experiment

| Type | Pattern 1 | Pattern 2 |
|---|---|---|
| Malay Words | 10 images | 10 images |
| Meaningless Words | 10 images | 10 images |

Table 3 Security evaluation results

| | MW | | RMW | |
|---|---|---|---|---|
| | $P_1$ | $P_2$ | $P_1$ | $P_2$ |
| Tesseract OCR | 0 | 0 | 0 | 0 |
| ABBYY FineReader | 0 | 0 | 0 | 0 |

### 5.2 Results

The result for the word recognition accuracies for security evaluation is shown in Table 3 where MW denotes Malay Words, RMW denotes Random Meaningless Words, $P_1$ denotes Pattern 1 and $P_2$ denotes Pattern 2. Modern Tesseract OCR, as well as ABBYY FineReader, failed to detect the word from the proposed CAPTCHA.

## 6. USABILITY EVALUATION

This section describes the usability evaluation experiment of the proposed method and the results of the experiment.

### 6.1 Purpose and Conditions

This experiment was divided into two parts, which are experiment 1 and experiment 2. Experiment 1 was conducted to check human users' accuracy in reading only one of the strings written, whether in Jawi script or Latin script. Experiment 2, which was the most important part of the experiment, was conducted to estimate humans' accuracy in reading both generated strings together.

There was a total of seven participants for experiment 1 and a total of 13 participants for experiment 2. Participants for this experiment are all able to read Jawi script and speak the Malay Language. The CAPTCHA images

Table 4 Usability evaluation experiment 1 results

| | MW | | | | RMW | | | |
|---|---|---|---|---|---|---|---|---|
| | $P_1$ | | $P_2$ | | $P_1$ | | $P_2$ | |
| | Latin | Jawi | Latin | Jawi | Latin | Jawi | Latin | Jawi |
| Average Time | 7.5 s | 17.9 s | 7.0 s | 14.8 s | 12.0 s | 27.3 s | 10.0 s | 21.2 s |
| Accuracy Rate | 91.4% | 37.1% | 85.7% | 54.3% | 68.6% | 8.6% | 65.7% | 20.0% |

Table 5 Usability evaluation experiment 2 results

| | MW | | RMW | |
|---|---|---|---|---|
| | $P_1$ | $P_2$ | $P_1$ | $P_2$ |
| Average Time | 6.8 s | 6.0 s | 10.1 s | 10.0 s |
| Accuracy Rate | 97.7% | 96.9% | 92.3% | 88.5% |

used for both experiment are generated using the scheme described in the previous section, with 2 distortion patterns (Pattern 1 and Pattern 2) and 2 types of string (Malay Words and Random Meaningless Words). The CAPTCHA image samples were generated, as shown in Table 2.

Both experiments were carried out according to the following procedure. First, the general procedure was explained to the participant before starting the experiment. After that, participants need to answer a practice section to understand the whole experiment process. Then, the main experiment begins starting with Malay words, followed by random meaningless words.

During the experiments, participants are asked to recognize and write the words displayed on the screen into the text box and submit them by clicking on the submit button. The system records the submitted answers and the time taken to solve the CAPTCHA. Each participant was also asked to answer a short survey about their experience.

### 6.2 Results

In the usability study, we measure the following two outcome metrics:

> **Time taken:** The time (in second) elapsed between the time the CAPTCHA image was shown to the user and when the 'Submit' button was clicked.
>
> **Accuracy:** The degree of conformity and correctness of typing a shown CAPTCHA.

For experiment 1, although we expected that it is difficult to solve the proposed CAPTCHA by reading only one of the two scripts, the Latin strings could lead users to the correct answers in many cases. Table 4 shows the average time taken and accuracy data for each type of CAPTCHA pattern collected from experiment 1. The results for experiment 1 showed that the accuracy results for Latin strings were higher than 60.0% for all types of strings and patterns. The results were quite different from what we expected, which would be lower if only reading one of the strings, and when combined with both strings, the accuracy rate would be better. The size of distortion and the type of font used may be the reason for this result to happen, and further studies should take this into account to improve the proposed CAPTCHA. On the other hand, it was confirmed that the two distortion patterns were both effective by the comments from many participants, especially for Jawi scripts.

For experiment 2, regardless of the type of string and pattern used for this experiment, the accuracy results seem promising for all types of strings and patterns. Table 5 shows the average time taken and accuracy data for each type of CAPTCHA pattern collected from experiment 2. The previous study in [7] stated that humans' average time and accuracy rate in solving the current text-based CAPTCHA image is 9.8 seconds and 87.0%. Thus, the average time taken and accuracy rate to solve this proposed CAPTCHA is acceptable.

## 7. CONCLUSION

This paper proposed a method using both Jawi script and Latin script to generate CAPTCHA. By combining both scripts, we can keep the CAPTCHA easy for humans to recognize despite increasing the CAPTCHA distortion intensity to a level that cannot be broken by a bot. To the best of our knowledge, this paper is the first to combine Jawi script and Latin script as a CAPTCHA scheme.

From the overall results, the proposed CAPTCHA can be solved by human users with far better effectiveness compared to modern OCR software. The efficiency of solving the proposed CAPTCHA is also excellent.

In the future, we intend to modify the proposed method by increasing the distortion level for Latin script and reducing the distortion level for Jawi script to improve the performance of the CAPTCHA against the bot and human.

## References

1) 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).

2) M. S. Shahreza: Verifying Spam SMS by Arabic CAPTCHA, in 2nd IEEE International Conference on Information and Communication Technologies (ICTTA'06), pp. 78-83, (2006).

3) B. Khan, K. Alghathbar, M. K.Khan, A. M.AlKelabi, and A. Alajaji: Cyber security using arabic captcha scheme, Int. Arab J. Inf. Technol., vol. 10, no. 1, pp. 76-84, (2013).

4) M. H. Shirali-Shahreza and M. Shirali-Shahreza: Advanced Nastaliq CAPTCHA, 7th IEEE International Conference on Cybernetic Intelligent Systems, London, pp. 1-3, (2008).

5) Lillibridge M., Abadi M., Bharat K., and Broder A: Method for Selectively Restricting Access to Computer Systems, United States Patent 6195698. Applied 1998 and Approved 2001.

6) H. S. Baird and M. Chew: BaffleText: a Human Interactive Proof, Proceedings of the 10th SPIE/IST Document Recognition and Retrieval Conference, Santa Clara, CA, 305-316, 2003.

7) E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell and D. Jurafsky: How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation, IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, 2010, pp. 399-413, (2010).