

DESIGN OF AN ACCENTED CHARACTER-BASED TEXT CAPTCHA SYSTEM

¹Olanrewaju, O. T., ²Omilabu, A. A., ³Asoro, B. O., ⁴Nwufor, C. V., ⁵Adewale, F. O. and
⁶Osunade, O.

^{1,4,5}Department of Computer Science, FCAH&PT, Apata, Ibadan,

²Department of Computer and Information Science, Tai Solarin University,

^{3,6}Department of Computer Science, University of Ibadan, Nigeria

¹Omowamiwa.tundetaiwo@fcahptib.edu.ng, ²omilabuaa@tasued.edu.ng,

³blessinggraymond4@gmail.com,

⁴chinonyelum.tabansi@yahoo.com, ⁵phummi03@yahoo.com, ⁶o.osunade@ui.edu.ng

Abstract

For the purpose of securing online transactions, the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) uses text, image, audio, and video as authentication techniques. The first and most popular version of CAPTCHA is text-based, but breaches are possible since CAPTCHA only accepts Latin characters. The use of other characters in text-based CAPTCHA design is being explored because the benefits of text-based CAPTCHA outweigh other types of CAPTCHAs. The goal of the study was to design a CAPTCHA system that would strengthen the security of online transactions by using Latin and accented characters from two Nigerian languages, Yoruba and Igbo called NaijaCAPTCHA. Latin and accented characters were used to create the CAPTCHA code types using a modified Gimpy algorithm. The design generated sixteen CAPTCHA code types using a mix of obfuscation methods. NaijaCAPTCHA was designed to generate five letters including two accented characters.

Keywords: NaijaCAPTCHA, authentication, Optical character recognition, text-based CAPTCHA

I. INTRODUCTION

In an effort to prevent unauthorized users from accessing web-based services, websites can make an automatic assessment via the Completely Automated Public Turing Test to Tell Computer and Human Apart (CAPTCHA), a dynamic authentication system first introduced by Luis von Han in 2003. However, several authentication security mechanisms, such as reCAPTCHA, login interfaces, Password Identification Numbers (PIN), One-Time Passwords (OTP), Strong Identity, and biometric-based solutions, have been implemented to protect digital assets. The most popular kind of authentication that web administrators employ is CAPTCHAs (Shivani, 2020).

There have been several CAPTCHA design options developed over the past decade, including text-based, video-based, and image-based ones, to name a few. However, text-based CAPTCHAs are most frequently employed on the web because it is simple to implement. Text-based CAPTCHAs present users with distorted text in images and ask them to identify the text to demonstrate that they are human. Although decoding such deformed characters with the background interference is a simple operation for humans, it is more challenging for machines. (Arain, et al 2018)

The first and most widely used CAPTCHAs are text-based, and they are composed of three different sorts of characters: alphabetic, syllabic, and logographic, with alphabets being the most common representation (Obimbo et al. 2013; Merriam-Webster 2021; Guerar et al.

2022). Because most text-based schemes are created utilizing characters from the English language, history demonstrates that these schemes are broken with a relatively high success rate. The majority of researchers were motivated by this concept to create non-English-based CAPTCHA schemes utilizing their own indigenous languages, such as Chinese, Arabic, Hindi, and Sindhi (Doi & Lei, 2014; Maitlo, et al., 2021; Alsuhibany & Alnoshan, 2021; Kumar et al., 2022). However, these innovations were restricted to uses by the local population.

Other natural languages, though, utilize characters with accents or diacritical's, including French, Spanish, Igbo, and Yoruba. All users throughout the universe can read and understand the characters that were taken from these languages. As a result, employing these characters can improve the creation of text-based CAPTCHA and reduce its susceptibility to assault. In creating a novel text-based CAPTCHA employing accented characters from two Nigerian languages Yorùbá and Igbo, this paper adopted the concept of earlier researchers.

II. RELATED WORKS

A security method known as CAPTCHA helps to distinguish between automated machines and human beings. (Olanrewaju and Osunade, 2017; Henri, 2018). Moni Naor initially proposed CAPTCHA, or reverse Turing tests as they are more generally known, in his 1996 work titled "Confirmation of a Human in the Loop" as a test or task that a Human would excel at but a Machine would flunk.

Based on research, Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford originally suggested CAPTCHA based on text in 2003, when a series of characters and numbers were inputted in a warped image. The text-based CAPTCHA replaced the present system's only reliance on text passwords by combining characters with various alphanumeric string types, including letters, numbers, and special characters. A password with an alphanumeric combination of letters and digits was also made. This inventiveness is used by Uma et al. (2019), Li et al. (2020), Alsuhibany & Alnoshan (2021), Maitlo et al. (2021), and Conti et al. (2022) to construct a number of text-based CAPTCHAs.

A novel text-based CAPTCHA system that allows users to differentiate alphabets from highly distorted by adding a blend of upper- and lower-case letters with a high degree of distortion was developed in response to vulnerability attacks that text-based CAPTCHA encountered. (Tangmanee, C. (2018): Tharad et al. 2018). This method has been designed so that a computer program will struggle to decipher the pattern of text, but a human will have no trouble understanding it.

However, Shivani, (2021) and Chi et al, 2022 also designed another text-based CAPTCHA employing text and Immutable Adversarial Noise (IAN) and advCAPTCHA respectively to improve the security issues. Maitlo, et al., 2021; Alsuhibany & Alnoshan, 2021; Kumar et al., 2022 build a novel text-based CAPTCHA employing distinct characters collected from their various languages to address flaws in the development of the current text-based CAPTCHA. The developed system's accuracy and users' reactions revealed improvements. These developments resemble our inventiveness.

There are three main languages spoken in Nigeria: Igbo, Yorùbá, and Hausa. Only Igbo and Yorùbá, whose character sets are determined by the Universal Character Set (UCS), utilize characters with accents or diacritical's that are not included in the Latin character set. According to the Unrepresented Nations and Peoples Organization (UNPO) (2020), a sizable

ethnic minority speaks the Yorùbá language in the Southern region of the Republic of Benin and the South-Western province of Nigeria in Western Africa. To depict Yoruba tonality and to accommodate the need to represent speech sounds that are outside the scope of the fundamental ANSI characters, the following basic characters can only be used: á, à, è, é, ẹ, ẹ́, ẹ̀, í, ì, ò, ó, ọ, ọ́, ọ̀, ù, ú, ɿ.

On the other side, people in Southeast Nigeria speak the Igbo language. Eight vowels (A, E, I, O, and U) and 28 consonants (GB, GH, GW, H, J, K, P, KW, L, M, N, NW, NY, P, R, S, SH, T, V, and W), also known as *udaume* and *mgbochiume*, respectively, make up the Igbo alphabet. In addition, the Igbo vowels can be divided into two groups: the light group (*Udamfe*: A I O U) and heavy group (*udaaru*: E I O U). In addition, the characters that have been extracted from these two Nigerian languages can be employed in the same way that Gimpy has used other characters to create text-based CAPTCHA.

Gimpy is one of the most dependable systems created for and in cooperation with Yahoo! It prevents bots from accessing their chat rooms, scripts from collecting an excessive amount of their email addresses, and computer programmes from publishing classified advertising.

Gimpy is predicated on the fact that while automate programs cannot read text that has been severely damaged and mangled, humans can. Gimpy works by selecting a predetermined number of words from a dictionary, corrupting and distorting them, and then asking the user to type the words that were displayed in the image. The words displayed are effortlessly inputted by human users, but automated programs lack the ability to do the same. Examples are shown in Fig 1

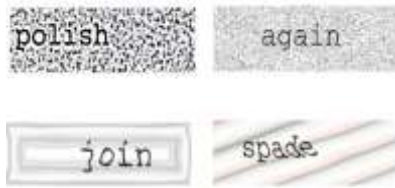


Fig 1a: EZ-Gimpy

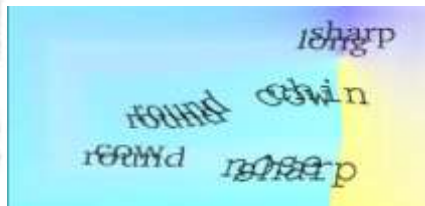


Fig 1b: Gimp

Characters were corrupted and warped using the obfuscator method. Obfuscation is the process of making something difficult to understand, confusing, or obscure. It is a method for transforming code into a form that is semantically equivalent to the original programme but challenging to interpret. Obfuscation will significantly reduce the file size, allowing for quick data transfers between the server and client.

Obfuscator is a software protection technique based on efficient code obfuscation; creating an efficient obfuscation algorithm can raise the expense of reverse engineering software. Making the generated codes difficult to understand, typically with complex and ambiguous wording using multiple backgrounds, obscures the intended meaning of communication. (Li et al 2022)

III. METHODOLOGY

This study used a quantitative methodology and was empirically conducted. The research phase entailed developing a CAPTCHA system based on accented characters. The CAPTCHA generator, obfuscator, CAPTCHA display, and database are the four modules that

make up the concept. The created accented character-based CAPTCHA system, which is based on the Gimpy scheme, is shown in Figure 1. The relationships and purposes of the four modules are explained.

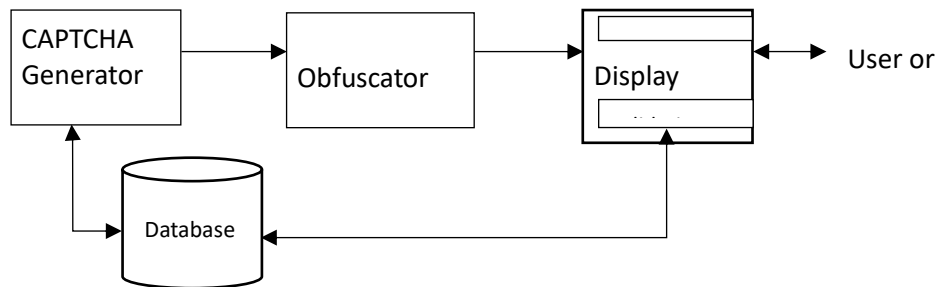


Figure 1 Accented Character-based CAPTCHA System

1. CAPTCHA Generator: This module uses the algorithm that randomly generates five characters to form the CAPTCHA code. Accented characters from Yorùbá and Igbo alphabet were used in forming the CHAPTCHA code with: (1) minimum length of characters, and (2) number of accented characters.
2. Obfuscator: This module employs an algorithm that serves as a security mechanism to make it difficult for automated software or bots to decipher the generated CAPTCHA code. The final code generated is unaffected by the manipulation's order. Consequently, the generated text can be altered both before and after the background is altered.
3. CAPTCHA Display: This module displays the accented CAPTCHA code that was created while asking the user to enter information. The three techniques discussed in this module are applied as follows:
 - a. User Keypad: This device has a virtual keyboard that makes it simple for users to input accented characters.
 - b. The Validation Unit: is made up of the response checker and response matcher.
 - i. Response Checker: This component analyzes the CAPTCHA input field while awaiting a user response.
 - ii. Response Matcher: When a user input is received, this device activates. By comparing the user's input to the database record for equality, the user input is verified.
4. Database: This module stores the CAPTCHA code created by the CAPTCHA generator as well as the character set it uses. It is accessed by the Validation unit to verify the accuracy or correctness of the response given by the user or Solver.

IV. RESULTS AND DISCUSSION

The design process gave rise to the following model and CAPTCHA code types that are presented here using algorithm, images and tables.

<p>Step 1: Start the session</p> <p>Step 2: Gen_Xter← Generate random string from characters stored – Latin and “àáèèèèìíðòóòùúşøęññ”. // string of accented character to be combined with Latin character to form the CAPTCHA code</p> <p>Step 3: Sel_Xter←Randomly generate 5 characters (Latin+ accented) // randomly generates five characters to form the CAPTCHA code</p> <p>Step 4: Cap_Tpye←Randomly select a CAPTCHA type // CAPTCHA code generated to produce CAPTCHA type</p> <p>Step 5: Cap_EmptyCanvas← Initialize an empty canvas for drawing CAPTCHA //create a container that hold the generated CAPTCHA code</p> <p>Step 6: Display accented characters on the canvas // show the five generated accented characters on the container</p> <p>Step 7: Draw canvas background // shows the background of the container</p> <p>Step 8a: Accept User input from the keyboard // the virtual keyboard to use by the user</p> <p>8b: If the user input is equal to generated accented characters: //authentication process</p> <p>8c: Save user response into the database // save user’s correctly and incorrectly solved codes into the database</p> <p>8d: Redirect to a page where it displays success message // displays successfully solved</p> <p>8e: Else:</p> <p>Step 9a: Display an Error message to the user // displays code do not match</p> <p>9b: Go To Start // re -start the process</p> <p>Step 10: Stop // end the process</p>

Algorithm 1: Accented CAPTCHA Code Generation









The steps used in generating the code types from NaijaCAPTCHA are given in Algorithm 1. In Step 2, a random string of characters is generated and 5 characters picked in Step 3 to form the CAPTCHA code. The CAPTCHA code type is selected in Step 4 and Steps 5 – 7 generates the code type as an image. The image is displayed to the user in Step 8 while the user's response determines what happens next.












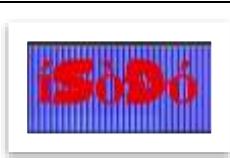
Table 1: Components of Generated CAPTCHA Codes

Background	Generated Characters
no background	Coloured
background with noise	Distorted
coloured background	Collapsed
random lines	Fragmented

Table 1 provides the text and background features used to generate CAPTCHAs code types (Steps 5 – 7). The background types available are four (4) with the text characters having four (4) presentation options: coloured, distorted, collapsed or fragmented.

Table 2: List of CAPTCHAS Types Possible

	CAPTCHA Types	Acronyms	Equivalent CAPTCHA Type
	Text No Background	TNB	
	Text with Background Noise	TBN	
	Text with Coloured Background	TCB	
	Text with Random Lines	TRL	
	Character Collapse No Background	CCNB	
	Character Collapse Background Noise	CCBN	
	Character Collapse with Coloured Background	CCCB	
	Character Collapse with Random Lines	CCRL	

Character Fragmentation No Background	CFNB	
Character Fragmentation with Background Noise	CFBN	
Character Fragmentation with Coloured Background	CFCB	
Character Fragmentation with Random Lines	CFRL	
Coloured Texts No Background	CTNB	
Coloured Texts with Background Noise	CTBN	
Coloured Texts with Coloured Background	CTCB	
Coloured Texts with Random Lines	CTRL	
Text Distortion No Background	TDNB	
Text Distortion with Background Noise	TDBN	
Text Distortion with Coloured Background	TDCB	
Text Distortion with Random Lines	TDRL	

The twenty possible CAPTCHA code types, from the backgrounds and character features, are presented in Table 2. The image, full names and acronyms adopted for each CAPTCHA code type allows a visual comparison of the expected outputs for each code generation type selected in Step 4 of Algorithm 1.

Table 3 : Design Comparison of Accented CAPTCHA Codes

CAPTCHA Types	Distortion	Collapse	Fragmentation	Background	lines	Colour
CCNB	Yes	Yes	Nil	Nil	Nil	Yes
CCBN	Yes	Yes	Nil	Yes	Nil	Yes
CCCB	Yes	Yes	Nil	Yes	Nil	Yes
CCRL	Yes	Yes	Nil	Yes	Yes	Yes
CFNB	Yes	Yes	Yes	Nil	Nil	Yes
CFBN	Yes	Yes	Yes	Yes	Nil	Yes
CFCB	Yes	Yes	Yes	Yes	Nil	Yes
CFRL	Yes	Yes	Yes	Yes	Yes	Yes
CTNB	Nil	Nil	Nil	Nil	Nil	Yes
CTBN	Nil	Nil	Nil	Yes	Nil	Yes
CTCB	Nil	Nil	Nil	Yes	Nil	Yes
CTRL	Nil	Nil	Nil	Yes	Yes	Yes
TDNB	Yes	Yes	Nil	Nil	Nil	Yes
TDBN	Yes	Yes	Nil	Yes	Nil	Yes
TDCB	Yes	Yes	Nil	Yes	Nil	Yes
TDRL	Yes	Yes	Nil	Yes	Yes	Yes

A comparison of the features from Table 1 used in the CAPTCHA code types generated is shown in Table 3. The presence of a feature is indicated by 'Yes' while the absence is indicated by 'Nil.' The table indicates that at least the presence of 1 feature is sufficient to create a unique CAPTCHA code type as shown in CTNB or TDRL.

V. DISCUSSION

Both Linux and Windows operating systems support the created NaijaCAPTCHA system. Different web browsers can read and use the NaijaCAPTCHA source code. The keyboard is one of the most important input devices for entering data into a computer system. During system implementation, a software keyboard was used. Despite not being an actual keyboard, it allows users to type using simulated keys, rather than being hardware, the virtual keyboard is software. However, Naija CAPTCHA's inclusion of a non-Latin keyboard made it simpler for users to respond to the system's challenge. Non-essential keys were reassigned to accented characters, but the keyboard was left undisturbed.

Numerous CAPTCHA types were created as a result of Naija CAPTCHA's development using two components, namely the background and generated characters, as

shown in Table 1. Obfuscator method was used as a software protection technique to corrupt and warp the generated character codes, making the generated CAPTCHA code challenging for automated programmes to understand. Typically, complex and ambiguous wording using multiple backgrounds were used.

Twenty (20) CAPTCHA types were created by combining the different character forms and backgrounds in Table 1 as indicated in Table 2. Visual analysis of the generated code indicated that some of the code types (TNB/CTNB, TRL/CTRL, CTCB/TCB, and CTBN/CBN) shared similarities, indicating that the same type of code was replicated using a different combination. There were sixteen (16) different CAPTCHA types in all, without any repetition.

The created CAPTCHA types are assigned labels based on the main features that were incorporated during the obfuscation module. The various CAPTCHA types are listed in Table 3 along with a description of the properties each kind employs. Numerous CAPTCHA types would be implied by the usage of different features; however, the created CAPTCHA shares the same properties as another CAPTCHA type. "Yes" signifies the presence of a feature, whereas "Nil" indicates its absence. The table shows that at least 1 feature must be present in order to generate a special CAPTCHA code type, as displayed in CTNB or TDRL.

VI. CONCLUSION

This work enhanced the existing design offered by text-based CAPTCHA with the inclusion of accented characters from two Nigerian languages. The generated CAPTCHA code included both Latin and accented characters. Twenty (20) types of CAPTCHAs were possible with accented characters and security features included in the design. The designed CAPTCHA system, NaijaCAPTCHA, used a mix of obfuscation methods such as distortion, fragmentation, lines, colour and background to generate unique CAPTCHA code types. In future work, security and usability of NaijaCAPTCHA will be tested using standard tests.

VI. REFERENCES

- Alsuhibany, S. & Alnoshan, A. (2021). Interactive Handwritten and Text-Based Handwritten Arabic CAPTCHA Scheme for Mobile Device. *Journal of IEEE Access*, License by Comparative Study Creative Commons Attribution, 9, 140991-141001.
<https://creativecommons.org/licenses/by/4.0/>
- Arain, R. H., Shaikh, R. A., Kumar, K., Maitlo, A., Kehar, A., Shah, S. A. & Shiakh, H. (2018). Verifying the Robustness of Text-based CAPTCHAs offered by Local E-Commerce Sites. *IJCSNS International Journal of Computer Science and Network Security*, 18(9), 79-84.
- Conti, M., Pajola, I. & Tricomi, P. P. (2022). CAPTCHA Attacks: Turning CAPTCHA against Human. *Cited from arXiv:2201.04014v2[cs.CR]*
- Doi, M. & Lei, H. (2014). STARS: Word processing for the Japanese language. *Proceedings of the IEEE*, 102(2), 222–228.
- Guerar, M., Merlo, A. & Migliardi, M. (2018). Completely Automated Public Physical test to tell Computers and Humans Apart: A usability study on mobile devices. *In Future Generation Computer Systems*, 8, 617-630.
- Henri. (2018). Images on the History of CAPTCHA. *Angewandte Chemie International Edition*, 6(11), 951–952, 617–630.

- <https://bing.com/search?q=obfuscation>. Conversation with Bing, 05/07/2023
- Kumar, M., Jindal, M. K., & Kumar, M. Design of innovative CAPTCHA for Hindi language. (2021). Article in Neural Computing and Applications. <https://doi.org/10.1007/s00521-021-06686-0>
- Li, C., Chen, X., Wang, H., Zhang, Y. & Wang, P. (2020). An End-to-End Attacks on Text-Based CAPTCHAs Based on Cycle-Consistent Generative Adversarial Network. *Cited from arXiv:2008.11603v1[cs.CV]*.
- Li, Y., Kang, F., Shu, H., Xiong, X., Sha, Z. and Sui, Z. (2022). COOPS: A Code Obfuscation Method Based on Obscuring Program Semantics. *Jornual of Hindawi Security and Communication Networks*, Article ID 6903370, 1-15. <https://doi.org/10.1155/2022/6903370>
- Maitlo, A, Shaikh, R. A, Nawaz, H., Soomro, A. H., Mangi, A. & Soomro, I. (2021). CAPTCHA Design: A Novel Security Method using Sindhi Language. *International Journal of Advance in Computer Science and Engineer*, 10(3), 2145-2149.
- Merriam-Webster. (2021). *Character*. Available at. <https://www.merriam-webster.com/dictionary/character>
- OBFUSCATION | English meaning - Cambridge Dictionary. <https://dictionary.cambridge.org/dictionary/english/obfuscation>.
- Obimbo, C. Halligan, A. & De Freitas, P. (2013). CaptchAll: An Improvement on the Modern Text-Based CAPTCHA. *Procedia Computer Science*, 20, 496–501.
- Ogiela, M. R., Krzyworzeka, N. & Ogiela, L. (2018). Application of knowledge-based cognitive CAPTCHA in Cloud of Things security. *Concurrency and Computation: Practice and Experience*, 30(21). <https://doi.org/10.1002/cpe.4769>
- Olanrewaju, O. T & Osunade, O. (2017). Design of Accented Character-based CAPTCHA with Usability Test for Online Transactions. *Zambia Information Communication Technology (ICT) Journal*, 1(1), 20-24.
- Shivani, R. C. 2020. CAPTCHA : A Systematic Review. *IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, 1. DOI: [10.1109/ICATMRI51801.2020.9398494](https://doi.org/10.1109/ICATMRI51801.2020.9398494)
- Shi, C., Ji, S., Liu, Q., Liu, C., Chen, Y., He, Y., ... Wang, T. (2020). *Text Captcha Is Dead? A Large-Scale Deployment and Empirical Study*. *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. doi:10.1145/3372297.3417258
- Shivani, C. & Krishna, R. (2021). Spiral with Adversarial Perturbation and Its Security Analysis with Convolutional Neural Network. 1-8. <http://link.springer.com/chapter/10.1007/978-981-16-3346-1>
- Uma, P. Siddivinayak, K. & Ramachandra, P. (2019). Smart CAPTCHA to provide High Security against Bots. *In Proceeding of World Congress on Engineering, London*, ISBN: 978-988-14048-6-2 ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)
- Tangmanee, C. (2018). *Lowercase Letters in Text-Based CAPTCHA: A Visual Perception Analysis*. *2018 10th International Conference on Knowledge and Smart Technology (KST)*. doi:10.1109/kst.2018.8426163
- Tharad, A., Bhatt, A., Srivastava, S. & Kumar, P. (2018). Analysis Impact of Current Captcha Approaches and its Significance. *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical System*, 487–493. DOI: [10.1109/CTEMS.2018.8769168](https://doi.org/10.1109/CTEMS.2018.8769168)