DEVACAPTCHA - A FRAMEWORK TO PREVENT BOT ATTACKS

Sushma Yalamanchili [1] and Kameswara Rao[2]

[1]Research Scholar, Department of Computer Science & Engineering
Acharya Nagarjuna University, Andhra Pradesh, India.
sushma_yalamanchili@yahoo.co.in
[2]Department of Computer Science, P.G.Centre, P.B.Siddhartha College, Vijayawada.
Kamesh.manchiraju@gmail.com

ABSTRACT

Human Interactive Proofs (HIPs) are automatic reverse Turing tests designed to distinguish between various groups of users. Completely Automatic Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a HIP system that distinguish between humans and malicious computer programs. Many CAPTCHAs have been proposed in the literature that text-graphical based, audio-based, puzzle-based and mathematical questions-based. The design and implementation of CAPTCHAs fall in the realm of Artificial Intelligence. We aim to utilize CAPTCHAs as a tool to improve the security of Internet based applications. In this paper we present a framework for a text-based CAPTCHA based on Devanagari script which can exploit the difference in the reading proficiency between humans and computer programs. Our selection of Devanagari script-based CAPTCHA is based on the fact that it is used by a large number of Indian languages including Hindi which is the third most spoken language. There is potential for an exponential rise in the applications that are likely to be developed in that script thereby making it easy to secure Indian language based applications.

KEYWORDS

CAPTCHA, Devanagari, Human Interactive Proof, Optical Character Recognition, text-based.

1. INTRODUCTION

Human Interactive Proofs (HIPs) [1] focus on automation tests that virtually all humans can pass but current computer programs fail [2]. Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) was an acronym that was coined in 2000. It is a type of challenge-response test that only a human completes successfully. In the simplest form of a CAPTCHA, an image consisting of digits and letters sufficiently distorted is presented and the user is required to input the characters that are displayed. Other forms of CAPTCHAs are based on text-graphics, audio, hand-writing and puzzles. CAPTCHAs have been widely used as a security measure to restrict access from Robots or bots. CAPTCHAs are based on Artificial Intelligence (AI) problems that cannot be solved by current computer programs or bots but are easily solvable by humans. A client who provides a correct response to a challenge is presumed to be a human; otherwise a bot.

2. CURRENT RESEARCH

In this section, we review the current research on CAPTCHAs and Optical Character Recognition (OCR) efforts for the Devanagari script. Websites use CAPTCHAs as a security measurement to

D.C. Wyld, et al. (Eds): CCSEA 2011, CS & IT 02, pp. 272–279, 2011.
 DOI: 10.5121/csit.2011.1225

distinguish human users from bots. While CAPTCHAs have been developed based on pure text, images, audio and video, text CAPTCHAs are almost exclusively used in real applications. In a text CAPTCHA, characters are deliberately distorted and connected to prevent recognition by bots. Security of an existing text CAPTCHA is enhanced by systematically adding noise and distortion, and arranging characters more tightly [3, 4]. Usability is always an important issue in designing a CAPTCHA [5]. Examples of text-based CAPTCHAs include the Gimpy method [6], the Baffletext method [7], Handwritten CAPTCHA [8], the PayPal method [9], Using Dynamic Visual Patterns [10], the Hotmail Method [11] and Pessimal Print method [12]. Successful text CAPTCHAs used by Microsoft, Yahoo and Google use techniques that are resistant to segmentation [13, 14, 15] attacks by using random acrs, connected random lines and crowding characters.

Figure 1 shows an example of a HIP challenge presented when registering for a MSN Hotmail account. After displaying the HIP, the user is asked to type the eight characters included within it, namely, X29JTUN3.

Figure 1 Microsoft Hotmail Text CAPTCHA

A few of the many different HIP styles that can be produced by manipulating hardness parameters are illustrated below (Figure 2).

Figure 2 Hardware Parameter variation in HIPs

Image-based CAPTCHAs [16] require users to identify labeled images or rotated images. They evince a larger gap between human users and bots because of the poor ability of bots in obtaining features of images. In Image Recognition CAPTCHAs, the user is provided with a small set of images to name or distinguish or identify anomalies. In implicit CAPTCHAs [17], the user does not have to read or type anything and just makes simple clicks on hot spots. Drawing captcha [18] generates numerous dots on a screen with noisy background, some of which are diverse from the others. The user has to connect them to each other. Apart from the visual-CAPTCHAs, there exist a number of audio CAPTCHAs [19, 20, 21] where the user must recognize and type the word that is played as a sound. Sound-based CAPTCHAs exploit a border range of human abilities which are mainly based on the auditory perception of human ability to identify words or letters in a sound clip after being distorted and with additive background noise. Their emergence has greatly aided vision impaired people. A typical sound-based CAPTCHA is reCAPTCHA [22] proposed by Carnegie Mellon University and later acquired by Google. Video-based CAPTCHAs also substitute a brief video display of characters for the usual letters. In video-based CAPTCHAs [23], users will be prompted to view a challenge video and then appropriately annotate (or tag) it. The challenges will be graded by exact matching of user response with a database of ground truth

tags for the video. Multilingual CAPTCHAs [24, 25] are another class where the user's native language is selected and the messages are shown in that language. Researchers enjoyed moderate success as part of computer vision research in breaking existing text-based CAPTCHAs and image-based CAPTCHAs [26, 27, 28]. Image recognition is considered to be a much harder problem than text recognition. OCR is a process of breaking Image -based captchas.Recognition of characters is itself a challenging problem due to change of fonts or by introducing noise. Efforts have been made to improve the performance of OCR applied to Devanagari script. Still a lot of research is needed to tackle the challenges in identifying specific characters of Devanagari script. Many works on OCR efforts for Devanagari script have been reported [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, and 41].

3. MOTIVATION

Popular web sites are subject to brute force attacks by programs or computers known as Robots or Bots. They can be used to break user accounts or submit an unlimited number of service requests such as email account creation, web connection requests and serving as shopping agents. Such activities often lead to abuse of privilege causing the server to exhaust its resources or worse cause it to shut down. Bots can be eliminated by introducing CAPTCHA at service request time to prevent Denial of Service attacks. Introducing CAPTCHA in the authentication scheme prevents automated brute force attack.

We propose a framework for text-based CAPTCHA that is based on Devanagari script that is the written form for several Indian languages including Hindi, Sanskrit, Kashmiri, Bihari, Bhojpuri, Marati and Nepali. As the population that uses these languages is extremely large, we have selected the Devanagari script for use in this CAPTCHA. DevaCAPTCHA can be used to enhance the security of Devanagari script-based Indian language content based web applications. Some applications that are typical in this scenario are online and collaborative authoring, online tutoring, email and social networking portals in Indian languages. These users are typically involved in content generation and access of applications in Devanagari-script based Indian languages. They already have the necessary keyboard emulation or are using custom keyboards that facilitate keyboard input in that script so that aspect is taken care of.

4. DEVANAGARI SCRIPT

Devanagari is the script used by more than 400 million people on the globe. Devanagari has 11 vowels and 33 simple consonants. Besides the consonants and the vowels, other constituent symbols in Devanagari are set of vowel modifiers called *matra* (placed to the left, right, above, or at the bottom of a character or *conjunct*), pure-consonant (also called half-letters) which when combined with other consonants yield conjuncts. A horizontal line called *shirorekha* runs through the entire span of work. Devanagari word is written into the three strips namely: a core strip, a top strip, and a bottom strip as shown in Figure 3. The core strip and top strip are differentiated by the header, while the lower modifier is attached to the core character.

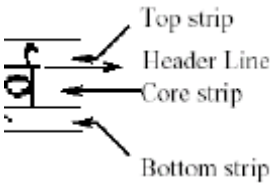

Figure 3 Character strips in Devanagari script

OCR for Devanagari script becomes even more difficult when compound character and modifier characteristics are combined in 'noisy' situations. Some illustrations of Devanagari Script are given in Figure 4.

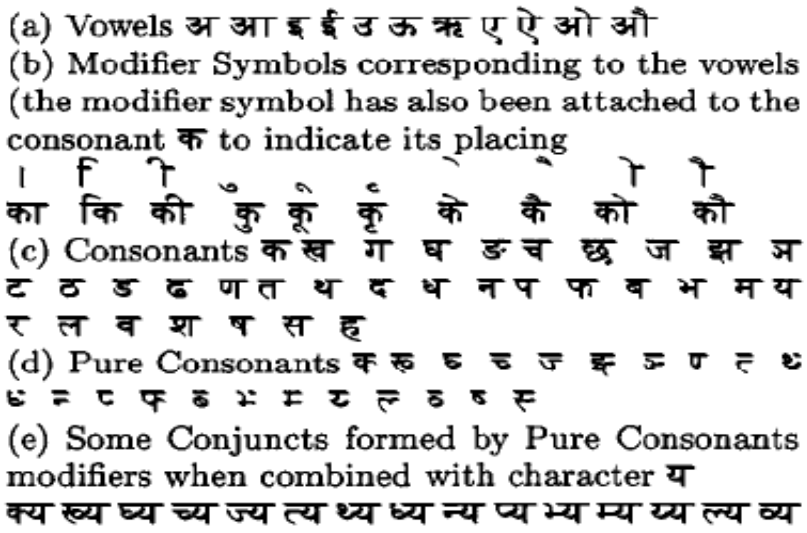

Figure 4 Devanagari script character set

5. PROPOSED FRAMEWORK

DevaCAPTCHA is a framework for administering a Devanagari script-based text CAPTCHA to assist in securing Indian-language web-based applications. The key components of the proposed DevaCAPTCHA are

1. DevaDB: A sufficiently large database of devanagari script samples (either in text or handwritten form)
2. Query Generator: A mechanism to query the database and obtain a random sample subject to the design
3. Obfuscator: A module that takes the random sample from the database that distorts and adds noise to it
4. DevaGUI: User challenge-response interface
5. Match Response: A determination of whether the user has submitted the accurate response for the challenge posed.

5.1 DevaDB

DevaDB is a collection of pages, phrases or words of words of Devanagari characters. Work is already in progress partially funded by the Government of India and other agencies that allow for generating of documents in Hindi that are the result of OCR techniques applied to scanned images of large repositories of old newspapers and books. These documents could be fed into a database from which it is possible to retrieve sentences, phrases, specified length words or a text string consisting of randomly arranged characters. The only requirement is for such a database to be online. Books for which the copyright has expired can also be used for such a purpose. A web crawler module can be planned to populate the database.

5.2 Query Generator

The Query Generator module simply generates a text string that will be pipelined into the Obfuscator module. The minimum and maximum length of the word, whether we select an existing word from the database or whether we generate a random assortment of characters are design choices or can be user specified in this module without compromising the security of the Turing test.

5.3 Obfuscator

The Obfuscator module is the crux of DevaCAPTCHA and uses specific characteristics of Devanagari script to deceive the adversary. In Devanagari script, all the individual characters are joined by a head line called "Shiro Rekha". The obfuscator may remove the headline and adds noise to the image using patterns like mosaic, arcs/jaws, vertically overlapped on the script. The Obfuscator further misleads machine by using different fonts of different sizes with variable character spacing. Sample images that are intended to be used in DevaCAPTCHA are shown below (Figure 5). Segmentation resistant methods will be employed in its design.

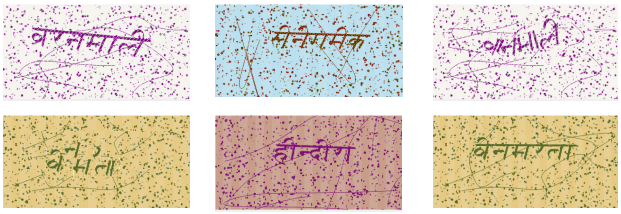

Figure 5 Visualized DevaCAPTCHA string samples

5.4 DevaGUI

In the DevaCAPTCHA, we display the substantially noisy and distorted image containing the chosen text string or word in the image display area. A text input box is provided where the user can type in the characters in sequence as they appear in the distorted image. A submit button is to be pressed to signal completion of input.

5.5 Match Response

The typed in text should match the pre-obfuscated string generated by the DevaDB query. If it does, the user is indeed a human and may be permitted to authenticate or request for web services.

6. CONCLUSION

We have seen that by distinguishing between humans and computers, CAPTCHAs offer protection against automated attacks on systems and applications. The criteria for success of a text-based CAPTCHA is its robustness. We have outlined techniques in the Obfuscator module to generate the random images that can be used to beat OCR technology. DevaCAPTCHA is thus robust as it is resistant to bot attacks through the use of OCR technology in deciphering the displayed string.

Another parameter to assess the worthiness of a CAPTCHA is usability or how easy it is for humans to pass the test. Since we are using words from books and newspapers or an assortment of characters that are non-words, it is not difficult for humans to visually perceive these characters despite the distortions and noise due to their superior visual capabilities and cognitive abilities to make connections with words that they have encountered in some context. Distorted images containing random strings are still easy for humans to read while computers spend endless time processing information. Thus usability is assured while the generated random image is difficult for bots to decipher. It is to avoid dictionary attacks that a design option of using a random string of text rather than words can be made. The implementation of DevaCAPTCHA and the participation in OCR testing efforts related to Indian language scripts is to be taken up as future research work.

REFERENCES

[1] Shirali-Shahreza1, S. & Shirali-Shahrezal, M., (2010) A Survey Of Human Interactive Proof Systems, International Journal of Innovative Computing, Information and Control, Volume 6, Number 3(A), March.

[2] Ahn, L. von, Blum, M., & Langford, J., (2003) "Telling humans and computers apart automatically", Communications of the ACM, vol 46, August, pp 57-60.

[3] Ahn, L. von, Blum, M., Hopper, N. J., & Langford, J., (2003, "CAPTCHA: Using hard AI problems for security", Proceedings of Eurocrypt 2003.

[4] Baird, H. S. & Popat, K., (2002) "Human interactive proofs and document image analysis", Proceedings of Document Analysis Systems 2002, pp 507–518.

[5] Yan, J. & El Ahmad, A. S., (2008) "Usability of CAPTCHAs or usability issue in CAPTCHA design", Proceedings of 4th Symposium on Usable Privacy and Security (2008), pp 44-52.

[6] Ahn, L. v., Blum M. & Langford, J., (2000) "The CAPTCHA Project (Completely Automatic Public Turing Test to tell Computers and Humans Apart)", Project at Department of Computer Science, Carnegie-Mellon University, http://www.captcha.net.

[7] Chew, M. & Baird, H. S., (2003) "BaffleText: a Human Interactive Proof.", Proceedings of 10th SPIE/IS&T Document Recognition and Retrieval Conference (DRR2003), Santa Clara, California, USA, pp 305-316.

[8] Rusu, A. & Govindaraju, V., (2004) "Handwritten CAPTCHA: using the difference in the abilities of humans and machines in reading handwritten words", Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Tokyo, Japan, pp. 226-231.

[9] PayPal (2006), PayPal registration. Website https://www.paypal.com.

[10] Liao, W.H. & Chang C., (2004) "Embedding information within dynamic visual patterns", Proceedings of the IEEE International Conference on Multimedia and Expo, 2004. (ICME 04), Taipei, Taiwan, vol. 2, pp 895-898.

[11] Microsoft Hotmail, Website http://www.hotmail.com.

[12] Coates, A.L., Baird, H.S. & Fateman, R. (2001) "Pessimal Print: a Reverse Turing Test," Proceedings of IAPR 6th International Conference. on Document Analysis and Recognition, pp 1154-1158, Seattle, Washington, USA, September 10-13.

[13] Baird, H.S. & Riopka, T., (2005) "ScatterType: a Reading CAPTCHA Resistant to Segmentation Attack", Proceedings of IS&T/SPIE Document Recognition & Retrieval XII Conference, San Jose, California, USA, January 16-20.

[14] Ahmad, A.S.E., Yan, J. & Marshall, L., (2010) "The robustness of a new CAPTCHA", Proceedings of EUROSEC 2010, Paris, France, ACM 978-1-4503-0059-9/10/0004.

[15] Baird, H.S., Moll, M.A. & Wang, S.Y., (2005) "A highly legible CAPTCHA that resists segmentation attacks", Proceedings of Second International Workshop on Human Interactive Proofs (HIP 2005), ed. By H.S.Baird and D.P.Lopresti, Springer Verlag, LNCS 3517, Bethlehem, Pennsylvania, USA 2005.

[16] Merler, M. & Jacob, J., (2009) "Breaking an Image based CAPTCHA", Technical Paper submitted to the Department of Computer Science, Columbia University, USA, Spring term, Available at www.cs.columbia.edu/~mmerler/project/FinalReport.pdf.

[17] Baird, H.S. & Bentley, J.L., (2005) "Implicit CAPTCHAs", Proceedings of the SPIE/IS&T Conference on Document Recognition and Retrieval XII (DR&R2005), San Jose, pp. 191-196.

[18] Shirali-Shahreza, M. & Shirali-Shahreza, S., (2006) "Drawing CAPTCHA", Proceedings of the 28th International Conference Information Technology Interfaces (ITI 2006), Cavtat, Dubrovnik, Croatia, June 19-22, 2006, pp 475-480.

[19] Kochanski, G. et al., (2002) "A Reverse Turing Test using speech", Proceedings of the Seventh International Conference on Spoken Language Processing (ICSLP2002 -INTERSPEECH 2002), Denver, Colorado, USA, September 16-20, 2002, pp 1357-1360

[20] Chan., T.Y. (2003) "Using a Text-to-Speech Synthesizer to Generate a Reverse Turing Test," Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp 226-232.

[21] Schlaikjer, A., (2007) "A Dual-Use Speech CAPTCHA: Aiding Visually Impaired Web Users while Providing Transcriptions of Audio Streams", Technical Report CMU-LTI-07-014, Carnegie Mellon University, November.

[22] Ahn, L.v., Maurer, B., McMillen, C., Abrahama, D. & Blum, M., (2008) "reCAPTCHA: Human-Based Character Recognition via Web Security Measures", Science, vol 321, 12 September, Available at http://www.cs.cmu.edu/~biglou/reCAPTCHA_Science.pdf.

[23] Kluever, K.A., (2008) "Evaluating the Usability and Security of a Video CAPTCHA", Master's thesis submitted to Rochester Institute of Technology, Rochester, New York, August.

[24] Shirali-Shahreza, M.H. & Shirali-Shahreza, M., (2007) "Multilingual CAPTCHA", ICCC 2007 IEEE International Conference on Computational Cybernetics, 19–21 October, Gammarth, Tunisiali.

[25] Shirali-Shahreza, M.H. & Shirali-Shahreza, M., (2006) "Persian/Arabic Baffletext CAPTCHA," Journal of Universal Computer Science , vol. 12, no. 12, pp 1783-1796, December.

[26] Chellapilla, K. & Simard, P.Y., (2004) "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)", Proceedings of NIPS.

[27] Mori, G. & Malik, J. (2003) "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," Proceedings of IEEE CVPR, 2003.

[28] Moy, G., Jones, N., Harkless, C. & Potter, R., (2004) "Distortion Estimation Techniques in Solving Visual CAPTCHAs," Proc. IEEE CVPR, 2004

[29] Bansal, V. & Sinha, R.M.K., (1999) "Partitioning and Searching Dictionary for Correction of Optically-Read Devanagari Character Strings", Proceedings of Fifth International Conference on Document Analysis and Recognition, IEEE Publication, Bangalore, September 21-23, pp 410-413.

[30] Sinha, R.M.K., (1987) "Rule based contextual post processing for Devanagari text recognition", Pattern Recognition, vol 20, num 5, pp 475-485.

[31] Sethi, I.K., (1977) "Machine recognition of constrained hand printed Devanagari", Pattern Recognition, vol. 9, pp 69-75.

[32] Bansal, V. & Sinha, R.M.K (2002) "Segmentation of Touching and Fused Devanagari Characters", Pattern Recognition, vol. 35, pp. 875-893, April.

[33] Bansal, V. & Sinha, R.M.K, (2001) "A Devanagari OCR and A Brief Overview of OCR for Indian Script", Proceedings of Symposium on Transaction support System (STRANS 2001), February 15-17, Kanpur, India.

[34] Arora, S., Bhattacharjee, D., Nasipuri, M. & Malik, L., (2007) "A two stage classification approach for hand-written devanagari characters", International Conference on Computational Intelligence and Multimedia Application (ICCIMA07), Sivkasi, Tamil Nadu, India.

[35] Agrawal, M. & Doermann, D., (2008) "Re-Targetable OCR with Intelligent Character Segmentation", DAS '08: Proceedings of the Eighth IAPR International Workshop on Document Analysis Systems, pp 183-190, September.

[36] Holambe, A.N. & Thool, R.C., (2010) "Printed and Handwritten Character & Number Recognition of Devanagari Script using SVM and KNN", International Journal of Recent Trends in Engineering and Technology, vol. 3, No. 2, May.

[37] Kompalli, S., Setlur, S. & Govindaraju, V., (2006) "Design and comparison of segmentation driven and recognition driven devanagari ocr", Proceedings of DIAL, pp 96–102, 2006.

[38] Bansal, V. & Sinha, R.M.K, (2001) "A Complete OCR for Printed Hindi Text in Devnagari Script", Sixth International Conference on Document Analysis and Recognition, IEEE Publication, Seatle USA, 2001, pp 800-804.

[39] Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D.K. & Kundu, M., (2008) "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, IIT Kharagpur, India Dec 2008.

[40] Arora, S., Malik, L. & Bhattacharjee, D., (2006) "A Novel Approach For Handwritten Devnagari Recognition" in IEEE –International Conference on Signal And Image Processing, Hubli, Karnataka, Dec 7-9, 2006.

[41] Bansal, V. & Sinha, R., (2000) "Integrating knowledge sources in Devanagari text recognition", IEEE Transactions on System, Man and Cybernetics, vol 30, num 4, pp 500–505.

[42] Bansal, V. & Sinha, R.M.K., (2002) "Segmentation of Touching and Fused Devanagari Characters", Pattern Recognition, vol. 35, pp 875-893, April 2002.

[43] Garain, U. & Chaudhuri, B.B., (2001) "Segmentation of Touching Characters in Printed Devnagari and Bangla Scripts using Fuzzy Multifactorial Analysis", Proceedings of 6th ICDAR, pp 805-809, 10-13 September 2001.

[44] Marwah, S.S., Mullick, S.K. & Sinha, R.M.K., (1994) "Recognition of Devanagari characters using a hierarchical binary decision tree classifier", IEEE International Conference on Systems, Man and Cybernetics, October 1994.

Authors

Sushma Yalamanchili completed her under-graduate studies in Computer Science at BITS, Pilani, India and graduate studies at Michigan State University, USA. She has been teaching graduate students since 1998. Her areas of interest are Network & Information Security, Human Interaction Proof, Pattern Recognition and Image Processing.

M. Kameswara Rao has MCA from Acharya Nagarjuna University and M.Phil from Bharatidasan University. He is currently pursuing M.Tech in Acharya Nagarjuna University. He has taught graduate students in Computer Science for ten years. His areas of interest are Network Security, Pattern Recognition and Image Processing.