I.J. Information Technology and Computer Science, **2012, 9, 53-58**
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.09.07

Random Handwritten CAPTCHA: Web Security with a Difference

Mukta Rao
Gurukul Kangri Vishwavidyalaya, Haridwar, India
mukta.mca@gmail.com

Nipur Singh
Gurukul Kangri Vishwavidyalaya, Haridwar, India
nipursingh@hotmail.com

***Abstract*—** It is hard to believe a web form without a CAPTCHA. The web survival in this cut-throat competition is impossible without the mechanisms for blocking spam-boats. The spam-boats have now been made intelligent enough to break through machine printed CAPTCHAs. Handwritten CAPTCHA image can be one solution. In this paper handwritten CAPTCHA images have been used to enhance the web security. Introduction of randomness at various stages is proven to increase the recognition complexity for the spam boats, whereas the ease of recognition of handwritten words by human beings eventually increases the usefulness of such CAPTCHA. The technique used to produce colored image of handwritten letters also has randomness associated with it. The proposed CAPTCHA images contain alphanumeric content, one word with letters and a number with handwritten numerals. CAPTCHA images developed using proposed technique have been tested across various OCRs and online available resources as well..

***Index Terms*—** CAPTCHA, Handwritten CAPTCHA, Random CAPTCHA, Web Form Securit

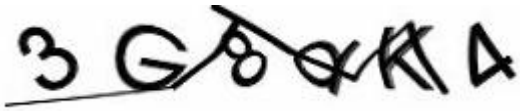
I. Introduction

The necessity of having a CAPTCHA on a web form is now not a topic of debate. It is a must have test to keep the website up and live. Since Alan Turing's first test [1] this field has found no bars. Spammers and CAPTCHA techniques are progressing hand in hand. Wide varieties of CAPTCHA systems are available on internet along with their implementation steps [2-3]. This gives a chance to spammers to crack them. Few examples collected from online resources along with their relative analysis are presented in table-1.

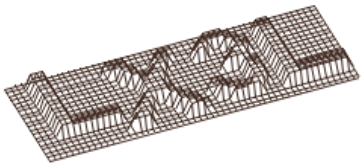
Table 1: Samples of CAPTCHA collected from internet with their complexity analysis.

Google: - The nonlinear distortion of the inscriptions, the offset relative to each other's character, proximity characters, different fonts, noise shall not apply. However, the characters do not always stick together without gaps.

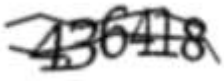
MSN: - Rotating and distortion of characters, the noise in the form of lines of the same color crossing characters.



Yahoo: - The nonlinear distortions of symbols, the noise in the form of intersecting characters are broken. The broken line can be separated from the characters.



rediffmail :- Rotate character, a small variation in fonts, low contrast with the background. In my opinion – a very good CAPTCHA (the characters are 3 dimensionally projectetd)

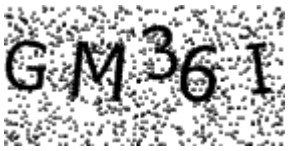


Mail.com old: - offset adjustment, the nonlinear distortion of the characters, the noise in the form of intersecting characters are broken in the same color.

Yandex :- Gluing outline symbols, harmonic distortion, noise in the form of white and dark lines.

IEEE Contact us:- The nonlinear distortion of characters. The characters, however, easily separated from each other and from the noise. It is only for character recognition, and it is not difficult even without the involvement of neural networks (given that the characters are distorted figures and mostly horizontal).

Beeline :- Small linear distortions of characters of different sizes. The symbols are placed on the background noise in the form of geometric shapes (ellipses) by inversion. Disadvantage: The figures contrast the background.



MTS old :- A small rotation, the shift character. Occasionally napolzanie them to each other. Noise reduction can be easily removed sharpness (blur). Pretty weak CAPTCHA

MTS New:- A small rotation, the shift character. Noise in the form of background textures and lines superimposed on the symbols. However, sometimes even a person is not always easy to read results.

Megaphone :- Offset symbols, the noise in the form of lines and dots. Parallel lines can be filtered, with sloping complicated.

Skylink :- Some sort of parody of the CAPTCHA. Constant font, unchanging arrangement of symbols. Can be easily read by the superposition of the mask.

Object recognition CAPTCHA'S have gained much of interest of researches [4, 20] so as moving CAPTCHA [5, 17]. They have been proven to be secured but their correct recognition by genuine end user takes more time as compared to text based CAPTCHA. Another important aspect related to it is the vast / wide spread usage a familiarity of text based CAPTCHA. It has become so popular cum – obvious that a regular user of internet services even does not read out the instruction before filling the CAPTCHA Test box. Thus, the text based CAPTCHA s are of our interest of research in this paper.

Ever since the invention of OCRs, human offline hand written text recognition has been an unsolved challenge for scientists the handwritten text is easy to read by human users but difficult for spammers / OCRs. The 100% offline handwriting recognition is still a challenge for OCRs. Thus use of handwritten text as CAPTCHA image is a revolutionary idea. It has seen many phases and changes related to distortion/colors/noise and familiarity aspects. It is known that text that is familiar in human readings perform better than on unfamiliar text in recognition process by human beings [19]. The difference in abilities of human and machines have led to use handwritten text in CAPTCHA images [6]. There has been successfully broken set of CAPTCHA even provided by Microsoft Word 2007, Google and yahoo [11]. The main reason behind this is the increasing technology and excellent success ratio of OCR against printed text [12, 21].

The work proposed by V. Govind Raju [6] is the first step in this direction but the kind of complexity they have introduced makes the CAPTCHA even non-understandable for most of the users. The trade off between readability and security should be perfectly balanced. The introduced of noise of 5 different arbitrary types have though increased the security [6**]** but at the same time the readability has decreased resulting in less usability [15]. Many users find text based CAPTCHA frustrating and break rates as high as 60% have been reported (for Microsoft's Hotmail) [13, 14].

Use of numerals along with English letters have also been proven to increase the complexity for automated text recognition [7] In this paper we have tried to extend the work in [7] by introducing randomness at various stages. Naming a few of them are selecting the number of letters and number of digits for numeric string, random order of appearance of string and numerals in the CAPTCHA image, color code selection for text/digits and the relative position of content is also decided at run-time using system generated random values/functions.

The authors have done adequate analysis and literature survey for the selection of random numbers generating functions, their run-time complexity, their memory requirements and usability. The sample collection for training and testing of ANN is also done using taking cautious steps and the possibility of correct segmentation of poorly written text is very low.

The paper is structured using 4 sections. Section-1 that has already reached to an end at this stage is and

introduction about the problem statement and the brief literature survey. Followed by- section 2 that describes the technical implementation of the process of CAPTCHA generation. Section 3 covers the result analysis of tests conducted over 5 different optical character recognition system available online and their discussion. The last section represents the conclusion and future work. The paper ends with bibliography/references.

II. Technical Implementation

This section details every minute detail of proposed handwritten CAPTCHA image generation. The core idea / consideration is to develop the CAPTCHA images which are easy to read by human users but very difficult for OCRs/spambots, hence a wide variety of available CAPTCHA techniques and samples have been studied. The implementation of handwritten CAPTCHA generation is based on the following observations [7]:

- All CAPTCHA images the authors have seen are made up of letters or numbers but not both.
- Maximum CAPTCHA images consist of a single word as also shown in examples in section 1 and in most cases same word length.
- If multiple words [eg- Facebook, gmail etc] then the words are in same horizontal line.
- The content of CAPTCHA image is not a word from dictionary, this practically consumes time of the user to type, thus becomes an annoying task. It's always easy and pleasant to type the words from dictionary rather than combination of arbitrary letters.

Based on these observations we have created CAPTCHA those have: -

- Numbers in the image.
- Image those contain two words.
- The words of CAPTCHA image will necessarily not be in same horizontal line, they can be placed in any order based on a random function result.
- The content of CAPTCHA shall be word from English dictionary and a number of 4-7 digits this would increase the recognition complexity but the friendliness shall be maintained.

The next subsections detail the different stages of handwritten CAPTCHA generation.

- SAMPLING AND PREPROCESSING.
- Clustering using ANN and noise introduction
- COMBINING to create the image.
- Entertaining freedom of user key strokes and authentication.

A. Sampling and Preprocessing

Since recognition of poorly written handwritten text is annoying for every end-user, we have collected the sample from 10 graduation institutes. The answer sheets of the students at university examination have been scanned and used for sampling in this paper.

Various successful noise removal techniques are available online. Using the techniques described in [8] we have removed the noise and segmented the word image into letters. Poor segmentation is discarded using HNN. The answer sheets of mathematics were scanned to get handwritten numerable and ideally there was no need for any post segmentation verification because handwritten numerals can be segmented with almost 100% success [9-10]. It is very easy to recognize and type word from dictionary rather than arbitrary set of letters. The random/arbitrary set of letters also consume more time. We have used words from Oxford University Press's English Dictionary. The words chosen are of length from 4-7. This subset of dictionary shall be used to generate the first word/alphabetic part of handwritten CAPTCHA image.

B. Clustering using ANN and noise introduction

Clustering technique of ANN is used to create 3 major clusters. Clustering is performed using a distance to group similar patterns. The attempt is to select those patterns which are the most representative for a given class. A selected pattern becomes a center of a cluster and our aim is to make as big clusters as possible. This way we keep the number of clusters fixed and small. The important aspect of this clustering is that care is taken not to include the alien patterns. If the cluster is too small, or if it can be included in another cluster larger than itself, then it is rejected [16].

As the training proceeds, we keep on selecting the cluster centers from the training set. We cover the entire region of the vector which includes the patterns of that particular class. The process is repeated till we exhaust all patterns for that class.

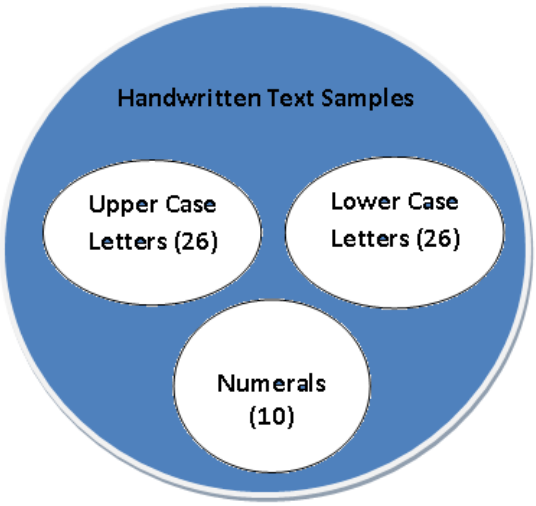


Fig.1: Division of Handwritten Text Samples using ANN

These major clusters are again divided into sub cluster to accommodate 26 letters of English alphabet set and 10 numeric digits.

The result of this process is a well arranged sample set that can be readily used for creation of CAPTCHA. The following algorithm is used in order to create a CAPTCHA images: -

C. Algorithm: Combining to Create The Image

1. Use a random function twice to get value out of {4, 5, 6, 7} resulting in size of CAPTCHA word.
2. Find a random number between 1- 10000 to get the index of dictionary word say **w1.**
3. Get the letters of **w1** and select the corresponding letters from clusters. Again use a random function to pick the letters from the sub clusters of the letter.
4. At server side maintain a cyclic queue of size 10 containing order codes, which would be used in following fashion as described in figure 2.

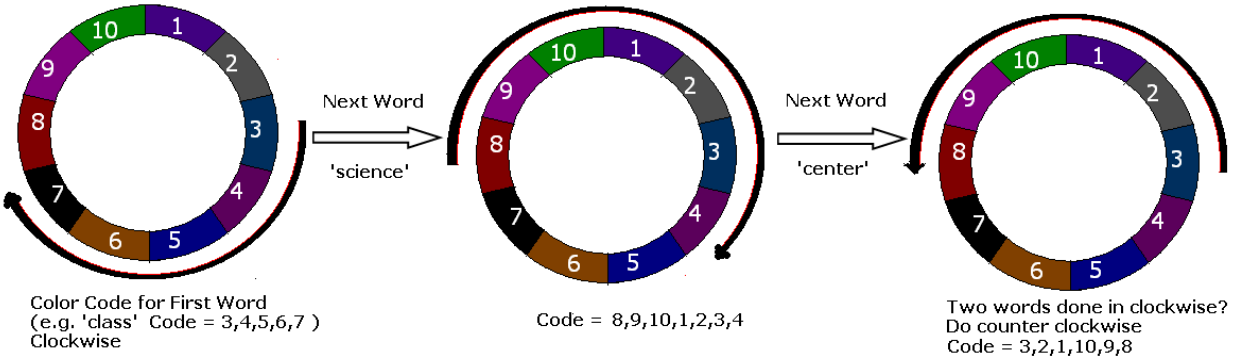


Fig.2: Color Code selection on run time for CAPTCHA image letters based on the length and clockwise/ anticlockwise arrangement for consecutive words

5. As described in Figure-2, find the length of word and then take color codes of the length from cyclic queue in clock-wise arrangement. Consider two words for clock-wise color codes and then after anti-clockwise for next two words. Thus the color codes shall be different for any 4 consecutive words at least.
6. Now the arrays of letters are colored images, ready to be grouped.
7. The outcome of step 1 determines the length of digits for CAPTCHA image. Based on this length we arbitrarily pick one number of the said length and the digits are colored as described in steps 3 and 4.
8. Use a binary value returning random function for horizontal or vertical stacking of the two words as well as their order of appearance.
9. Based on the outcome of the step-6 create CAPTHA.

Fig.3: Random Handwritten CAPTCHA image generated using algorithm mentioned above.

D. Freedom & Authentication

The user has been provided with the freedom of spaces. Between two words leading and trailing spaces can be easily ignored. The authentication of user's key strokes is done at server side that maintains a hash value of text of CAPTCHA. This makes the server authentication more secure. The hash value of user's key strokes is also captured. If both of them match then the end user is authenticated as a human being else spams bot.

III. Test Results and Discussion

Testing of any hypothesis/algorithm is a key step. It allows the user to check the correctness, usefulness and user-friendliness of the said technique. Since the proposed system contains randomness at various levels, we have tested the proposed technique/algorithm for various random/arbitrary functions. Since the implementation has been done using MATLAB, hence the used random functions were varied in nature and working style. The were normrnd(mu,sigma,m,n), randn(m,n), randi(imax,imin, n) and Multiply-with-carry method invented by George Marsaglia[18]. It is computationally fast and has good (albeit not cryptographically strong) randomness properties. The system has been developed in MATLAB 7.0 (A software solution for engineering works by Mathworks, Inc.). The OCRs used to test the recognition CAPTCHA are freely available online OCRs and they are www.onlineocr.net referred as OCR-1, www.free-ocr.com referred as OCR-2, www.free-online-ocr.com referred as OCR-3, www.newocr.com referred as OCR-4 and i2OCR at www.sciweavers.org/free-online-ocr referred as OCR-5 . The same set of test samples have been conducted on 128 test samples and every sample has been tested with 5 OCRs hence in total 128x5 tests have been carried out online. The correct and incorrect recognition ratio of these three OCRs is

provided in Table-2 in following content. The correct recognition is not of our interest as the CAPTCHA is meant for securing the websites, our aim is to minimize the correct recognition and maximize the incorrect recognition. The test results as shown in Table-1 depict the usefulness of the proposed CAPTCHA generation algorithm.

Table-2: The sample test results of Handwritten CAPTCHA images against various OCRs

	OCR-1	*OCR-2*	*OCR-3*	*OCR-4*	*OCR-5*
Correct Recognition	*163*	*89*	*106*	*71*	*112*
Incorrect Recognition	*477/* ***74.5%***	*551/* ***86.09%***	*134/* ***83.43%***	*569/* ***88.9%***	*528/* ***82.5%***

The table mentioned above shows the percentage of recognized CAPTCHA images for different OCRs. It is clearly evident that the overall success (Non Recognition of CAPTCHA) percentage is remarkable. The function Multiply-with-carry method produces better CAPTCHA images/words so that their recognition possibility is too low. It is very evident that introduction of color, numerals and random handwritten letters for CAPTCHA generation has got across good accomplishment rate for web security, at the same time usefulness and ease of recognition of text by human being is also gained benefit.

IV. Conclusion

Human Interactive Proofs or computer based challenges have now become synonyms for CAPTCHA. At current scenario of web usage and competition, text-CAPTCHA is still effective and secure measure, because there is no universal algorithm for text recognition that produces efficient results. In this paper we have presented an effective random algorithm to produce handwritten CAPTCHA which are relatively difficult for OCRs to recognize, while being very user/human friendly as made up of human handwritten letters. The usage of dictionary words has also increased the possibility of usability and friendliness.

The various weaknesses that were revealed in the literature study were examined and overcome in the algorithm. The Proposed CAPTCHA algorithm was well tested on the online server and was found to withstand the direction of the current OCR programs. Another advantage is that it is lightweight, it is safe and highly efficient. Although the test results indicate adequate success rates, still completely secured CAPTCHA image generation is an open challenge.

There are some open issues with current implementations of CAPTCHA which represent dimension for future advancements. The biggest concern is time taken to understand and type the words of CAPTCHA image and incase the incorrect recognition has taken place then the subsequent annoyance and time consumption of the user. To enhance security of web content, the authors are next working on introduction of sentences and phrases under Natural Language Processing with handwritten CAPTCHA. Considering the fact that user does not like to solve complex CAPTCHA and spammers are always there to crack simple CAPTCHAs, more ways should be discovered to make the web forms secure and spam-bot resistant.

References

[1] A.M. Turing, Computing machinery and intelligence, Mind 59 (236), 1950, pages 433–460.

[2] M. Blum, L. von Ahn, J. Langford, and N. Hopper. The captcha project: Completely automatic public turing test to tell computers and humans apart. http://www.captcha.net, November 2000.

[3] M. Chew and H. Baird. Baffletext: A human interactive proof. Proc. SPIE-IST Electronic Imaging, Document Recognition and Retrieval, pages 305–316, January 2003.

[4] Jing-Song Cui, Jing-Ting Mei, Xia Wang, Da Zhang, and Wu-Zhou Zhang. International Conference on E-Business and E-Government (ICEE), 2010. pp 1277 – 1280

[5] Jing-Song Cui, Jing-Ting Mei, Xia Wang, Da Zhang, and Wu-Zhou Zhang. 2009. A CAPTCHA Implementation Based on 3D Animation. In Proceedings of the 2009 International Conference on Multimedia Information Networking and Security - Volume 02 (MINES '09), Vol. 2. IEEE Computer Society, Washington, DC, USA, 179-182. 2009

[6] A. Rusu and V. Govindaraju, "Handwritten CAPTCHA: Using the Difference in the Abilities of Humans and Machines in Reading Handwritten Words". Proc. 9th IAPR International Workshop on Frontiers of Handwriting Recognition (IWFHR 2004), IEEE Computer Society, ISBN 0-7695-2187-8, pp. 226-231, 2004

[7] Mukta Rao, Nipur Singh, "Secure and Easy to Read Automated Handwritten CAPTCHA". IMS Manthan : The international Journal of Innovation, Vol 5, no-2, pp 47 - 56. 2011

[8] http://math.berkeley.edu/~sethian/2006/Applications/ImageProcessing/noiseremoval.html

[9] E. Vellasques , L. S. Oliveira , A. S. Britto, Jr. , A. L. Koerich , R. Sabourin, Filtering segmentation cuts for digit string recognition, Pattern Recognition, v.41 n.10, p.3044-3053, October, 2008

[10] Elnagar, A., and Alhajj, R. (2003). Segmentation of connected handwritten numeral strings. Pattern Recognition 36, 625-634, 2003

[11] http://www.zdnet.com/blog/security/gmail-yahoo-and-hotmails-captcha-broken-by-spammers/1418

[12] www.dlib.org/dlib/march09/holley/03holley.html and https://www.freelancer.com/users/851399.html and many more

[13] Kurt Alfred Kluever, Richard Zanibbi. "Video CAPTCHAs: Usability vs. Security" in Proceedings of the IEEE Western New York Image Processing Workshop (WNYIP '08), IEEE Press (2008)

[14] W3C Working Group, "Inaccessibility of CAPTCHA - Alternatives to Visual Turing Tests on the Web", Nov, 2005. Available at http://www.w3.org/TR/turingtest/.

[15] K Chellapilla, K Larson, P Simard and M Czerwinski, "Building Segmentation Based Human-friendly Human Interaction Proofs", 2nd Int'l Workshop on Human Interaction Proofs, Springer-Verlag, LNCS 3517, 2005.

[16] Starzyk, J.A.; Ansari, N.; , "Feedforward neural network for handwritten character recognition ," Circuits and Systems, 2002. ISCAS '02. Proceedings., 2002 IEEE International Symposium on , vol.16, no., pp.2884-2887 vol.16, 10-13 May 2002

[17] Y. Rui and Z. Liu. Artifacial: Automated reverse turing test using facial features. Proc. The 11th ACM international conference on Multimedia, November 2003.

[18] Marsaglia, George, 2003, Random number generators, Journal of Modern Applied Statistical Methods, 2 No. 2, 2003

[19] O. J. L. f and H. Singer. Perception of Print: Reading Research in Experimental Psychology. Lawrence Erlbaum Associates, Inc.

[20] D. Lopresti. Leveraging the CAPTCHA problem. In Proc. of the Second International Workshop on Human Interactive Proofs, pp. 97{110. Springer Verlag, 2005.

[21] J. Elson, J. Douceur, J. Howell and J. Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In Proc. of ACM CCS 2007, pp. 366-374. http://research.microsoft.com/asirra/

Mukta Rao: Research Scholar at Gurukul Kangri Vishwavidyalay, Haridwar, India in Computer Science. Her research area includes image processing, security and authentication using ANN.

Dr Nipur Singh: Professor of Computer Science in Gurukul Kangri University, Haridwar, India. A renowned personality and academician in the fields of Information security, mobile agents and network security.

