Enhancing Web-Security with Stronger Captchas

Dilip Kumar Kushwaha
Department of MCA
BBDIT, Ghaziabad, India

Harleen Kaur
FMIT, JAMIA HAMDARD
New Delhi, India

KM Priyanka
Department of MCA
BBDIT, Ghaziabad, India

Abstract:. Captcha are used widely over the World Wide Web to prevent automated programs in order to scrape a data from websites. Captcha is a challenge response test used to ensure that the response is generated by a person not by a computer. Users are asked to read and type a string of distorted characters in order to ensure that the user is human or not. Automation is real problem for web application. Automated attacks can exploit many services:

1. Blogs 2. Forums 3. Phishing 4. Theft of data

Registration Websites use CAPTCHA (completely automated public turing test to tell computers and human apart) systems to prevent the bot programs from wasting their resources. Today is the Era of where technologies are changes very rapidly. So spammers are hackers are also trying something new to cracking captcha. That’s why it is necessary to developing an advanced technology to generating a captcha. Just like simply generating a Captcha Images from text, or rotating an object within images.

Keywords: CAPTCHA, Spammers, Hackers, Bot, Images generation, Rotation of Object

1. INTRODUCTION:

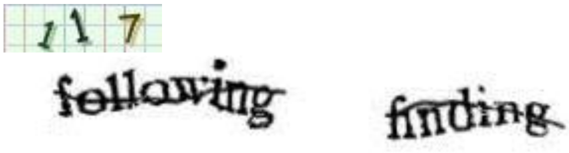
Completely Automated Public Turing Test To Tell Computers and Humans Apart.

Captcha (Completely automated public turing test to tell computers and human aparts) are used widely over the World Wide Web to prevent automated programs to in order to a data from websites.

A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a Challenge-response system test designed to differentiate humans from automated programs.

A CAPTCHA differentiates between human and bot by setting some task that is easy for most humans to perform but is more difficult and time-consuming for current bots to complete.

CAPTCHAs are often used to stop bots and other automated programs from using blogs to affect search engine rankings, signing up for e-mail accounts to send out spam or take part in on-line polls.



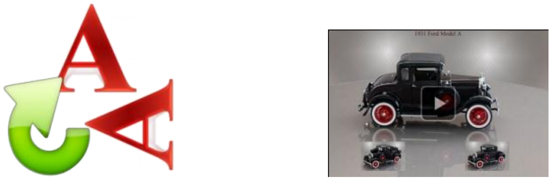


Figure 1.Some Captcha Images

There are many problems in generating a captcha just like simple captcha are easy to crack, difficult in reading a captcha. Sometimes Captcha are difficult to read (***for old age person, for different language person***). Suppose a person using a service to open an email account, that person does not know the English language but the captcha is generated in English language, so the person are unable to understand the captcha, means the captcha are generated in static manner. This is the problem associated with Captcha.

So I can solve this problem by generating Captcha in dynamic manner.

Generating captcha in dynamic manner means according to the situation likes a old age person using web services OR a person who knows different language. But sometimes the CAPTCHA test is difficult for human users to read because the users have different ages, languages, etc. In this paper we propose a solution for this problem. The registration system can select a kind of CAPTCHA system among different available CAPTCHA systems regarding to the user's needs such as user language and disabilities.

For example if a user is illiterate, then he/she easily detect images, so for this purpose I generate a images of text, or increasing the complexity of captcha by rotating objects within images.

2.Brief History:

Turing [1] in 1950 proposed a test for AI in which a computer must fool a panel of humans into believing the machine is human. Blum, Ahn, Langford(2000)[2] proposed a class of ATT(called human interactive proof HID), which Hopper(2001)[3] describes as a protocol “ that allows a human to prove something to a computer.

Hopper and Blum (2001)[4] propose a HIP called secure Human Identification Protocol or Humanoid, in which computer must verify a human membership in a group without requiring a password , biometric data, electronic key or any other physical evidence.

Blum,Ahn,Langford (2000)[5] propose a “Completely Automated Public Turing Test To Tell Humans and Computer Apart” in which the computer must be able to generate and grade “ A test that most hum can pass.”

Example: EzGimpy-deter to spam attacks. First used by yahoo(2001)[6].

3.TYPES OF CAPTCHA:

3.1 Text based-a CAPTCHA features an image file of slightly distorted alphanumeric characters.(GIMPY)

3.2 Graphics based- Pix," presents the user with six images of a single subject, such as babies or horses, and asks them to define the subject of the pictures.

3.3 Audio based-To accomodate the visually-impaired, some CAPTCHAs use audio files.

3.4 Logic based-Which of ten, 8, 29, sixty four, 83 or fifty four is the smallest?

3.5 Math based- 4+5=?

4. APPLICATION OF CAPTCHA:

Automated attacks can exploit many services:

- Blogs
- Forums
- Phishing
- Theft of data

Registration Websites use CAPTCHA systems to prevent the bot programs from wasting their resources.

5. REALTED WORK

CAPTCHA was first introduced in 1997 when Andrei Broder devised the CAPTCHA method. In the same year, Altavista web site used this method to distinguish between computer programs and human user. In this method, a distorted English word was shown to the user and the user was asked to type it

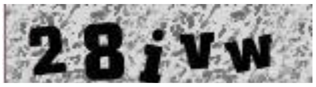


Figure. 2

5.1 OCR-Based CAPTCHA Methods

In OCR-based methods [7], the image of a word with distortion and various pictorial effects is shown to the user and he/she asked to type that word. Due to presence of various pictorial effects, the computer will encounter problems in the recognition of these words and only a human user can recognize the word. But these methods usually result in dissatisfaction of users. On the other hand, efforts have been made for attacking these methods . Examples of these methods include Gimpy, Handwritten CAPTCHA, and Persian/Arabic CAPTCHA.

5.2. Non-OCR-Based CAPTCHA Methods

In contrast, we can point to Non-OCR-based methods which are easier to work with than OCR-based ones. Examples of these methods include PIX, Text-to-Speech method, and Drawing CAPTCHA. Various methods have been proposed for breaking up the CAPTCHA. It can be done with the help of the segmentation and character recognition techniques. This can be done with the help of the Neural Network using divide and conquer approach. To provide defense against such techniques resistance method for CAPTCHA have also been proposed that includes Font tricks, Noise, Color model and Overlap.

6. PROBLEM STATEMENT:

The aim of this paper is to propose a method for generating image based on Text or rotating object within images to make difficult for automatic program to detect the captcha.

6.1 Proposed Solution for Generating Image Captcha:

6.2 Generating image based on Text:

For this purpose I will use image processing operations to convert simple text into image. The steps involved in the proposed method are as follows:

Step 1: Set Image type (JPEG).

Step 2: Create new image in buffer.

Step 3: Set background image white.

Step 4: Set Gradient font of text to be converted to graphics image.

Step 5: Write text in a image

Step 6: Release resources and write the output to output stream.

These steps are shown in figure 5.

6.3 Algorithm for Captcha Image Generation from simple text:gure

```
 imagecaptcha(inputstring)

{

BufferedImage image;

image=texttoimage(inputstring);

return image;

}
```

6.4 Algorithm for Image generation:

```
Image texttoimage (inputstring)

{

response.setContentType ("image/jpeg");
ServletOutputStream output=response.getOutputStream();


BufferedImage image1 = new BufferedImage (300, 50,

BufferedImage.TYPE_BYTE_INDEXED);

Graphics2D graphics1 = image1.createGraphics ();

// Set back ground of the generated image to white

graphics1.setColor (Color.WHITE);

graphics1.fillRect (0, 0, 300, 50);

// set gradient font of text to be converted to image

GradientPaint gradientPaint1 = new GradientPaint (10, 5,

Color.BLUE, 20, 10, Color.LIGHT_GRAY, true);

graphics1.setPaint (gradientPaint1);

Font font = new Font("Times New Roman", Font.ITALIC,

35);

graphics1.setFont (font);

// write input string in the image

graphics1.drawString (inputstring, 5, 30);

// release resources used by graphics context

graphics1.dispose ();

// encode the image as a JPEG data stream

JPEGCodec.createJPEGEncoder (output).encode (image1);

// close the stream

output.close ();

}
```

7.Results:

Above algorithm generates images by text as shown below:

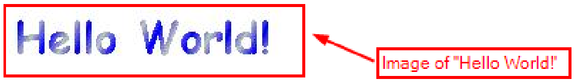


Figure3: Final output

This algorithm makes difficult for automatic program to Bypass the captcha.

Following graphs illustrate more time taken by automatic program to bypass the captcha for form filling. In the chart shown below, I show the time taken by automatic programs to bypass the captcha with using and without using this software and also shows the relative study for the same as follows (for approx 10 cases):

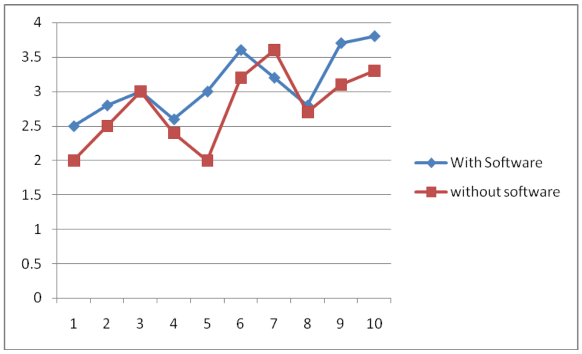


Figure 4:comparative study

8. FUTURE SCOPE:

The security analysis shows that this new design can prevent attacks efficiently from existing algorithms as well as possible ones using multiple frames. In short, this project will be a good guide for the design of next generation CAPTCHA. Our future research will be on how to design a more practical and safer dynamic CAPTCHA and the improvement in performance of the websites when these CAPTCHAs are used (Generally when these type of CAPTCHAs are used the performance decreases as the generation requires time for execution) .

9.CONCLUSION:

In this project we implemented a practical and safe image CAPTCHA from text, ensuring it not only extremely hard to recognize, but easy to identify for humans as well. It also makes full use of disadvantages of computers in recognizing images from a complicated background, making it still very difficult for computer programs to break.

CAPTCHAs are an effective way to counter bots and reduce spam. This proposed problem gives us the solution for dynamic Captcha. Applications are varied– from stopping bots to character recognition & pattern matching. Some issues with current implementations represent challenges for future improvements

Acknowledgement:

This research paper is made possible through the help and support from everyone,

Including: faculty members who give his/her valuable suggestions to me , family, friends, and in essence, all sentient beings.

Especially, please allow me to dedicate my acknowledgment of gratitude toward the following significant advisors and contributors:

First and foremost, I would like to thank Dr. Harleen Kaur for his most support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper.

Second, I would like to thank as well as all the other faculty members who have taught me about image processing and captcha over the past three years of my pursuit of the master degree.

Finally, I sincerely thank to my friends, who provide the advice. The product of this research paper would not be possible without all of them.

10 REFERENCES:

[1] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: using hard AI problems for security. In Eurocrypt, 2003.Luis von Ahn, Manuel Blum, Nicholas J. Hopper and John Langford. TheCAPTCHA Web Page: http://www.captcha.net. 2000.

[2]. Luis von Ahn, Manuel Blum and John Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI. To appear in Communications of the ACM.

[3]. Mihir Bellare, Russell Impagliazzo and Moni Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In 38th IEEE Symposium on Foundations of Computer Science (FOCS' 97),

[4] A. L. Coates, H. S. Baird, and R. J. Fateman. Pessimal Print: A Reverse Turing Test. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR' 01), pages 1154-1159. Seattle WA, 2001.

[5] http://www.lafdc.com/captcha /

[6] K. Chellapilla, P. Simard. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)

[7] Y. Lecun, L. Bottou, Y, Benshio, P. Haffber. Gradient-Based Learning Applied to Document Recognition. Proc. of the IEEE, November 1999.

[8] Mike O'Neill. Neural network for Recognition of Handwritten Digits.

[9] L. Bottou, Y, Benshio, Y. Lecun. Global Training of Document Processing Systems using Graph Transformer Networks.

[10] Y. Lecun, L. Bottou, Y, Benshio. Reading checks with multilayer graph transformer network.

[11] P. Simard, D. Steinkraus, J. Platt. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis.

[12] C Burges, O. Matan, Y. LeCun, J. Denker, L. Jackel, C.Stenard, C. Nohl, J. Ben. Shortest Path Segmentation: a method for training a Neural Network to Recognize Character Strings.

[13] D. You, G. Kim. An approach for locating segmentation points of handwritten digit strings using a neural network.

[14] G. Mori, J. Malik. Recognizing Objects in Adversial Clutter: Breaking a Visual Captcha.

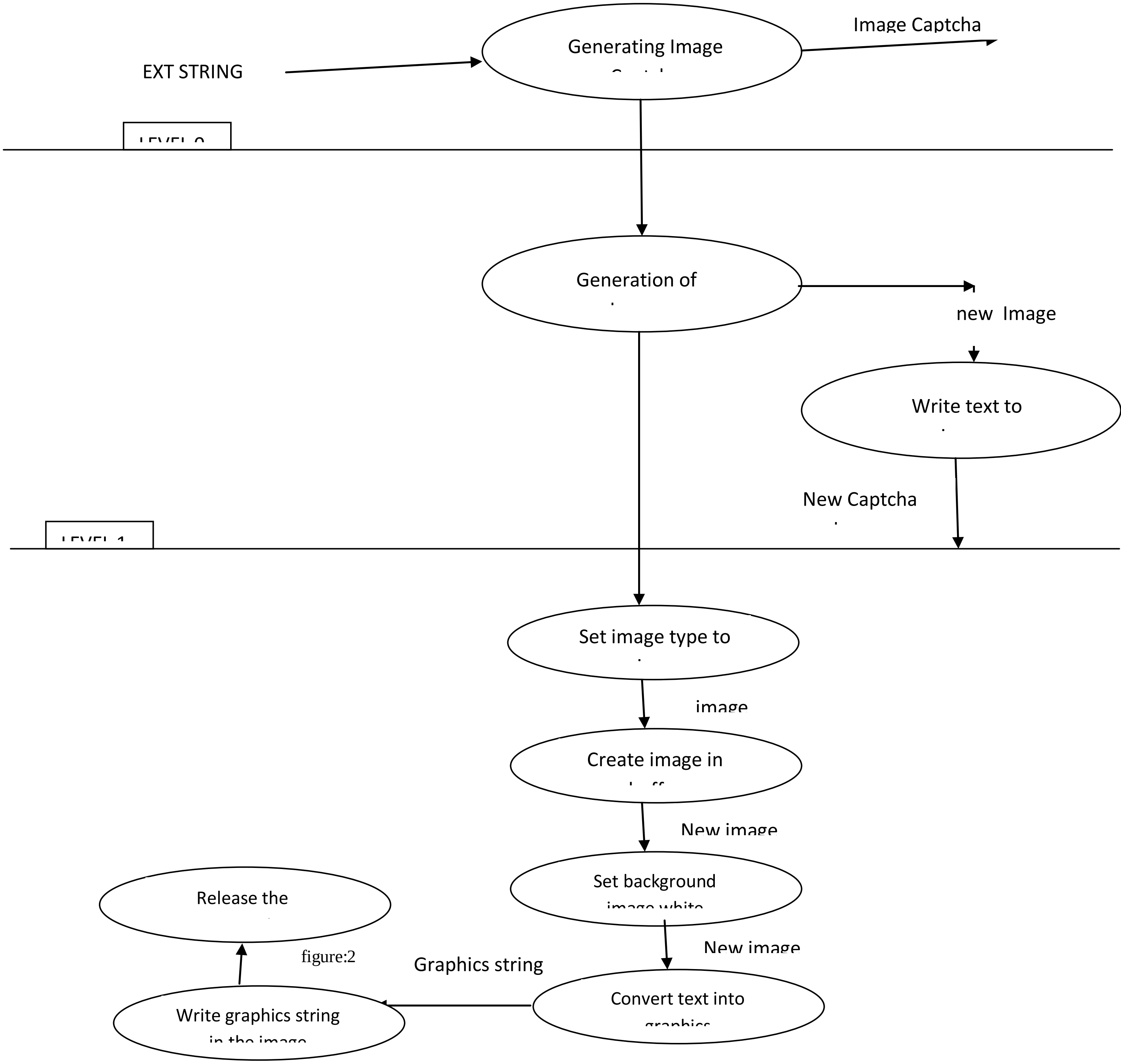


Figure 5: DFD for image generation from captcha