Certain investigation on Captcha design based on splitting, rotating and grid's usage in the images against OCR

M.RAJA[#1], A.BHARANIDHARAN [*2]

[#1]*PG Scholar ME-Software Engineering Final yea, Sri Ramakrishna Engineering College Coimbatore, Tamil Nadu, India*

[*2] *Assistant Professor*

Sri Ramakrishna Engineering College Coimbatore, Tamil Nadu, India.

Abstract— **Network security consists of the provisions and policies adopted by a network administrator. Network security prevents and monitors unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources.**

A CAPTCHA is a program that generates and grades tests that are human solvable, but beyond the capabilities of current computer programs. This technology is now almost a standard security mechanism for addressing undesirable or malicious Internet bot programs and has found widespread application on numerous commercial web sites including Google, Yahoo, and Microsoft's MSN. It is widely accepted that a good CAPTCHA must be both robust and usable.

The problem in existing system is that the CPU memory required to randomly generate CAPTCHA images is somewhat significant, and if CAPTCHA forms are being used on a lot of websites simultaneously this can reduce the performance of the web server. In order to provide effective CAPTCHA, A method used in CAPTCHA is implementing the images of words. This method is based on the weak points of Optical Character Recognition (OCR) programs. OCRs can recognize the high quality texts using the common formats and standards. It will be more secure to add noisy backgrounds, colors and increasing the level of distortion against character recognition programs. It is difficult for them to read low-quality text and the manuscripts.

Keyword s **— Captchas, Colour, Robustness, Usability, Internet security.**

I. INTRODUCTION

Internet computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This Internet model promotes availability and is composed of five essential characteristics, three service models and four deployment models

Most of the daily activities such as education, shopping or commerce are being carried out through the Internet. Users are commonly asked to fill out registration forms by entering required information to be able to operate specific tasks on the web sites. However; registration can be done by automated hacking software. Some people commit vandalistic acts such as attacking web sites with computer programs, and even can stop the running of the website. These programs automatically fill out form with wrong information to get in the web site. Therefore, web site holders are supposed to take precautions against those attacks for security. Several defense systems have been proposed and presented in order to prevent such attacks. It is crucial for the websites to have a system which has the capability of distinguishing human users and computer programs in reading images of text. CAPTCHAs are challenge puzzles used to determine whether a user is human or not.

Intuitively, a CAPTCHA is a program that can generate and grade tests that most humans can pass but current computer programs cannot pass. It stands for **C**ompletely **A**utomated **P**ublic **T**uring **T**est to **T**ell **C**omputers and **H**uman **A**part, and Public means that the code and the data used should be publicly available. There are several types of testing such as pictures of objects, distorted text, or even audio clips for impaired users. A more technical definition of CAPTCHA is provided: "CAPTCHA is a cryptographic protocol whose underlying hardness assumption is based on an AI problem". The most common applications for practical security by CAPTCHA test include online polls, free email services, shopping agents, search engine bots, worms and spam, and preventing dictionary attack. For instance, email provider services such as Hotmail and Yahoo provide a CAPTCHA test as a final step of the registration process to stop bots from subscribing and using their resources for spam distribution.

Turing test is used for providing the intelligence of a computer in the domain of Artificial Intelligence (AI). Turing tests use a method which put a human user and a computer in different rooms. There is also third room for the human interrogator to ask them questions. If the interrogator cannot recognize the locations of human and computer, it results that the computer has passed the Turing test. CAPTCHA is a Turing test but it is quite different than the definition above. If the interrogator is replaced with computer

rather than a human, then it is called as CAPTCHA. The main function of this method is human user can easily answer the interrogator's question but present computer programs are hardly or never can answer.

II. PROBLEM FORMULATION

In general, breaking a CAPTCHA (in the sense of writing computer programs that automatically solve the test) involves a segmentation task, which is to locate individual characters in the right order, and a recognition task, which is to recognize which character is which. CAPTCHA can be reduced to the problem of recognizing individual characters in its challenge, and then this CAPTCHA is effectively broken.
Their segmentation attack works as follows: They first used the CFS method to extract black components, which define most of the actual content of each character and are never shared with adjacent characters. The result of extraction of all black components each was being highlighted with a different color.

The second step of their attack is to identify and extract shared white components, which are the connecting areas between adjacent characters. We first apply the CFS method to detect all white components. Then we exclude the main image background (i.e. the outside of the image text), which is the largest white component in terms of pixel count, and the remaining white components.

The final step is to put the shared white components in the right location to merge with corresponding black components to form each complete character. We know that n number of characters when connected horizontally should typically produce n-1 connection areas between them, and that shared white components that are juxtaposed vertically with each other must belong to the same connection area.

There are a number of important characteristics that a CAPTCHA can exhibit. These include the difficulty to be solved by OCR and any attack programs, readable common distortions, resisting malicious attacks, carrying many bits of information, the capability of coexisting with other CAPTCHAs, and little cognitive computation requirement by the user. The relative importance of these characteristics depends on the CAPTCHA type. The principles behind CAPTCHA are as follows:

- The user is presented with a garbled image on which some text is displayed. This image is generated by the server using random text.
- The user must enter the same letters in the text into a text field that is displayed on the form to protect.
- When the form is submitted, the server checks if the text entered by the user matches the initial generated text. If it does, the transaction continues. Otherwise, an error message is displayed and the user has to enter a new code.
- Exploits observation that humans are still much better than computers at many pattern recognition tasks.

Users are required to register regarding website in order to enroll web activities. However, registration can be done by automated hacking software. That software make false enrollments which occupy the resources of the website by reducing the performance and efficiency of servers, even stop the entire web service. It is crucial for the websites to have a system which has the capability of differing human users and computer programs in reading images of text. Completely Automated Public Turing Test to Tell Computers and Human Apart (CAPTCHA) is such a defense system against Optical Character Recognition (OCR) software. OCR can be defined as software which work for defeating CAPTCHA images and make countless number of registrations on the websites

Reading-based Captcha challenges typically comprise a segmentation challenge followed by recognition challenges2. Solving the segmentation challenge requires the identification of character locations in the right order. The random location of characters, background textures, foreground and background grids or lines, and clutter in the form of arcs make the segmentation problem difficult. Image warp exacerbates the segmentation problem by reducing the effectiveness of preprocessing stages of a segmentation algorithm that attempt to estimate and remove the background textures and foreground lines, etc. Once character locations are reliably identified (in the right order) each of the characters needs to be recognized correctly giving rise to the recognition problem. The character recognition problem is made difficult through changes in scale, rotation, local and global warp, and intersecting random arcs.

1. CAPTCHA *security*

The strength of a captcha (against a computer algorithm) is a combination of the strengths of the constituent segmentation and recognition problems. Several recent efforts have shown that weaknesses in particular HIPs can be easily exploited to break them . Many of the online HIPs are pure recognition tasks that can be easily broken using machine learning (in these CAPTCHAs the segmentation problem is trivial to solve). In light of these results, while the recognition challenges still pose a problem, the segmentation challenge is more important in determining CAPTCHA strength.

2. *Single Character Recognition*

The recognition challenges posed by the CAPTCHAs are specifically designed to fool off-the-shelf OCR systems and several others. These general purpose OCR systems are designed for high quality document scans or images and are brittle to character warp and degrade rapidly in the presence of clutter. On the contrary, one can attempt to solve the recognition problem posed by a particular CAPTCHA building a custom recognizer using machine learning. The custom recognizer is trained on distorted characters extracted from CAPTCHA samples. This approach requires a new recognizer to be built for each CAPTCHA type. This was exactly the approach adopted in Convolution neural networks and it used to build recognizers for the Mailblocks, Register, Yahoo!, Ticketmaster, and Google CAPTCHAs. When solving the recognition problem, the segmentation problem is

assumed to be solved, i.e., we already know the number of characters in the CAPTCHA image and their locations. These locations need not be exact. Some tolerance is allowed (a few pixels) as a certain degree of translation invariance can be expected from machine learning based recognizers.

III. METHODOLOGY

In this technique, a new method has been developed for differing human users and computer programs from each other by mainly splitting CAPTCHA image into several parts with rotation and drawing a great deal of lines and circles randomly to the background. Additionally, a grid effect has been added to the background. Lines and circles have been randomly drawn in the color of text so that OCR program confuse while distinguishing which one is character or not. In our method, CAPTCHA text consists of the characters and number in a range of"ABDEFHKLMNPRSTUVWXZabdefgikmnopqrstuvwxyz 023456789". The text is composed of five characters, and each character has its own bending and size value. Characters are split into several parts and each part is given randomly a rotation value in a certain angle domain interval such as: [-1˚ 1˚], [-3˚ 3˚], [-5˚ 5˚]. Image parts are also split individually with random width and height values which provide an extra difficulty for OCR programs while finding the start and end of the images. Rotation in character parts provides confusion in recognizing the exact one. The text shown in Figure 6.3 below is indeed‘W9XZq’. This text is easily recognizable by the human but not OCR program. This CAPTCHA image is split into 8 parts as (4 X 2) matrix shape and each split has a random rotation angle value between -3 and3 degrees. Splits have random width and height values. Background and CAPTCHA text are in similar colors.

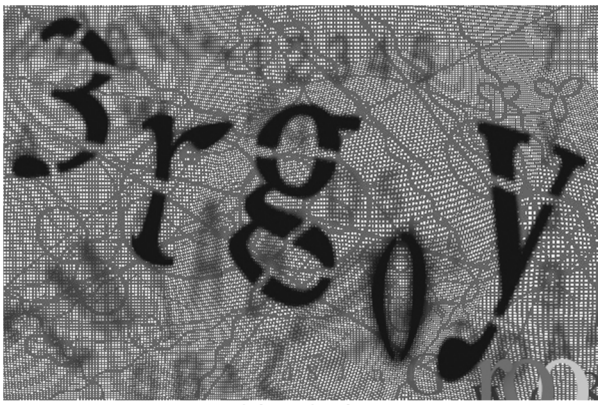

Fig 1: Drawing CAPTCHA image.

There is a grid in black color at the background. Lines and circles are also drawn in black color such as the CAPTCHA text. When you look at the first character, it is not easy to recognize the letter exactly due to rotation and splitting of character image. It seems like ‘V’ or ‘I’ or ‘W’. In fact, it is ‘W’ but it is not recognizable for OCR program because character‘W’ is split into two different parts as ‘V’ and ‘I’. The other letters in CAPTCHA image have same difficulty for OCR program. The programming steps of the algorithm that developed to generate CAPTCHA images are given with pseudo code and runtime output screenshots as in follows;

Step 1. Start the session.
Step2. Generate n letters random string from the string "ABDEFHKLMNPRSTUVWXZabdefgikmnopqrstuvwxyz023456789".// Take out some easy letters not to be confused by the user; C/G I/l Q/O h/b.// Users of this algorithm may choose other languages such as Arabic and Korean by modifying the string as they wish.
Step 3. Create the hash for the random text and put it into the session.
Step 4. Create transparent CAPTCHA image with wbyh image size and add CAPTCHA text over it. Transparent CAPTCHA image with text can be created by specific PHP (Personal Home Page) built-in function: imagettftext().
Step 5. Set the initial X-position and Y-position ofcaptcha image to 0.
Step 6. Split the captcha into k by l Matrix shape bydividing the captcha width into k parts and the height intol.
Step 7. Start a loop from 0 to k*l
// After completing the first row in order to split into kparts, then pass to next row.
If (i+1) Mod k+1 = 0 ThenSet initial X-Position to 0 and initial Y-Position toSplit Height (Image Height / l)End If
Step 7.1. Create an array to put the split parts and putthe split images into array.
Step 7.2. Randomize integer between -d and d to giverandom rotation to the splits. Rotate the splits withrandomized variable that is random between -d and d.
Step 7.3. To pass to another split in one row, increasethe initial X-Position by Split Width (Image Width / k)in each loop step.
Step 7.4. End Loop.
Step 8. Combine the splits to create new CAPTCHAwith split and rotation.
Step 9. Add background to transparent new Captchaimage object with randomly drawn lines and specialeffects (Number of lines=250, line color is black andadd grid effect).
Step 10. Export the final CAPTCHA image as a JPEGfile in the name of ‘captcha.jpg’.
Step 11. Destroy the final CAPTCHA image object tobe refreshed in each session.

IV. THE FRAMEWORK

A. Separating foreground from background

Uniformity in foreground or background would reduce resistance to segmentation attacks in general. A good security design principle would be minimal uniformity in foreground and background with a controlled cost for readability.

Contrast between foreground and background, or contrast among foreground characters would reduce resistance to segmentation attacks in general. For this reason, coloring that increases such contrasts is undesirable.

Perceptually connected but physically disconnected component is another good security design principle; however, coloring may enhance or destroy perceptual grouping with security consequence. Therefore a careful evaluation of the end design's security is essential.

B. Segmentation task

Using the method of Color Filling Segmentation (CFS), the segmentation process will be done works as follows. The basic idea is to detect every connected large component, which often corresponds to each individual character (or stroke). Our algorithm first detects a black pixel in the image on the right hand side and then traces all its black neighbors until all the connecting black pixels are traversed - that is, a component is identified and segmented. Next, the algorithm locates a black pixel outside the area of the already identified component(s), and starts another traversal to identify the next component. This continues until all black components are identified. This method is effectively like using a distinct color to flood each connected component. In the end, each black component is highlighted with a distinct color, and the number of colors used is the number of black components in the image.

C. Recognition task

After segmentation, it is trivial to apply standard techniques to recognize each individual character at a high speed. We effectively extracted all challenge characters using the CFS method. As such, this scheme is effectively broken. On the other hand, in this case, gray scale images do not appear to cause much degradation of usability. So the OCR can be defined as software which work for defeating CAPTCHA images and make countless number of registrations on the websites. This study focuses on a new method which is splitting CAPTCHA images into several parts with random rotation values, and drawing random lines on a grid background. Lines are in the same color with the CAPTCHA text and they provide a distortion of image with grid background.

D. Splitting and rotating The images against ocrs for CAPTCHA design

Completely Automated Public Turing Test to Tell Computers and Human Apart (CAPTCHA) is such a defense system against Optical Character Recognition (OCR) software. OCR can be defined as software which work for defeating CAPTCHA images and make countless number of registrations on the websites. This module focuses on a new method which is splitting CAPTCHA images into several parts with random rotation values, and drawing random lines on a grid background. Lines are in the same color with the CAPTCHA text and they provide a distortion of image with grid background. In this module, a new method has been developed for differing human users and computer programs from each other by mainly splitting CAPTCHA image into several parts with rotation and drawing a great deal of lines and circles randomly to the background. Additionally, a grid effect has been added to the background. Lines and circles have been randomly drawn in the color of text so that OCR

V. CONCLUSIONS

CAPTCHAs are amongst the most widely used Human Interaction Proofs (HIP) on the Internet. Their working principle lies in distorting text characters in such a way that recognition became difficult for computers and remains easy for humans.

The new CAPTCHA security was developed by rotating the characters used in captcha images and using the distortions on background of the image, making uniformity in foreground and provides contrast among foreground characters.

The new CAPTCHA method use same input methods similar to the other many well known websites and services where users type some keywords or characters into the input boxes. Therefore it can be easily learnt and used by regular users. It can be user belonging to all ages without any training.

REFERENCES

[1] L von Ahn, M Blum, J Langford. "Telling Humans and Computer Apart Automatically", CACM, V47, No2, 2004.
[2] K Chellapilla, K Larson, P Simard, M Czerwinski, "Designing human friendly human interaction proofs", ACM CHI'05, 2005.
[3] K Chellapilla, K Larson, P Simard, M Czerwinski, "Building Segmentation Based Human-friendly Human Interaction Proofs", 2nd Int'l Workshop on Human
Interaction Proofs, 2005.
[4] K Chellapilla, K Larson, P Simard, M Czerwinski, "Computers beat humans at single character recognition in reading-based Human Interaction Proofs", 2nd Conference on Email and Anti-Spam, 2005.
[5] HS Baird, MA Moll, SY Wang. "A highly legible captcha that resists segmentation attacks".2nd Int'l Workshop on Human Interaction Proofs, 2005.
[6] LW MacDonald. "Using Colour Effectively in Computer Graphics". IEEE Computer Graphics and Applications, July/August 1999.
[7] W3C Working Group, "Inaccessibility of CAPTCHA - Alternatives to Visual Turing Tests on the Web", Nov, 2005.
[8] J Yan, AE Ahmad. "Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms", *ACSAC'07*. pp 279-291.
[9] J Yan, AE Ahmad. "A Low-cost Attack on a Microsoft CAPTCHA", 15th ACM CCS'08. pp. 543-554.
[10] AE Ahmad, Jeff Yan, Lindsay Marshall. "The Robustness of a New CAPTCHA". EuroSec 2010.
[11] J Yan, AE Ahmad. "Usability of CAPTCHAs or usability issues in CAPTCHA design", SOUPS, 2008, pp. 44-52.
[12] H Yeend. "Breaking CAPTCHAs without using OCR". 2005.
[13] G Mori, J Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," IEEE CVPR, 2003
[14] G Moy, N Jones, C Harkless, R Potter, "Distortion estimation techniques in solving visual CAPTCHAs," IEEE CVPR, 2004.