The ISC Int'l Journal of Information Security

July 2009, Volume 1, Number 2 (pp. 105–123)

http://www.isecure-journal.org

Image Flip CAPTCHA

M. Tariq Banday [a,*], Nisar A. Shah [a]

[a] *P. G. Department of Electronics and Instrumentation Technology, University of Kashmir, Srinagar, 190006, India*

ARTICLE INFO.

Article history:
Received: 19 March 2009
Revised: 26 June 2009
Accepted: 5 July 2009
Published Online: 25 July 2009

Keywords:
CAPTCHA, HIP, Botnet, Image CAPTCHA, Clickable CAPTCHA, CAPTCHA Security, CAPTCHA Usability

ABSTRACT

The massive and automated access to Web resources through robots has made it essential for Web service providers to make some conclusion about whether the "user" is a human or a robot. A Human Interaction Proof (HIP) like Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) offers a way to make such a distinction. CAPTCHA is a reverse Turing test used by Web service providers to secure human interaction assumed services from Web bots. Several Web services that include and are not limited to free e-mail accounts, online polls, chat rooms, search engines, blogs, password systems, etc. use CAPTCHA as a defensive mechanism against automated Web bots. In this paper, we present a new clickable image-based CAPTCHA technique. The technique presents user with a CAPTCHA image composed of several sub-images. Properties of the proposed technique offer all of the benefits of image-based CAPTCHAs; grant improved security than that of usual OCR-based techniques, consume less Web page area than most of image-based techniques and at the same time improve the user-friendliness of the Web page.

1 Introduction

Web bots [1] are scripts or applications designed to perform predefined functions repeatedly and automatically after being triggered intentionally or through a system infection. Although bots were originated as a useful feature on the Web for carrying out repetitive and time consuming operations but its ability to imitate human behavior has been exploited for malicious intent. According to the level of complexity in the operations performed by a bot, it may be classified as a 1st generation, 2nd generation or 3rd generation program [2]. 1st generation bots are used to download a predefined set of resources without interpreting their content. 2nd generation bots are capable of analyzing the downloaded content and perform other actions on the basis of the meanings derived from that content. 3rd generation bots are capable of fully interpreting client-side script languages such as VBScript, Java Script, and Flash. Further, 3rd generation bots can derive meaning from the downloaded content with intelligence similar to that of a human user. Several anti-bot defense strategies that include HTTP client and server-side strategies have been developed to protect Web applications from these bots. HTTP-based strategies have been able to protect Web applications to a larger extend from 1st generation bots and to some extend from 2nd generation bots but cannot be used against 3rd generations of bots [3].

Human Interaction Proof (HIP) [4] which are schemes that require some kind of interaction from a human in order to be recognized as a human or a

* Corresponding author.
Email addresses: sgrmtb@kashmiruniversity.ac.in (M. T. Banday), elec@kashmiruniversity.ac.in (N. A. Shah).
ISSN: 2008-2045

member of a group [5] have been able to effectively prevent malicious programs from getting access to the Web services. Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [6] is a class of HIP tests and is easier for humans to qualify and tough for bots to simulate. CAPTCHAs underlying hardness is based on Artificial Intelligence and the test may be Optical Character Recognition (OCR)-based, image-based or audio-based. A good CAPTCHA should be generated in a manner that satisfies a set of desired properties [6] that include: a) the test must be generated automatically, b) the answer to the test should be quick and easy, c) the test should accept all human users and d) the test should resist attacks even if the algorithm is known. CAPTCHAs are used in diverse Web services for securing online polls, preventing spammers from spamming and getting free e-mail accounts [7], detecting phishing attacks [8], preventing bot entry into the chat system, preventing online dictionary attacks in password systems [9, 10], preventing unruly search engine bots from indexing private Web pages, preventing Web bots from adding advertisements to comment field in blogs, preventing download bots from downloading and archiving Web sites or FTP servers [11, 12]. Different types of CAPTCHA tests pose different disadvantages to legitimate users. These include one or more of the following: unfriendliness, requirement for larger image library, increased load on servers, delays in Web page download, accessibility problems and annoyance to genuine users.

The main contribution of this paper is the design of a new clickable image-based CAPTCHA technique that has been tested on a limited number of humans. The technique presents a high success rate and effective usability. It offers advantages of faster CAPTCHA image downloads and occupies less screen area, making it easy to integrate within the Web page. It is an image-based CAPTCHA technique and does not require the user to read any message or type a response and therefore is as friendly to foreigners as it is to native users. Further, it is simple to be designed and worked with; the process of entry of response is easy and fast making it usable for portable devices. The paper further contributes by identifying some usability issues for image-based CAPTCHA techniques.

The remaining text of this paper is organized as follows: Section 2 presents a review of existing CAPTCHA techniques. Section 3 illustrates the proposed Image Flip CAPTCHA technique. In sections 4 and 5, we respectively discuss security and usability of the proposed technique. In Section 6, we summarize the results obtained through user studies and experiments.

2 Related Work

Most of the CAPTCHA tests are OCR-based and are in the form of visual image containing a difficulty to recognize text string to be identified and typed by the user in a text box provided near the CAPTCHA image on the Web page. The CAPTCHA image is often of low quality with different forms of noise and strong degradation applied to it. Andrei Broder, chief scientist of Altavista and his colleagues devised the CAPTCHA method in 1997 and in the same year Altavista website used this method as an HIP [13]. This method used a distorted English word that a user was asked to type. The distorted word was easier for users to understand but difficult for bots to recognize, using OCR techniques. Blum and Von Ahn at Carnegic Melton University in Collaboration with Yahoo devised EZ-Gimpy and Gimpy CAPTCHA [14] to protect chat rooms from spammers. These CAPTCHAs due to limited words in their dictionary (860 words) have been broken [15]. A more secure type of OCR-based HIP, called reCAPTCHA [16] has been proposed by the same authors. Baffle Text CAPTCHA [17] is the Xerox Pato Alto Research Center (PARC) version of Gimpy test. Among various other OCR-based CAPTCHAs proposed in literature, some are: Scatter Type [18], Handwritten Word-based CAPTCHA [19], Human Visual System masking Characteristic CAPTCHA [20], Persion/Arabic CAPTCHA [21] and Question-based CAPTCHA [22] techniques. Various service providers on the Internet like PayPal, YouTube, Yahoo and Hotmail use their own versions of OCR-based CAPTCHAs on their websites and update them with newer versions frequently. Recently, Richard Chow *et al.* presented a generic technique for converting a textual CAPTCHA into a clickable one [23]. This technique proposes placement of multiple text CAPTCHA images in a grid among which some are English words while others are not. To solve this CAPTCHA, the user must click on all valid English words. Samples of some OCR-based CAPTCHA techniques are shown in Figure 1.

The second class of CAPTCHA tests is image-based and presents a visual pattern or concept that the user needs to identify and act accordingly. Different image-based CAPTCHA schemes use different patterns or concepts which are easy to be recognized by the users and much more difficult for the bot programs to simulate. Initially image-based CAPTCHA called ESP-PIX CAPTCHA [6] was proposed by Blum and Von Ahn. It used a larger database of photographs and animated images of everyday ob-

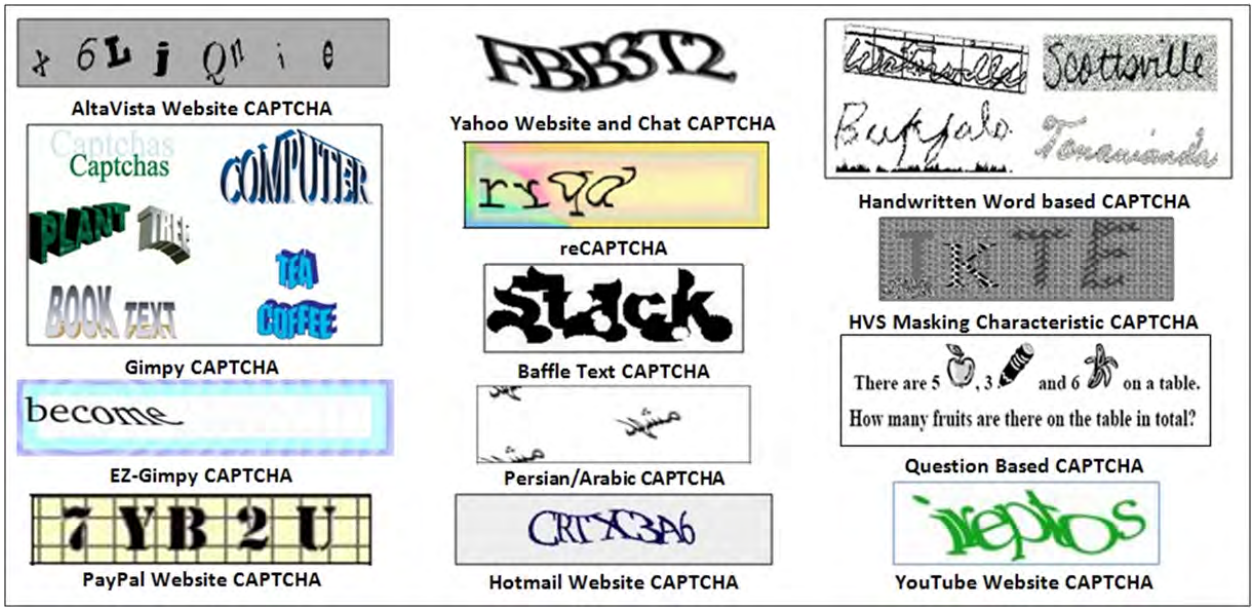

Figure 1. Samples of OCR-Based CAPTCHA Techniques

jects. The CAPTCHA system presented a user with a set of images all associated with the same object or concept. The user was required to enter the object or concept to which all the images belonged to e.g. the program might present pictures of globe, volleyball, planet and baseball expecting the user to correctly associate all these pictures with the word ball. Besides several others, this class of CAPTCHA methods include: Microsoft Asirra [24], IMAge Generation for INternet AuthenticaTION (IMAGINATION) [25], Image Block Exchange [26], Face Recognition [27], Multilingual [28] and KittenAuth (www.thepcspy.com/kittenauth) CAPTCHAs. Yet, another CAPTCHA challenge is Implicit CAPTCHA [29]. It proposes single click challenges distinguished as necessary browsing links which can be answered through experience of the context of the particular Website. A CAPTCHA based on image orientation [30] has been recently proposed by Google. This CAPTCHA method is based on adjusting randomly rotated images to their upright orientation. Mosaic-based Human Interactive Proof called MosaHIP [11] proposes a CAPTCHA scheme for securing the download of resource against Web-bots. It uses a single larger image called mosaic image which is composed of smaller and partially overlapping real and fake pictures. The user needs to drag a resource expressed in form of a movable text object on the web page and drop it onto the area of the mosaic picture containing the image indicated in the CAPTCHA image. Samples of some image-based CAPTCHA techniques are shown in Figure 2.

The third class of CAPTCHA tests are called Audio CAPTCHAs [31] which generally take a random sequence drawn from recordings of simple words or numbers, combine them and add some disturbance and noise to it. The CAPTCHA system then asks the user to enter the words and/or numbers read in the recording. The first audio-based CAPTCHA was implemented by Nancy Chan. Audio CAPTCHAs are more difficult to solve, hard to internationalize and more demanding in terms of time and efforts in comparison with OCR-based and image CAPTCHAs. However, audio-based CAPTCHA tests have become an alternative for visually-impaired people.

Apart from these three classes, Collaborative Filtering [32] has also been proposed in literature but is not practically used as an HIP. Collaborative filtering attempts to extract complex patterns that reflect human choices. This CAPTCHA approach differs from others in the scenes that Collaborative Filtering CAPTCHA designers do not initially know the correct answer and measure it from human opinion.

OCR-based CAPTCHAs are most widely deployed and are in use since years in major web sites. Further, they are intuitive to users and can provide strong security if properly designed. Early OCR-based CAPTCHAs were straight-forward for humans to solve. Advances in OCR techniques and consequently efficiency of bots in breaking OCR-based CAPTCHAs are improved as a result of which OCR-based CAPTCHAs are designed hard. It seems that OCR-based CAPTCHAs have become hard for ordinary users and as such they face difficulties in solving them. Often ordinary users fail to solve hard OCR-based CAPTCHAs in their first attempt. OCR-based CAPTCHA techniques have localization issues and

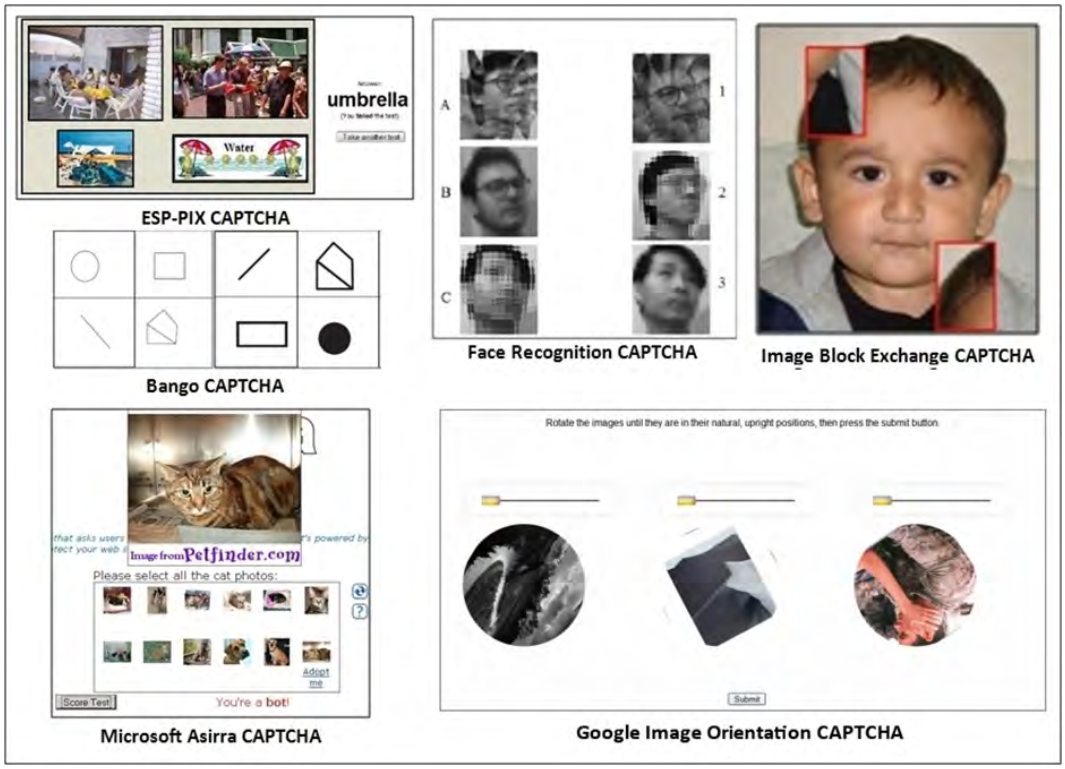

Figure 2. Samples of Image-Based CAPTCHA Techniques

thus are not friendly to foreigners. They use text box to input responses from the users, which in comparison with other user interfaces besides being difficult to work with, are also time consuming. Image-based CAPTCHA schemes have been proposed as an alternative to OCR-based CAPTCHAs but they have not been able to replace OCR-based methods mainly because image based schemes require larger Web page area, involve huge server processing and need an image database to be maintained at the server. Breaking a CAPTCHA challenge is difficult and it is very rare to find 100% success rate, however several CAPTCHA implementations have been broken and thus are proved to be inefficient. Research work carried out in [33] discusses the breaking of Microsoft CAPTCHA technique. Jeff Yen and his colleagues in the research work reported in [34] demonstrate the breaking of most visual CAPTCHA schemes publicly available as Web service for CAPTCHA generation at http://captchaservice.org. Similar research work carried out in [35] and [36] report the breaking of CAPTCHAs using distortion estimation and machine learning techniques. The advances in OCR techniques in terms of pattern recognition and computer vision have made CAPTCHAs prone to more and more attacks [37] and thus, it is reasonable to create new CAPTCHA challenges that are both secure and usable.

3 Proposed Method

In this section we present a new CAPTCHA technique that aims to determine legitimate users and at the same time does not alienate them. The scheme works on current difficulty of image segmentation algorithms in presence of a complex background. The algorithm implementation has been made using GIMP (GNU Image Manipulation Program) [38]. The security analysis has been carried out using GIMP and Edge Detection and Image Segmentation (EDISON) System.

In the proposed technique a composite CAPTCHA image of a reasonable dimension and resolution is shown to the user. The user has to identify positions of all embedded images that appear as normal with no flip applied to them from the shown composite image. The user needs to click on every non-flipped embedded image to prove human interaction. CAPTCHA images shown in figures 3 and 4 depict this technique. In both CAPTCHA images eight small images are embedded out of which three images appear as normal without any flip applied to them and the rest appear in flipped state. Both images have been developed using the same algorithm but to image shown in Figure 4, a background image with a desired transparency factor has been applied.

In this technique, a subject-wise database of small-sized, well-known small real world objects is created.

Before storing an image in the image database it is scaled to a small size. It is not required to store the tags corresponding to each image, however if it is desired to show images from a particular category of images or ensure generation of images in each CAPTCHA image from more than one category, tags may be stored with each image. In this case the images and the corresponding tags associated with each image are stored in separate places which are related with each other on an encoded key within a single database so that even if a hacker is able to break into the database, he is not able to get any meaningful content. As an option, a dynamic image database could be created by downloading images from the Internet [39]. In such cases, a secure mechanism has to be devised to collect the asymmetric images and classify them in flipped and non-flipped classes. A CAPTCHA image is created by placing randomly selected images from this database on a larger image to which some complex background has been applied. Various transformations are applied to each selected image before placing them on the larger CAPTCHA image. Coordinates of all non-flipped images are preserved at the server. The created CAPTCHA image thus is shown to the user to prove his humanity. Once the user clicks on all non-flipped images, he gains entry to the service otherwise after a few failed attempts a new CAPTCHA image is shown to the user.

The algorithm for creation of CAPTCHA image is mentioned hereunder:

Step I. Create a composite CAPTCHA image C_{img} of size $N \times M$ pixels with a color gradient/RGB noise between randomly chosen colors from RGB.

Step II. Draw Z random colored and /or filled and random sized shapes of circles, arcs and lines at random places on the CAPTCHA image C_{img} created in step I. Choose Z as any value between minimum Z_{min} and maximum Z_{max} depending upon the required complexity of the CAPTCHA image. This step may be repeated with different values for Z after step VII.

Step III. Apply a required transparency factor to the CAPTCHA image C_{img}. Optionally choose an image from the image database Img_{DB} and place it on C_{img} with a desired transparency factor.

Step IV. Choose P images from the image database Img_{DB} and add them to the set of selected images Sel_{imgs}. P should satisfy the following relationship with the CAPTCHA image C_{img}.

$$P \leq \frac{Sizeof(C_{img})}{MaxSizeof(Sel_{imgs})}$$

Step V. Randomly choose two or more images from the selected set of images Sel_{imgs} as the flipped images $SelF_{imgs}$ and apply a 180° flip to each such image.

Step VI. For each image in Sel_{imgs} apply a transparency factor T_F and scaling factor S_F such that:

$$TMin_F \leq T_F \leq TMax_F \text{ and}$$
$$SMin_F \leq S_F \leq SMax_F$$

Here, $TMin_F$, $TMax_F$, $SMin_F$, and $SMax_F$ are respectively the minimum and maximum values for transparency and scaling factors that can be applied to each selected image.

Step VII. Place each image from the selected set of images Sel_{imgs} including the flipped images on the CAPTCHA image C_{img} without overlapping them at random positions. Calculate the co-ordinates of all non-flipped sub-images $SelNF_{imgs}$ from the size of respective sub-image and its randomly generated position.

Step VIII. Return the CAPTCHA image C_{img} and the coordinate set C of all non-flipped images $SelNF_{imgs}$.

Once the CAPTCHA image and the co-ordinate set of all non-flipped images are generated, the CAPTCHA image is shown to the user and the co-ordinate set is used at the server to evaluate the user's response.

Optionally, instead of using small-sized well-known small real world objects, photographs of individuals and groups can also be used as an image source. However, in this case the above stated algorithm requires the following changes:

- The photographs need to be scaled down in size before being stored in the image database.
- Depending upon the size of the CAPTCHA image and number of photos to be placed on the CAPTCHA image, randomly selected photographs from the image database are resized and placed evenly on the CAPTCHA image instead of placing them randomly.
- Further, randomly colored horizontal and vertical lines of varying lengths are drawn on the resultant CAPTCHA image to make it difficult for the image segmentation algorithms to detect the boundaries of the photographs within the CAPTCHA image.

A CAPTCHA image shown in Figure 5 depicts this technique. As can be seen from this image that there are 16 groups and individual photographs out of

Figure 3. Image Flip CAPTCHA (Image 1)

Figure 4. Image Flip CAPTCHA (Image 2)

Figure 5. Image Flip CAPTCHA (Photographs) (Image 3)

which five are in non-flipped state. The user needs to click on these photographs to gain entry into the system. A chosen size of the CAPTCHA image can have more or less than 16 sub-photographs so as to make segmentation difficult.

Figure 6. Laplace Edge Detection (Image 1)

Figure 7. Laplace Edge Detection (Image 2)

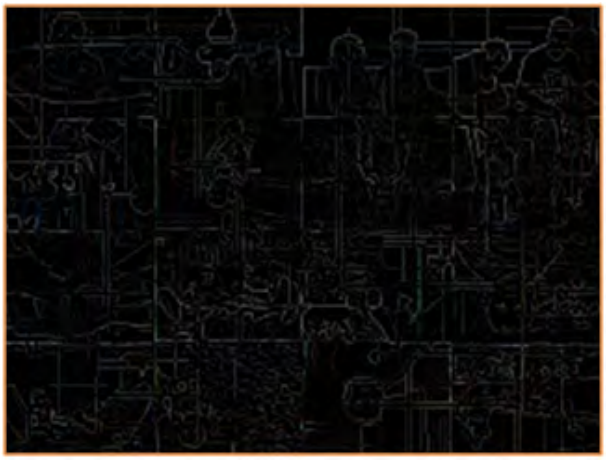

Figure 8. Laplace Edge Detection (Image 3)

Minimum and maximum allowed values for transparency factor T_F and scaling factor S_F can be separately maintained for each sub-image by incorporating additional fields in the image database which can be automatically controlled by keeping track of images that legitimate users have failed to recognize us-

ing an algorithm similar to Partial Credit Algorithm [24].

The proposed technique has some similarities with IMAGINATION, MosaHIP, Google Image Orientation and Microsoft Assira CAPTCHAs. All of these CAPTCHA techniques aim to exploit human imagination power in understanding images. In all above mentioned systems except Google Image Orientation, the user is presented with a set of images to form a single composite image. IMAGINATION CAPTCHA uses both images and text and is a two step process; first, the user needs to click near center of any image he wants to annotate and next he has to choose a word pertaining to the clicked image. Unlike IMAGINATION CAPTCHA, the proposed technique is a single step process where the user has to identify specific images to click upon. The proposed technique being truly image-based and one-stage system is more usable and faster than IMAGINATION CAPTCHA. Google Image Orientation and Microsoft Assira CAPTCHAs do not apply distortions to the sub-images while as our proposed technique like MosaHIP applies complex distortions to sub-images so as to make segmentation difficult. Unlike MosaHIP which uses real and fake images, our proposed technique uses only real pictures. It is beyond the scope of this work to perform a complete comparison of the above mentioned techniques. A key comparison of some image CAPTCHA techniques with our proposed technique is presented in Table 1. The proposed technique like all other image CAPTCHA techniques has a requirement of large set of images in its image database, which can be considered as a drawback when compared to OCR-based CAPTCHA techniques.

4 Security Analysis

Security of a CAPTCHA test is its resistance to attacks aimed to break its underlying protocols via man-in-the-middle or oracle attacks [34]. A CAPTCHA test may be considered secure that is at least as expensive for a hacker as it would cost him using human operators [18, 24]. The security of a particular CAPTCHA test can be analyzed by investigating its resistance to attacks that possibly may be used to break it [24]. Further, tests against real users and bots can greatly help with ascertaining its security state. Similar approaches for security analysis have been used in research work carried out in [11, 26]. In this section, we present various possible approaches that may be used to defeat our proposed CAPTCHA technique. We also discuss the security control of our proposed CAPTCHA technique against each such possible attack.

The first possible approach applies various image segmentation operations for content extraction and identification to identify the portions of each sub-image and their flipped state. The second possible approach is shape matching, which attempts to collect all sub-images that appeared previously and use image comparison to identify the flipped state of sub-image in case the same image appears again. The third approach applies a random guess to identify the flipped sub-images. The forth possible approach known as farming-out attacks, involve use of an unaware user to solve the test for the bot program. Further, image database attack and implementation attacks may also be tried to break the proposed technique.

4.1 Security against Image Segmentation

Content-Based Image Retrieval (CBIR) methods [40] are used for segmentation of an image in regions, identification of regions of interest and extraction of semantic content expressed by the image or part of it. Segmentation process involves use of edge detection or thresholding to segment an image into regions. Other phases of CBIR methods and thus the success of an attack to break our proposed technique highly depend upon the accuracy obtained in the segmentation process. An edge detection technique detects outline of objects in an image by detecting jumps in its image intensity function. Most existing edge detectors are based on first order and second order derivatives of image intensity function. In thresholding, pixels of the image are set to white if their intensity exceeds a certain threshold value otherwise they are set to black. An image histogram graphically displays pixel intensity values of pixels in an image. Background of an image can be separated from its foreground if threshold value(s) clearly separate(s) the two and in this case the image histogram will contain distinct peaks. In case the image histogram does not produce distinct peaks, an adaptive thresholding which changes the intensity threshold for every pixel of the image in relation with the pixel intensity values of its neighboring pixels may be used to separate the foreground from the background.

We tested CAPTCHA images produced by our proposed Image Flip CAPTCHA technique by applying edge detection and thresholding techniques using GIMP on them to investigate the possibility of image segmentation. Figures 6 to 11 show images after applying famous edge detection techniques to the images produced with Image Flip CAPTCHA method shown in figures 3 to 5. Intensity Histograms for images produced in Image Flip CAPTCH are shown in figures 12 to 14 and images obtained after applying thresholding are shown in

Table 1. Comparison of Image CAPTCHA techniques

CAPTCHA Technique	Key Characteristics
IMAGINATION CAPTCHA [25]	• Proposes Image area of 800×600 pixels and total area of 800×700 Pixels • Reported accuracy of 90%. • A random guess probability of 0.000062% with click and annotate phases and 0.44% random guess probability without click phase. • Distortion generated in less than a second and word choice set generation takes about 20 seconds on 450 MHZ Sun Ultra Server. • User response type is click and then selection from a dropdown list.
MosaHIP CAPTCHA [11]	• Proposes image area of 400×400 or 400×275 pixels and total area of 480×485 or 480×377 pixels. • Reported accuracy of 98.4% for total image area of 480×485 and 83.5% for total image area of 480×377 pixels. • A random guess probability of 2.9% and 4.1% for total image areas of 480×485 and 480×377 respectively. • User response type is drag and drop. • User response time is respectively 7 seconds and 10 seconds for concept based and top most bases approaches.
Google Image Orientation CAPTCHA [30]	• Proposes image area of [(180×180)× No. of Image] pixels and total area of approximately 500×250 pixels for three image based tests. • Reported accuracy of 84% for three image based tests and 94.4% for one image based tests. • A random guess probability of 0.009% with 16 degrees of upright orientation for three image based tests and 4.44% for one image based tests. • User response type is image rotation using slider or mouse movement or up-down control. • Position of the images are static and as reported only 68.75% users have preferred it over text based CAPTCHA systems.
Microsoft Assira CAPTCHA [24]	• Proposes image area of 350×166 pixels and total area of 450×200 pixels. • Reported accuracy of 98.5%. • A random guess probability of 0.39% and 0.024% for 8 and 12 image based tests respectively and without partial credit algorithm. • User response type is mouse clicks. • User response time is 15 seconds for 12 image tests.
Image Flip CAPTCHA	• Proposes any image area between 240×180 pixels and 480×360 pixels with two to four non-flipped sub-images. • Accuracy of 96.5% and 93.70% has been obtained for image areas of 240×180 pixels with two and four non flipped sub-images respectively. Accuracy of 98.5% and 97.50% has been obtained for image size of 480×360 pixels with two and four non flipped sub-images respectively. • The calculated random guess probability for image size of 240×180 pixels is 0.35% and 0.0016% for two and four non-flipped images respectively. This probability decreases to 0.021% and 0.000005% for an image area of 480×360 pixels with two and four non-flipped images respectively. • It takes less than a second to generate the CAPTCHA image of 240×180 pixels size. • User response type is mouse clicks. • User response time for CAPTCHA image of 240×180 pixels size is 9.5 seconds having two non-flipped image and 15.5 seconds for image having 4 non-flipped sub-images. This response time decreases to 7.3 seconds and 10 seconds for a CAPTCHA image of 480×360 pixels. Response time can be further decreased by reducing the degree of distortion.

Figure 9. Difference of Gaussians Edge Detection (Image 1)

Figure 10. Difference of Gaussians Edge Detection (Image 2)

Figure 11. Difference of Gaussians Edge Detection (Image 3)

figures 15 to 17. The application of edge detection techniques have extracted some features of each sub image but the presence of complex background and randomly added object resulted in images of inter-connected components. The presence of complex background or clutter impose severe challenges

Figure 12. Image Histogram (Image 1)

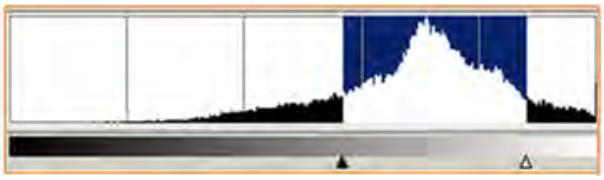

Figure 13. Image Histogram (Image 2)

Figure 14. Image Histogram (Image 3)

Figure 15. Thresholding (Image 1)

Figure 16. Thresholding (Image 2)

Figure 17. Thresholding (Image 3)

Figure 18. Edge Detection EDISON (Image 2)

to perform visual concept detection and identification [41] and thus a successful segmentation is not ordinarily possible. Applications of edge detection and image segmentation using EDISON algorithm (http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html) has also resulted in images of inseparable interconnected content. Synergistic image segmentation using EDISON algorithm on varying gradient window radius, mixture parameter and edge strength threshold was performed on CAPTCHA image of our proposed technique for possibility of edge detection and segmentation. The achieved edge detection with 95% rank and 90% confidence and segmentation images are shown in figures 18 and 19. Confidence map, gradient map, weight map for the CAPTCHA image shown in Figure 4 are shown in figures 20 to 22. Thresholding has not been able to properly segment the image generated with proposed Image Flip CAPTCHA technique. Removal of background information from the CAPTCHA image also removed required information about sub-images. These tests revealed that the placement of some noise in the form of lines, arcs and rectangles in the foreground make both adaptive and simple thresholding to fail. Further, as per the recent research reported in [42], CBIR has proved extremely difficult because of the inherent problems in proper automated analysis and feature extraction of the image to facilitate efficient retrieval especially when the image contains more than one object. The proposed technique as such is secure against possible attacks based on image segmentation.

Figure 19. Segmentation EDISON (Image 2)

Figure 20. Confidence Map (Image 2)

4.2 Security against Shape Matching

Shape matching [43], though a complex and time-consuming process may be used to break an image CAPTCHA technique that has a finite set of images in its static image database and its algorithm does not make transformations to sub-images before placing them on the CAPTCHA image. Shape matching involves image collection of previously presented images, restoring these images to their original form and then comparing them with images presented in the subsequent tests to reveal the difference between

Figure 21. Gradient Map (Image 2)

Figure 22. Weight Map (Image 2)

them. Our proposed Image Flip CAPTCHA technique is secure against shape matching attacks owing to the following reasons: Firstly, as has been discussed in sub-section 4.1, successful image segmentation is very difficult and thus it is not possible to collect images which are required for shape matching. Secondly, even if sub-images are collected through tedious and time-consuming human intervention, the chances of a successful match are minimum because of the use of transformation like scaling and transparency applied by our algorithm to sub-images before placing them on the CAPTCHA image. Thirdly, the application of adding various random objects including arcs, lines, filled and unfilled rectangles at random positions add different distortions to each sub-image. Finally, even without consideration of transformations applied to sub-images and the possibility of manual collection of previously presented sub-images, probability of generation of two similar or same CAPTCHA images generated by our proposed algorithm is very low. Assuming an image database to contain as few as 1000 images with 8 sub-images included in a CAPTCHA image, the total number of unique CAPTCHA images that can be generated by our proposed algorithm is about 10^{10}. On an average, the manual collection will require approximately 0.5×10^{10} CAPTCHA tests to collect all the 1000 sub-images stored in the image database. Further, it is proposed that the image database be periodically updated by including new images and removing the exposed images.

4.3 Security against Random Guessing

A simple possible attack to the proposed technique involves random guessing wherein an attacker may click randomly on any portion(s) of the image. This possibility can be calculated by calculating conditional probability of making successful random guessing. Consider that $A_{C_{img}}$ and $A_{S_{img}}$ are respectively the areas of CAPTCHA image and each sub-image. The probability that the first non-flipped image is correctly guessed $P(SelNF_{imgs}1)$ is thus given by:

$$P(SelNF_{imgs}1) = \frac{A_{S_{img}}}{A_{C_{img}}}$$

The probability that second non-flipped image is correctly guessed with given first non-flipped image (conditional probability) $P(SelNF_{imgs}2\,given\,SelNF_{imgs}1)$ is:

$$P(SelNF_{imgs}2\,given\,SelNF_{imgs}1) = \frac{A_{S_{img}}}{A_{C_{img}} - A_{S_{img}}}$$

The probability that both non-flipped images will be correctly guessed $P(P(SelNF_{imgs}1)\ and\ P(SelNF_{imgs}2))$ is:

$P(P(SelNF_{imgs}1)\ and\ P(SelNF_{imgs}2)) =$
$P(SelNF_{imgs}1)\times P(SelNF_{imgs}2\,given\,SelNF_{imgs}1)$

For our CAPTCHA image, the sub images S_{img} have been chosen with dimensions ranging from 45×45 to 55×55 pixels giving an average area of 2500 pixels. The CAPTCHA image C_{img} has 240×180 pixels dimension, thus having an area of 43200 pixels. Using the formulas listed above the following results have been obtained:

$$P(SelNF_{imgs}1) = 0.0578$$

$$P(SelNF_{imgs}2\ given\ SelNF_{imgs}1) = 0.0614$$

$$P(P(SelNF_{imgs}1)\ and\ P(SelNF_{imgs}2)) = 0.0035$$

Thus the estimated probability that a brute force guessing is successful with the above given dimensions of CAPTCHA and the sub images is only 0.35%. Likewise the probabilities for other sizes i.e. 300×225 pixels and 360×270 pixels which are near to the optimum size are 0.20% and 0.18% respectively. This probability is based on only two non-flipped images. In case the CAPTCHA image has more than two non-

flipped images, chances of brute force guessing are further reduced. The probability of random guessing for optimum CAPTCHA image dimension i.e. 240×180 pixels and near optimum CAPTCHA image dimensions i.e. 300×225 pixels and 360×270 pixels containing 4 non-flipped images is 0.001629%, 0.00023% and 0.000051% respectively. Resource metering technique suggested in [2] and CAPTCHA token buckets suggested in [24] can be used to further reduce the chances of random guessing attacks without causing Denial of Services (DoS).

4.4 Security against Farming-Out

Proxy through unaware users [6] may be used by an attacker to break the CAPTCHA test. In this type of attack, attackers may download the CAPTCHA image and present it to unaware users on unrelated Web sites controlled by the attackers. As an example, an attacker may control a large network of pornographic or other similar Web sites where a visitor would be prompted with a CAPTCHA test to access the content. The user, not aware of the underlying mistrust, solves the CAPTCHA challenge. Thus the obtained response is sent by the Web site to the attacker who may try to use the received response along with other asked properties to gain access to the protected Web service. Making the CAPTCHA image meaningful only in the specific context of the Web site that is protected will make CAPTCHA image not fungible and thus secure against such an attack [29]. Invalidating the CAPTCHA image after a specific time can also be used as a solution against such a threat [44]. Further, since the CAPTCHA image generated by our proposed technique is dynamic, the solution obtained through proxy for a particular proxied CAPTCHA image cannot break the proposed scheme as the chances of regenerating the same CAPTCHA image as explained in section 4.2 are very low.

4.5 Security against other attacks

In addition to the above discussed possible attacks, database attack [24] may be carried out on image-based CAPTCHA technique. Database attack involves the creation of the entire image database. A portion of the image database is revealed each time a challenge is displayed and thus by solving enough challenges, the entire database is eventually revealed. Database attack is similar to shape-matching and involves image collection, restoration and comparison. The proposed technique is secure against database attack for the reasons stated in 4.2. Further, to make the underlying image database secure against a hacker who may succeed to sneak into the system; the images and the corresponding tags associated with each image can be stored in separate tables which are related with each other on an encoded key within a single database.

Further, a weak Implementation [24] of any CAPTCHA technique like allowing a session ID authorized by a single successful challenge to be re-used repeatedly to gain access to some protected service makes that CAPTCHA technique insecure. Insecurity on account of weak Implementation of CAPTCHA methods can only be eliminated by careful analysis, code reviews and timely updates.

5 Usability Analysis

ISO 9241 (titled as Ergonomics of Human System Interaction) defines usability as a measure of the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in a particular environment. Jeff Yen and Ahmad Salar El Ahmad in their recent study [45] have provided a three dimensional framework for examining and improving the usability of OCR-based CAPTCHAs. Under each dimension namely distortion, content and presentation, various usability issues have been identified and explained. This framework with slight modification in usability issues under each dimension is applicable to image-based CAPTCHAs as well. In this section, we present the usability analysis carried out on our proposed Image Flip CAPTCHA technique under the guidelines of the above mentioned framework.

5.1 Usability issues under Distortion Dimension

Distortion is used in CAPTCHAs to improve its security control; however the use of excessive or unmanaged distortion level and methods may not only make CAPTCHAs unusable but also will lower its security control because the system would have to allow multiple attempts for failed tests [45]. Distortions also create ambiguous characters, hard to tell apart from each other and identify and thus are unfriendly to foreigners who are not native speakers of the language in which CAPTCHA is implemented.

In the proposed Image Flip CAPTCHA technique, the application of various distortions to each sub-image affects the visibility of the CAPTCHA image. However, in this technique a user needs not to understand the image fully and instead has only to identify its place within the CAPTCHA image and has to recognize any flip in it, a reasonable distortion in its visibility does not affect a user to make the correct response. This has been observed through usability tests carried out on the proposed technique which are

detailed in section 6. The identification of the sub-images and their flip can further be made easy by showing a current portion of the CAPTCHA image in zoomed-in state when the user hovers over that portion of the image. The recently proposed Partial Credit Algorithm [24], in which "almost right" answers are treated as strong evidence that a user is human can be used to improve usability against complex distortions. Further, the maximum and minimum allowable distortion levels for each sub-image can be automatically controlled by keeping track of images that legitimate users have failed to recognize. In this case algorithm has to be modified and appropriate fields need to be added in the image database. Since our proposed CAPTCHA technique is an image-based clickable CAPTCHA and does not require the user to read any messages or type a response, it is as friendly to foreigners as it is to native users.

5.2 Usability issues under Content Dimension

Various usability issues falling in the content dimension for image-based CAPTCHAs are issues pertaining to the images in its image database. They are number, size and type of images and offensive or unsolicited images. Having a large number of images with small size, image database will reduce chances of successful brute force guessing and thus improve security but at the same time server processing and difficulty in recognizing images increases and thus reduce usability. Any unsolicited or offensive image appearing in the CAPTCHA test will considerably reduce its usability. Further, use of well known images i.e. images which can be recognized easily by most of the users will improve usability.

In the proposed CAPTCHA technique, the use of small-sized images of objects which are small in size or use of photographs of people in reduced sizes results in a compact image database and thus image retrieval from the image database and their further processing is fast. The use of small sub-images in CAPTCHA image will not deteriorate the usability owing to the reason that human users can recognize images of objects that are themselves small easily than the images of large objects in smaller sizes. Use of photographs of people in reduced sizes will also have less impact on the usability because of the fact that users are only asked to recognize any flip in the photograph. Unlike OCR-based CAPTCHAs wherein offensive words may be generated as a result of random selection of letters from the available character set [45], image CAPTCHAs do not suffer from this problem as long as image database does not contain any offensive or unsolicited image.

5.3 Usability issues under Presentation Dimension

A CAPTCHA test can alienate or even frustrate a legitimate user if its presentation is poor. To improve the presentation of a CAPTCHA test various usability issues [23, 24, 45, 46] that must be addressed while designing a CAPTCHA challenge are use of color, user interface and appropriate screen area so as to make the challenge simple, easy to answer, easy to integrate with the Web page and highly accurate.

Use of color enhances the user interface but its misuse can cause both usability and security problems [45]. Research work carried out in [45] reported effective segmentation of overlapped characters generated through Cryptographp CAPTCHA (see http://www.cryptographp.com) by picking up pixels from the CAPTCHA image having same color. The proposed Image Flip CAPTCHA technique uses colored images thus making it colorful which is a desired property and improve usability. The distribution of colors in this proposed technique is random because pictures have a wide variety of backgrounds, angles, poses and lightings. These factors do not permit easy classification by colors.

A CAPTCHA user interface may require a user to input response by typing characters in a text box or by selecting the answers from a dropdown list or by clicking on correct portion(s) of the CAPTCHA image. Most of the existing CAPTCHA challenges particularly OCR-based use text box to input responses from the users, which in comparison with other user interfaces besides being difficult to work with is time consuming. A clickable interface simplifies and speeds up the entry of the CAPTCHA solutions which improves user friendliness and permits the use of CAPTCHA on devices with small displays where they would otherwise be unusable [23]. The proposed Image Flip CAPTCHA technique has a clickable interface. Being clickable it is simple to work with and the process of entry of response is fast. As reported in [23] clickable CAPTCHA can be solved faster than textual CAPTCHAs on mobile devices, our proposed technique has a desired property of being usable for mobile devices. Due to its improved usability, it may also be used in other Web defenses that include defense against Click Fraud [23].

An optimum size of CAPTCHA image is highly desired to have a balance between usability and security [24]. Large dimensions reduce the chances of successful blind attacks and thus improve security control. It also fastens the response time due to improved visibility of the sub-images within the CAPTCHA image. However, large images involve huge server processing for applying transformations and for transferring

the images from the image database which decrease the performance. On the other side, smaller images offer advantages of faster image downloads and occupy less screen area making it easy to integrate the CAPTCHA challenge within the Web page. We tested several CAPTCHA image sizes detailed in section 6 to find out the best usable and secure image size. Image sizes 240×180 pixels and immediate higher sizes have been found optimum for the proposed image flip CAPTCHA technique. The areas required by these sized images are much smaller than that required in most other reported image CAPTCHA methods. A comparison of screen areas required in various CAPTCHA techniques is provided in Figure 23.

6 User Study and Experimental Results

In this section, we summarize the results obtained through several user studies and experiments. CAPTCHA images of different sizes were shown to 225 users whose expertise and skill in using Internet varied considerably. Sample user studies by implementing the proposed technique through programming not fully optimized for real use and with as few as 225 users may not precisely determine its security and usability. However, it may greatly help with approximating its performance and usability for practical use. Experiments for determining the success of possible brute force guessing were carried out by putting the CAPTCHA images of various sizes to test attacks from a sample bot program. Besides simulating an ordinary blind attack, we also simulated an intelligent blind attack to identify the correct position of non-flipped images. In this experiment we first applied image segmentation to the CAPTCHA image and then restricted the blind guess to regions outlined by the segmentation process. We categorize various aspects of the proposed technique on data obtained through user studies and experiments.

We tested various image sizes from 10800 pixels to 172800 pixels to find out the best usable and secure image size for our proposed CAPTCHA technique. We studied the use of two and four non-flipped sub-images in the CAPTCHA image. Figure 24 plots percentage accuracy vs. CAPTCHA image size. The results showed that more number of users were accurate in responding to CAPTCHA images containing two non-flipped sub-images in comparison with CAPTCHA images containing four non-flipped sub-images. This difference in accuracy is high for smaller sizes of CAPTCHA image in comparison with that for CAPTCHA images of larger sizes. The rate of accuracy improved with increase in CAPTCHA image size but image sizes larger than 240×180 pixels showed no or little improvement. The overall accuracy rate remained nearly 96.5% for CAPTCHA image of 240×180 pixels containing four non-flipped sub-images.

We also used the data obtained through user studies to evaluate the effect of CAPTCHA image size on response time. The results are shown in Figure 25 wherein, we plot the median response time against CAPTCHA image size. Users responded faster to CAPTCHA images containing two non-flipped sub-images than they did to those containing four non-flipped sub-images. Smaller CAPTCHA images caused degradation in response time which improved for larger sizes. On average, users took 9.5 seconds to solve CAPTCHA image challenge of 240×180 pixels size containing 2 non-flipped sub-images.

The result of simulating an intelligent blind attack which used segmentation algorithm to restrict blind guess area to randomly identify the correct positions of non-flipped images is shown in Figure 26. Results have shown that success rate of random guessing has remained very low but higher than the calculated values especially for images of smaller sizes. The average successful blind guess has remained around 0.65%. In comparison with the ordinary blind attack, the success rate of this intelligent blind attack has remained high. The results plotted in Figure 26 are for CAPTCHA images generated by our algorithm with very low level of distortion applied to them.

Among various possible candidates for best image size, the size of 240×180 pixels was chosen for further analysis because it is easier to integrate small sized images in Web pages and thus improving its usability. Further, sizes above 240×180 pixels showed little improvement in response time, accuracy and robustness. Users were shown CAPTCHA images of 240×180 pixels with different distortions applied to them. Three different distortion levels as detailed in Table 2 were used. Average time to create CAPTCHA image at the server with different distortion levels on a PC with 2 GHZ processor and 1 GB RAM is also shown in Table 2. The time to create CAPTCHA image will vary significantly from one system to another as it depends on number of factors that include processing power of the system used to generate the image and number and size of images in the image database. The result of application of different distortion levels as detailed in Table 2 on accuracy and response time are shown in figures 27 and 28. Accuracy decreased by about 2% from low to medium distortion levels and by a further 6% for high distortion level. Higher distortion levels deteriorated the response time especially for CAPTCHA images containing four non-flipped sub-

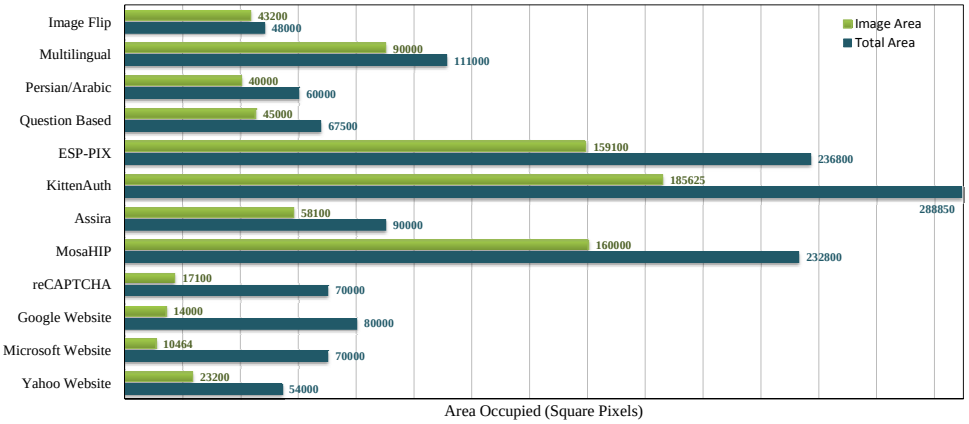

Figure 23. Comparison of Screen Area required in various CAPTCHA Techniques

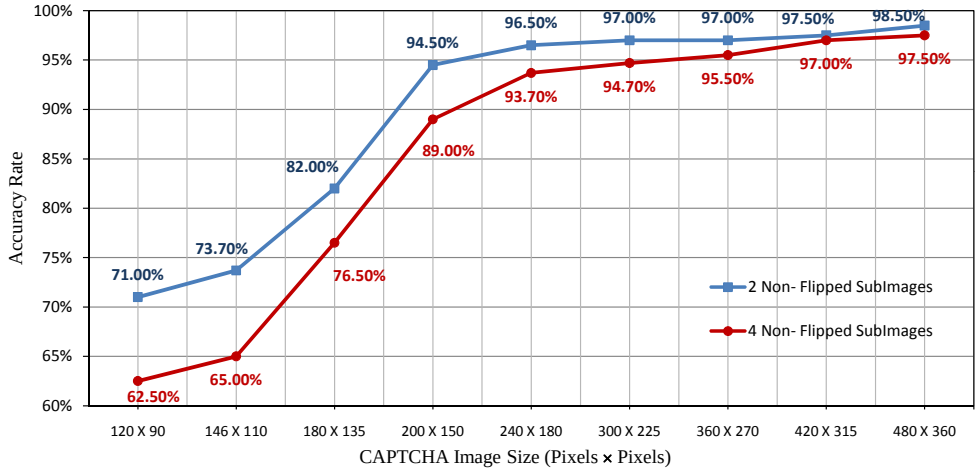

Figure 24. Classification accuracy achieved by users in Image Flip CAPTCHA Technique vs. Image Area

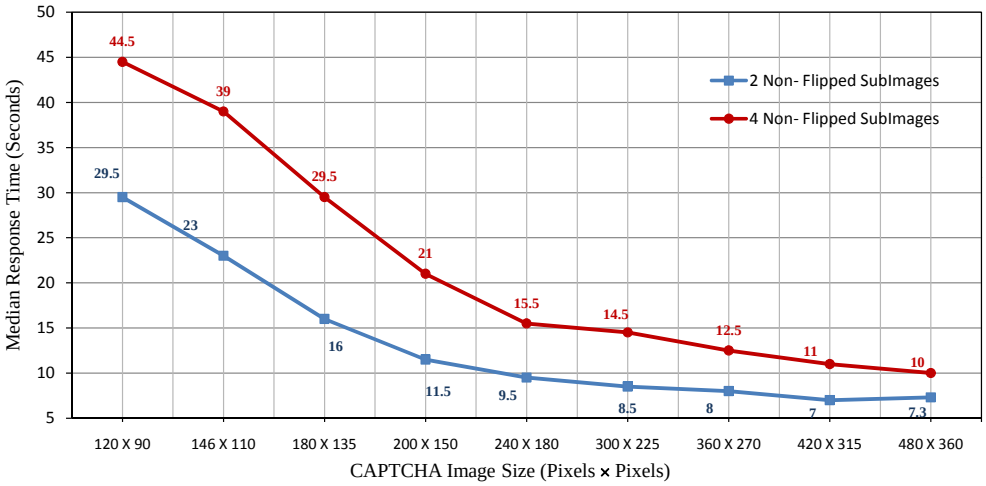

Figure 25. Time required by users to qualify CAPTCHA test vs. Image Area

Table 2. Distortion Levels and Time to create CAPTCHA image for various distortion levels

Distortion Level	Transparency Factor for Sub-Images	Scaling Factor for Sub-Images	No. of Random Objects Added	Average Time to Create Image
Low	81%-90%	None	200	940ms
Medium	61%-80%	81%-90%	300	1085ms
High	40%-60%	70%-80%	500	1167ms

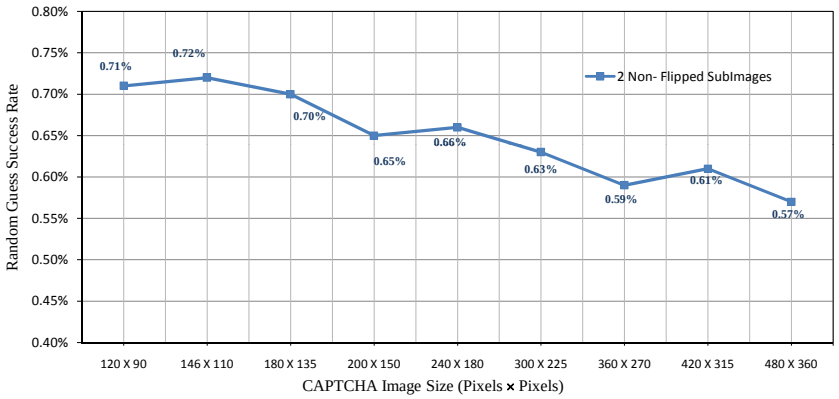

Figure 26. Random Guess Success Rate vs. CAPTCHA Image Area

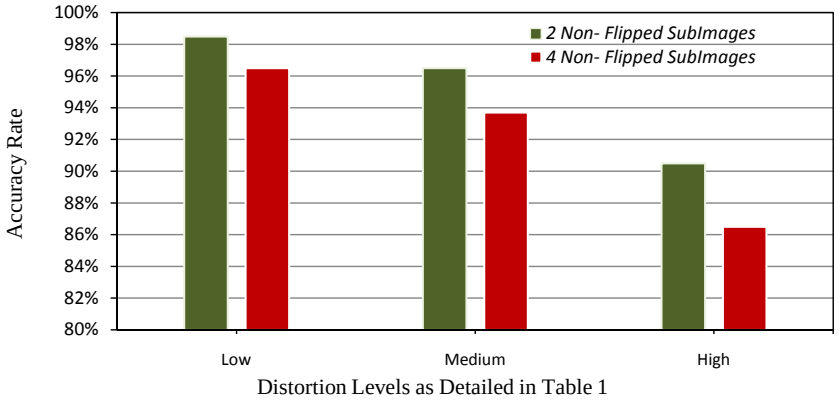

Figure 27. Effect of various distortion levels on accuracy (Image Size of 240×180 pixels)

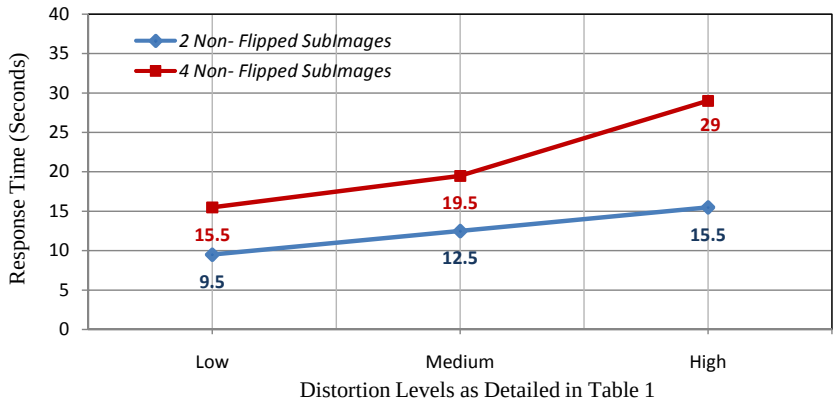

Figure 28. Effect of various distortion levels on Response Time (Image Size of 240×180 pixels)

images.

7 Conclusion

In this paper a new clickable image-based CAPTCHA technique is presented. The technique besides being simple to generate is resistant to attacks from automated Web tools and is highly usable. It uses small screen area in comparison with that used in other image-based CAPTCHA techniques. The paper presents possible approaches that may be tried to defeat the proposed CAPTCHA technique and discusses its security control against farming-out attacks and attacks that involve use of image segmentation, shape matching and random guessing. The paper presents usability issues of image-based CAPTCHA techniques and analyzes the proposed CAPTCHA technique under various usability issues pertaining to distortion, content and presentation dimensions. The paper further categorized various aspects of the proposed technique that include image size, accuracy, response time, time for image generation, distortion and robustness against attacks on data obtained through user studies and experiments. The results obtained have validated the efficiency of proposed CAPTCHA technique. Thus an endeavor has been made in the development of efficient image-based CAPTCHA technique.

References

[1] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Proceedings of the 6th ACM SIGCOMM on Internet Measurement (IMC'06)*, pages 41–52, Rio de Janeriro, Brazil, 2006. ACM.

[2] G. Ollmann. Stopping Automated Attack Tools, 2005. Available online at: http://www.ngssoftware.com/papers/StoppingAutomatedAttackTools.pdf, Accessed 25, Aug 2008.

[3] M. D. Vivo, G. O. D. Vivo, R. Koeneke, and G. Isern. Internet Vulnerabilities Related to TCP/IP and T/TCP. *SIGCOMM Computer Communications Review*, 29(1):81–85, 1999.

[4] H. Baird and K. Popat. Human Interactive Proofs and Document Image Analysis. In *Proceedings of the 5th IAPR International Workshop on Document Analysis Systems (DAS'02)*, volume 2423 of *Lecture Notes in Computer Science (LNCS)*, pages 531–537, Princeton, NJ, USA, 2002. Springer.

[5] N. J. Hopper and M. Blum. Secure Human Identification Protocols. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01)*, volume 2248 of *Lecture Notes in Computer Science (LNCS)*, pages 52–66, Gold Coast, Australia, 2001. Springer.

[6] C. Pope and K. Kaur. Is It Human or Computer? Defending E-Commerce with Captchas. *IEEE IT Professional*, 7(2):43–49, 2005.

[7] S. Shirali-Shahreza and A. Movaghar. A New Anti-Spam Protocol Using CAPTCHA. In *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, pages 234–238, London, UK, 2007.

[8] R. Dhamija and J. D. Tygar. Phish and HIPs: Human Interactive Proofs to Detect Phishing Attacks. In *Proceedings of the 2nd International Workshop on Human Interactive Proofs (HIP'05)*, pages 127–141, Bethlehem, PA, USA, 2005. Springer.

[9] R. P. Karrer. EC: An Edge-Based Architecture Against DDoS Attacks and Malware Spread. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)*, volume 2, pages 49–56, Vienna, Austria, 2006.

[10] W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, and D. Rubenstein. Using Graphic Turing Tests to Counter Automated DDoS Attacks Against Web Servers. In *Proceedings of the ACM Conference on Computer and Communication Security*, pages 8–19, Washington D.C., USA, 2003.

[11] A. Basso. *Multimedia Content Protection from Massive Automated Access and Unauthorized Distribution*. PhD thesis, 2008.

[12] I. Fisher and T. Herfet. Visual CAPTCHA for Document Authentication. In *Proceedings of the IEEE 8th Workshop on Multimedia Signal Processing*, pages 471–474, Victoria, BC, Canada., 2006.

[13] M. D. Lillibridge, M. Abadi, K. Bharat, and A. Z. Broder. Method for Selectively Restricting Access to Computer Systems, 2001.

[14] L. von Ahn, M. Blum, and J. Langford. Telling Humans and Computers Apart Automatically. *Communications of the ACM*, 47(2):57–60, 2004.

[15] G. Mori and J. Malik. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 134–141, Madison, USA, 2003.

[16] L. V. Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science Express*, 321(5895):1465–

1468, 2008.

[17] M. Chew and H.S. Baird. BaffleText, a Human Interaction Proof. In *Proceedings of the 10th SPIE/IS&T Document Recognition and Retrieval Conference (DRR'03)*, pages 305–316, Santa Clara, CA, USA, 2003.

[18] M. Chew and J. D. Tygar. Image Recognition CAPTCHAs. In *Proceedings of the 7th International Information Security Conference (ISC 2004)*. Springer, 2004.

[19] A. Rusu and V. Govindaraju. Handwritten CAPTCHA: Using the Difference in the Abilities of Humans and Machines in Reading Handwritten Words. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR- 9 2004)*, pages 226–231, Kokubunji, Tokyo, Japan, 2004.

[20] R. Ferzli, R. Bazzi, and L. J. Karam. A CAPTCHA Based on the Human Visual Systems Masking Characterists. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo (ICME'06)*, pages 517–520, Toronto, Ontario, Canada, 2006.

[21] M. H. Shirali-Shahreza and M. Shirali-Shahreza. Persian/Arabic Baffletext CAPTCHA. *Journal of Universal Computer Science*, 12(12):1783–1796, 2006.

[22] M. H. Shirali-Shahreza and M. Shirali-Shahreza. Question-Based CAPTCHA. In *Proceedings of International Conference on Computational Intelligence and Multimedia Applications*, volume 4, pages 54–58, Sivakasi, Tamil Nadu, India, 2007.

[23] R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang. Making CAPTCHAs Clickable. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, pages 91–94, Napa Valley, CA, USA, 2008.

[24] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul. Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*, pages 366–374, Alexandria, Virginia, USA, 2007. ACM.

[25] R. Datta, J. Li, and J. Z. Wang. Imagination: A Robust Image-Based CAPTCHA Generation System. In *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA05)*, pages 331–334, New York, NY, USA, 2005. ACM Press.

[26] W. H. Liao. A CAPTCHA Mechanism by Exchanging Image Blocks. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR06)*, volume 1, pages 1179–1183, Hong Kong, 2006.

[27] D. Misra and K. Gaj. Face Recognition CAPTCHAs. In *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW'06)*, pages 122–127, Guadeloupe, French Caribbean, 2006.

[28] M. Shirali-Shahreza and S. Shirali-Shahreza. Multilingual CAPTCHA. In *Proceedings of the 5th IEEE International Conference on Computational Cybernetics (ICCC'07)*, pages 135–139, Gammarth, Tunisia, 2007.

[29] H. S. Baird and J. L. Bentley. Implicit CAPTCHAs. In *Proceedings of the IS&T/SPIE Document Recognition & Retrieval XII Conference*, pages 191–196, San Jose, CA, USA, 2005.

[30] R. Gossweiler, M. Kamvar, and S. Baluja. Whats Up CAPTCHA? A CAPTCHA Based On Image Orientation. In *Proceedings of the 18th International Conference on World Wide Web*, pages 841–850, Madrid, Spain, 2009.

[31] T.Y Chan. Using a Text-to-Speech Synthesizer to Generate a Reverse Turing Test. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 226–232, Sacramento, CA, USA, 2003.

[32] M. Chew and J. D. Tygar. Collaborative Filtering CAPTCHAs. In *Proceedings of the 2nd International Workshop on Human Interactive Proofs (HIP'05)*, pages 66–81, Bethlehem, PA, USA, 2005. Springer.

[33] J. Yen and A. S. Ahmad. A Low-cost Attack on Microsoft CAPTCHA. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 543–554, Alexandria, Virginia, USA, 2008.

[34] J. Yen and A. S. Ahmad. Breaking Visual CAPTCHAs with Nave Pattern Recognition Algorithms. In *Proceedings of the 23rd Annual Computer Security Applications Conference*, pages 279–291, Miami Beach, Florida, USA, 2007. IEEE Computer Society.

[35] G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion Estimation Techniques in Solving Visual CAPTCHAs. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume 2, pages 23–28, Washington, DC, USA, 2004.

[36] K. Chellapilla and P. Y. Simard. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'04)*, pages 265–272, Cambridge, MA, 2005. MIT Press.

[37] K. Chellapilla, P. Simard, and M. Czerwinski.

Computers Beat Humans at Single Character Recognition in Reading-Based Human Interaction Proofs (HIPs). In *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS)*, Palo Alto, CA, USA, 2005.

[38] GIMP. GNU Image Manipulation Program. Available at http://www.gimp.org, Accessed 25, Aug 2008.

[39] K. Yanai, M. Shindo, and K. Noshita. A Fast Image Gathering System from the World-Wide Web Using a PC Cluster. *Image and Vision Computing*, 22(1):59–71, 2004.

[40] F. Long, H. Zhang, and D. Feng. *Fundamentals of Content-Based Image Retrieval (Chapter 1)*. Springer, 2003.

[41] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-Based Multimedia Information Retrieval: State of the Art and Challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2 (1):1–19, 2006.

[42] S. Deb. Overview of Image Segmentation Techniques and Searching for Future Directions of Research in Content-Based Image Retrieval. In *Proceedings of the 1st IEEE International Conference on Ubi-Media Computing and Workshops*, pages 184–189, Lanzhou, China, 2008.

[43] M. Yang, K. Kidiyo, and R. Joseph. Shape Matching and Object Recognition Using Chord Contexts. In *Proceedings of the International Conference on Visualizations*, pages 63–69, London, UK, 2008.

[44] R. Jones and K. L. Klinker. Beyond the Session Timeout: Automatic Hierarchical Segementation of Search Topics in Query Logs. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 699–708, Napa Valley, California, USA, 2008.

[45] J. Yan and A. S. Ahmad. Usability of CAPTCHAs or Usability Issues in CAPTCHA Design. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, pages 44–52, Pittsburgh, PA, USA, 2008.

[46] G. Sauer, H. Hochheiser, J. Feng, and J. Lazar. Towards a Universally Usable CAPTCHA. In *Proceedings of the Symposium on Accessible Privacy and Security, ACM Symposium On Usable Privacy and Security (SOUPS'08)*, Pittsburgh, PA, USA, 2008.

M. Tariq Banday was born in 1969. He did his M. Sc. and M. Phil. Degrees from the department of Electronics, University of Kashmir, Srinagar, India in 1996 and 2008 respectively. At present he is working as Assistant Professor in the Department of Electronics & Instrumentation Technology, University of Kashmir, Srinagar, India. He has to his credit several research publications in reputed journals and conferences. His current research interests include Network Security, Internet Protocols and Network Architecture.

Nisar A. Shah was born in 1953. He did his M. Sc. and Ph. D. Degrees from the department of Physics, University of Kashmir, Srinagar, India in 1976 and 1981 respectively. At present he is working as Professor in the Department of Electronics & Instrumentation Technology, University of Kashmir. He has to his credit about 150 research publications which have been published in national and international journals of repute. He has supervised several research scholars in M. Phil. and Ph. D. programs. His current research interests include Digital Signal Processing and Network Security.