Cursor CAPTCHA – Captcha Mechanism using Mouse Cursor

Varun Ambrose Thomas
Student
Lovely Professional University
Phagwara, Punjab

Karanvir Kaur
Assistant Professor
Lovely Professional University
Phagwara, Punjab

ABSTRACT
With the growing and prevalent usage of Internet, threat for the service providers has arisen. One of the most important aspects of modern security concerns is to deal with users that are computer programs or bots. These programs pretend to be human users and exploit servers by submitting data automatically, thereby, hindering the services of the server. Security mechanism dealing with such attacks is called Completely Automated Turing test to tell Computers and Humans Apart (CAPTCHA). This paper discusses and analyzes broad CAPTCHA mechanisms and recent researches that have either reformed or flunked, previous techniques and methods. Also, a technique is proposed that counteracts the shortcomings or inabilities of such techniques.

General Terms
CAPTCHA, Algorithm, Cursor Matching

Keywords
CAPTCHA, Cursor – Match, Image CAPTCHA

1. INTRODUCTION

1.1 Definition
The Internet is the most prevalent and popular form of providing service, and most organizations whether private or government, rely and prefer taking data to and from the people using Internet forms (where user submit their data and other details). Examples of such forms are Online Examination Form submission of a private or government college/institution, Gmail – providing free e-mail accesses to people etc. What people, henceforth users, do is first they have to register on websites by providing their details before using the services of the organization via Internet.

Recently, computer programs are being developed to automatically fill these details over the Internet through direct requests at high speeds, consequently obstructing the server to fulfill other requests, generally referred to as Denial of Service (DoS) attacks. In 2000, a technique, CAPTCHA [1] was introduced. It is an acronym for Completely Automated Public Turing Test to tell Computers and Humans Apart. It enables services to find out whether the user accessing the server is a human or computer program by challenging user with a test that is designed to be solvable only a human and not at all or very difficult for an automated computer program to solve.

Application of CAPTCHA lies in registering on websites, Blog comments and, typically in account creation and login applications where the chances of unauthentic usage are high. The main aspects of any CAPTCHA techniques [2] are:

- Easy to generate
- Easily solvable by humans
- Impossible or very difficult for computer programs to solve

So, any technique developed has to satisfy these basic aspects to be feasible.

1.2 Types of CAPTCHAs
A broad classification of CAPTCHA mechanisms [3] is described as:

1.2.1 Text based CAPTCHA
These techniques generate a random sequence of characters and embed them in an image that is highly distorted and designed in an unrecognizable (by computer programs) form so that only humans can identify the characters embedded in the image (see Figure 1). The user has to identify the characters and write them in the place provided and submit. The user is justified as a human when the characters match with that used by server and CAPTCHA generator system. Though many techniques involve use of one image only, a technique [3] makes use of set of 12 images (see Figure 3) and asks user to identify 3 images that contain English language word by just clicking onto them instead of providing the contents of the images as the answer.

Figure 1: Example of Text captcha

1.2.2 Image – based CAPTCHAs
Also known as Image Recognition CAPTCHA (IRC), because most of the techniques involve users in identifying, either the nature or type of the image(s) being displayed and then respond with the correct answer. Latest example of such techniques is Anomaly CATPCHA [2] (see Figure 4). User has to identify an image that does not belong to the feature/category depicted by rest of the images.

1.2.3 Audio based CAPTCHA
Such techniques require users to listen to noisy audio narrated by a voice that mentions some characters. Users have to identify those characters and submit them to server in order to prove themselves as humans. The voice signal is distorted by

Figure 2: Using operations, Preprocessing, Segmentation, Post – segmentation on eBay CAPTCHA

Figure 3: Example of Clickable CAPTCHA [3]

Figure 4: Example of Image CAPTCHA [2]

additional noise so that the characters are not easily identifiable by artificially intelligent systems or AI attacks.

1.2.4 Video based CAPTCHA

These involve a video to be shown to the user and user has to watch it closely and observe the actions in the video and surmising a word the suits to it most appropriately.

1.2.5 Puzzle based CAPTCHA

In this, two types of techniques are involved [11]. The first is the Mathematical Expression based Turing tests. These tests ask the user to provide an answer to a mathematical question. Example to this can be "What is seventeen plus two plus thirteen minus twenty?"

In another technique, in the category of PBC, a logical question is put to the user to answer. The responses in such technique rely on the I.Q. of the users. Examples from such technique involve questions such as: "What is the name of the fruit that has its own color?" (Orange), or "What is the name of a yellow curved fruit?" (Banana)

The most commonly used techniques are Text and Image based CAPTCHAs as Audio and Video based techniques are typically large in size, uses comparatively more bandwidth and are time consuming especially for re – attempts. The next section describes the features and traits associated with above mentioned categories.

2. RELATED WORK

2.1 Text based CAPTCHA

These techniques are constrained by problems such as:

- Confusing characters: While distorting the characters randomness sequences such l and I, 5 and S, G and 6, etc. impose perceiving problems as shown (see Figure 5) [4].

Figure 5: Example of Text CAPTCHA images with confusing characters sequences

- A technique has been designed [5] that is able to generate similar CAPTCHA images as that generated by eBay, Google and ReCaptcha with accuracies over 90% for each. Also, they have used pre – processing, segmentation and post – segmentation that works till finding out the number of characters as shown (see Figure 2).
- These techniques are vulnerable to Optical Character Recognition attacks as their security is dependable upon the extent of distortion whilst maintaining the ease of ascertaining the characters by human users [6].
- Other factors are such as language dependency because non – English users who do not have idea for how the characters are written or do not know the English alphanumeric characters have problems identifying the characters. It has even been signified that language does affect the effectiveness of a CAPTCHA's implementation for global users [7].

2.2 Image based CAPTCHA

HumanAuth solver fetched the challenge from a website implementing HumanAuth IRC and identified the solution correctly. So it is concluded that IRCs are recognizable. IRCs are also vulnerable to AI attacks [8].

Other factors associated with IRCs that preclude their usage are:

Table 1. Decaptcha's Coverage, per – digit precision and per – captcha precision [12]

Scheme	Len	Coverage	Fast Fourier Transform		Cepstrum		Cepstrum + Mel	
			Digit	Captcha	Digit	Captcha	Digit	Captcha
Authorize	5	100	93.73	80.39	96.08	87.25	97.06	89.22
eBay	6	85.60	81.58	44.36	92.48	82.88	92.61	80.93
Microsoft	10	80.60	76.57	14.69	89.58	48.95	89.30	47.55
Yahoo	7	99.10	33.77	0.00	74.71	45.45	68.13	30.30

- Large database is required to create a challenge because dynamic creation of images from which a user can infer some meaning is not feasible as of today
- A single challenge requires large number of images and, thus, imposes a limit on the repetition of same image for successive challenge
- Sufficiently large images requires so that they can convey a meaning perceivable by Human user

2.3 Audio based CAPTCHA

A study on Audio based techniques [12] proposed that using a simple two phase signal analysis design with DeCAPTCHA breaking the noise – based Audio CAPTCHA was possible with an accuracy significantly above the threshold value of 1% of breaking a CAPTCHA. They also conclude that computers are more resilient towards the noise embedded in the Audio CAPTCHAs than the Humans and can process the signals efficiently when more efficient hardware and technique is used. Table 1 depicts the conclusion that shows the coverage, the fraction of CAPTCHAs solvable by Decaptcha, and the precision is the accuracy of the solver. The columns are the stages in the algorithm, Decaptcha [12].

2.4 Video based CAPTCHA

These techniques provide user with a moving graphic that depict certain visualization that can be expressed in one word. These videos are generally not solvable by foreign users or non – native users of the thing occurring in the video. This imposes hindrance in global usage. Size of the challenge is another drawback of these techniques.

2.5 Puzzle based CAPTCHA

Techniques using mathematical expressions are vulnerable with respect to cognitive disabilities of the users and also a custom calculator can be built by identifying the patterns and presentation way of the questions.

Logical questions witness vulnerability to a dictionary based attack [11]. Also, the shortcoming lies in the limit of the number of such questions that have a unique answer and that too which are easily recallable.

3. PROPOSED MODEL

3.1 Constructs of the Model

The work is done on an Image based technique that does not require image recognition. The technique exploits random numbers to generate the challenges and images to be used as Custom Mouse Cursor. So far, it is yet not possible to save a CMC's image directly on the client side using a client script without making a second request or reference to the URL (Uniform Resource Locator) of the original cursor. The challenge consists of 5 images randomly located on an area over an HTML page confined within a box (challenge panel). The user has to match the CMC by moving the mouse pointer and hover it over the challenge area where a match is found and overlap.

The user then has to click on the image without moving the mouse any further. The user is confirmed to be a Human user if it is able to match the Clicked coordinates to the actual location of the underlying image on the challenge panel. It can be inferred [10] that mouse movements are incontrollable by scripts as far as matching of the images is concerned. Hence, the proposed technique exploits this mechanism.

3.2 Objectives of the Implementation

The proposed technique is undergoing its implementation and is expected to attain the following objectives:

- Language independence: Since our technique is based on images, it is totally language independent and the instructions of solving the CAPTCHA can be provided on their own/native language as the technique does not involve use of language that is changing at runtime.
- Less storage: As compared to other techniques, images in the proposed technique will be small in size, typically of the order of mouse cursors and, hence easy visibility is the target.
- No Image Recognition: Since the proposed mechanism will use image matching mechanism, there is a certain possibility that no IRC attacks possible for obtaining the meaning of images.
- No OCR: Since there are no characters involved, there is expected to be no character recognition. Attacks on recognition will not aid to solve it automatically.

Apart from these four aspects, a comparison (see Table 2) can be made among the existing class of techniques for CAPTCHA and the proposed model on the basis of the following factors: language of the challenge, storage size required beforehand for creating the challenge, vulnerability to OCR and IR attacks, domain of constituting elements in a single challenge, frequency of repetitive challenges, global intellectual dependency (to evaluate global throng of users), single challenge size (data to be transferred to and fro).

3.3 Flowchart

Figure (see Figure 6) displays the flowchart which shows the activities performed by the server side and the user. The first two activities are to be done by the CAPTCHA generator algorithm and answers are previously stored.

The next two activities are inputs provided by User. The results are then reverted back to the CAPTCHA service and the results of User interaction are matched with that stored with the service. The results are calculated by adhering to a

Table 2. Comparisons of existing classes of techniques and proposed model

Factors	Text based	Image Recognition	Audio based	Video based	Puzzle based	***Proposed Model***
Language Dependency	Yes	No	Yes	Depends on Challenge	Yes	*No*
Storage Size for pre – computed data	Very Low	High	Very Low	Very High	High	*Very Low*
Vulnerable to OCR Attacks	Yes	No	No	No	No	*No*
Vulnerable to IR Attacks	No	Yes	No	No	No	*No*
Domain of Elements in Challenges	Limited to 26 Characters and 10 numbers	Very Limited	Limited to 26 Characters and 10 numbers	Very Limited	Very Limited	*Theoretically Unlimited*
Frequency of repetitive Challenges	Very Low	High	Very Low	Very High	Very High	*Very Low*
Intellectual Dependency	Not Required	Required	Not Required	Required	Required	*Not Required*
Single Captcha Challenge Size	Small	Large	Large	Very Large	Small	*Small*

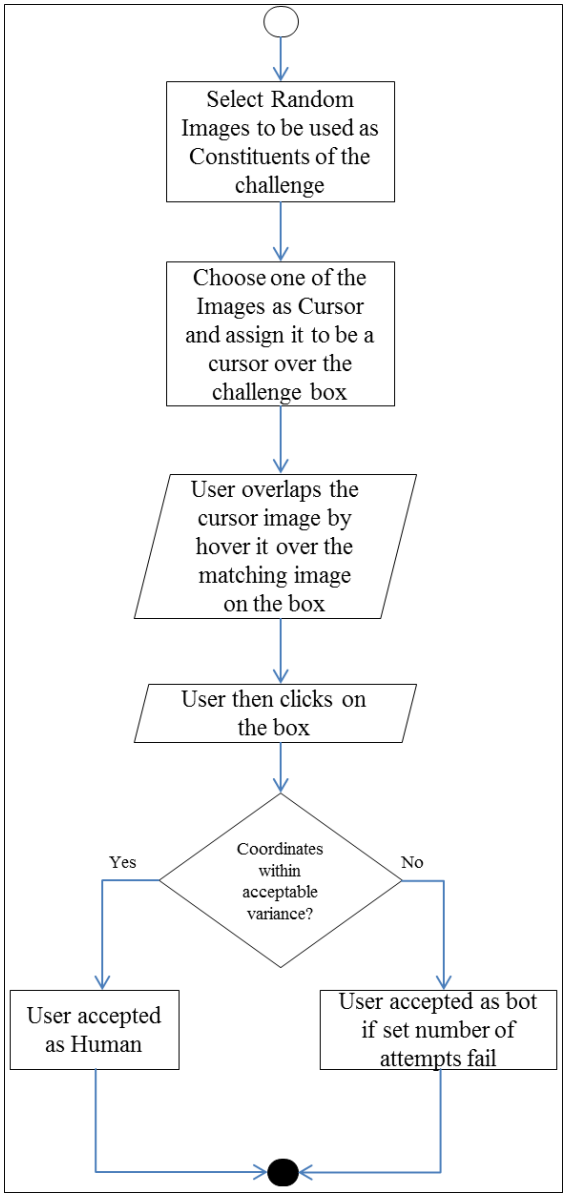

Figure 6: Proposed Cursor CAPTCHA Mechanism

predefined acceptable variance of the coordinates in the results of the clicked position of the user.

The user is authenticated to be a Human user if it completes the challenge within a limited number of attempts or, otherwise, the user is classified as a bot/computer program.

4. CONCLUSION AND FUTURE WORK

The proposed Cursor CAPTCHA technique is modeled to meet the shortcomings of most popular text – based CAPTCHA and Image based CAPTCHA implementation. Future work consists of the implementation of the technique and work has been started on it. Usability of the proposed technique will also be evaluated. Furthermore, work can be extended by implementing the challenge panel as single image by an image creation algorithm. This model can be developed for touchscreen mobile applications.

5. REFERENCES

[1] Ahn, Luis von, Blum, Manuel, Hopper, Nicholas J., Langford, John 2003. CAPTCHA: Using Hard AI Problems for Security. In Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, pp 294-311

[2] Chew, Monica, Tygar, J. D. 2004. Image Recognition CAPTCHAs. In Proceedings of the 7th International Information Security Conference (ISC 2004), Springer, pp. 268-279

[3] Chow, Richard, Golle, Philippe, Jakobsson, Markus, Wang, Lusha, Wang, Xiaofeng 2008. Making CAPTCHAs Clickable. In proceedings of the 9th workshop on Mobile computing systems and applications, CA, USA , pp 91-94

[4] Yan, Jeff, Ahmad, Ahmad Salah El 2008. Usability of CAPTCHAs Or usability issues in CAPTCHA design. In Symposium On Usable Privacy and Security (SOUPS), Pittsburgh, PA, USA

[5] Bursztein, Elie, Martin, Matthieu, Mitchell, John C. 2011. Text-based CAPTCHA strengths and weaknesses. In ACM Computer and Communication Security

[6] Raj, Aditya, Jain, Ashish, Pahwa, Tushar, Jain, Abhimanyu 2010. Picture CAPTCHAs with Sequencing:

[7] Their Types and Analysis. In International Journal of Digital Society (IJDS), Volume 1, Issue 3

[8] Bursztein, Elie, Bethard, Steven, Fabry, Celine, Mitchell, John C., Jurafsky, Dan 2010. How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation. In IEEE Symposium on Security and Privacy

[9] Fritsch, Christoph, Netter, Michael, Reisser, Andreas, Pernul, Günther 2010. Attacking Image Recognition Captchas – A Naïve but Effective Approach. In 7th International Conference, TrustBus, Bilbao, Spain, pp 13-25

[10] Turbobit – A File Sharing Service "http://turbobit.net/captcha/securimg_1"

[11] Desai, Arpan, Patadia, Pragnesh 2009.Drag and Drop: A Better Approach to CAPTCHA. In Annual IEEE India Conference (INDICON) , Gujarat, India, pp 1-4

[12] Amrinder, Arora 2007. Statistics Hacking – Exploiting Vulnerabilities in News Websites. In International Journal of Computer Science and Network Security Vol. 7 No. 3

[13] Bursztein Elie, Beauxis Romain, Paaskov Hristo, Perito Daniele, Fabry Celine and Mitchell John, 2011. The Failure of Noise-Based Non-Continuous Audio CAPTCHAs. In IEEE Symposium on Security and Privacy, Berkeley, California pp 19 – 31