Image-based CAPTCHA with JACI

Ryan Doyle
Box Hill Institute, Centre for ICT
465 Elgar Road, Box Hill, Melbourne, Victoria Australia
ryan@doylenet.net

***Abstract*— This paper proposes JACI (Just Another CAPTCHA Implementation). JACI is an image-based CAPTCHA technology that requires users to match like images with their partner image in order to pass the test. JACI is not dependant on keyboard interaction which aids some forms of accessibility and does not require internationalisation. Interaction is controlled with a pointing device such as a mouse; as each image is dragged and dropped over the other partner image, generating an overall rich user experience.**

I. Introduction

CAPTCHA (Completely Automated Tuning test to tell Computers and Humans Apart) technology is used most commonly on the web to tell the difference between a human using a web service and an automated bot. Many websites, large and small, now implement some form of CAPTCHA to avoid abuse of the services that they provide. The most common form of CAPTCHA is the obstructed word CAPTCHA that requires users to enter in the text from an image that is warped or otherwise distorted, making Optical Character Recognition technology incapable of deciphering the text. While this is successful at deterring some automation tools, only the latest generation of distorted word CAPTCHA are considered secure[4]. As complexity grows, so does the possibility of false positives, as words become more and more distorted.

Proposed is JACI[6]; an image-based CAPTCHA that works on the principle that a human will be able to recognise images with similar content. The current level of artificial intelligence is definitely no where near this capability so this type of image recognition CAPTCHA is assumed effective for quite some time. Users are required to drag and drop images onto their partner image to pass the test. There is no need to touch the keyboard which makes this CAPTCHA useful when there is no keyboard interaction wanted or available (such as a kiosk environment).

II. Overview of CAPTCHA Technology

A CAPTCHA is defined as "*a test, any test, that can be automatically generated, which most humans can pass, but that current computer programs cannot pass*[1]". The CAPTCHA must be automatically generated and also judged by the computer.

One of the main uses of CAPTCHA technology is to ensure that whenever a user is submitting data to a website that it is indeed a human that is submitting the data and not an automated bot. This applies for free email accounts such as Yahoo! Mail and GMail, online polls and most Web2.0 services where the user in some way has the ability to generate, alter or bias the content[5].

A. Distorted Word CAPTCHA

This paper will focus on the common distorted image CAPTCHA found in Yahoo! and GMail account sign-up processes to draw conclusions against JACI.

Fig 1. Example of a CAPTCHA found in the GMail sign-up process

GMail CAPTCHA have yet to be reliably broken. It relies only on warping on the letters on a white background. The words are random lower-case only letters. The Yahoo! CAPTCHA is similar to GMail, but includes upper-case letters and numbers as well. Both are similar in their readability. The latest generation of Yahoo! CAPTCHA are also yet to be reliably broken.

Fig 2. Example of a Yahoo! CAPTCHA. This includes upper-case letters as well as numbers

B. Image-based CAPTCHA

Image-based CAPTCHA technology relies on the assumption that an automated bot will not be able to recognise and interpret an image. IMAGINATION[2,3] is a system proposed by researches at the Pennsylvania State University. It is a two stage image-based CAPTCHA. Initially the user is required to click an approximate centre of several combined sets of images. The images are slightly distorted to ward off possible attacks. Secondly, the user is presented with an image to which they must choose an appropriate description for the image presented. Although not explicitly defined by IMAGINATION, the images seem pooled from a bank of images unique to IMAGINATION.

IMAGINATION is more user friendly than distorted word CAPTCHA as it only requires a pointing device for input. Unfortunately, the second stage of the test could be problematic when the user is required to match an image to a word. There are countermeasures put in place to ensure that a picture will not overlap with two possible descriptions. This requires human intervention to screen the images and then associate the correct tags.

III. Operation of JACI

A. Overview

JACI requires users to match images based on like content. Images can be represented as *a(0) – a(max)* with the corresponding partners *b(0) – b(max)*. One image from *a(x)* is then dragged over *b(x)*. The images then disappear and this relationship is stored. Once *a(final)* is dropped on *b(final)* the relationship is submitted and checked against the correct relationship that was generated. The order of *a(x)* to *b(x)* is completely random. Different subjects are also randomly selected each time the test takes place.

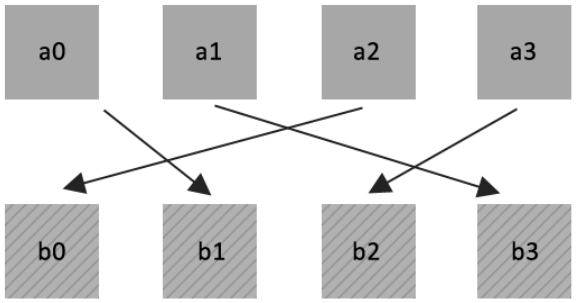

Fig 3. An example mapping of *a(x)* to *b(x)*. The order and images are selected dynamically and randomly each time.

B. Content Selection

Currently the content/subjects for the images are hard-coded into Jaci. The mock-up Jaci used the following list of subjects

- Cat
- Dog
- Lake
- Frog
- Elephant
- Monkey
- Popcorn

These subjects were chosen completely at random. Subjects should not overlap. For example, lake and ocean should not be used in the same list. Subjects need to be general enough but also not too specific that there won't be enough images. Static subjects could be seen as a disadvantage, but with the amount of images that can be generated from only one subject; the amount of user intervention versus the potential large pool of images is negligible. It is possible that JACI could be extended to dynamically generate subjects or have a large online repository. These are all extensions that could be added to the original JACI implementation. The method by which these images are sourced will be discussed in the next section.

C. Dynamic Image Fetching

Google Image Search is used as a source for Jaci. Jaci submits a subject from the predefined list twice per subject. Google Image Search allows for up to 1000 results and two out of these results are then saved to be passed onto the user performing the test.

Google Image Search stores a cached thumbnail of each image that it indexes. When searching for an image the images that appear are coming from Google. Once you click onto this image, Google will then take you to the website. Google stores its thumbnails at http://tbn0.google.com/images?q=tnb:*xxxx* where *xxxx* is a random string of upper/lowercase letters, numbers and underscores of about 14 characters.

Unfortunately the image source is *not* located in the HTML source of the page. Rather, the entire page is generated through JavaScript. Initially it was thought that JACI would need to parse the page through some sort of JavaScript processor and then extract the HTML. This proved to be quite difficult and time consuming so the source of the JavaScript was inspected. The ID of the image in the form of *xxxx* was easily accessible as well as the width and height of the image.

The regular expression, `/dyn\.Img\(\".*\,\[\]\)\;/is` was used to extract the section of JavaScript relating to the image thumbnails. The elements of this are then put into an array and accessed accordingly. The source of the image can then be constructed and passed to the user performing the test.

In order to get an image, another HTTP GET variable is passed, *start=y*, where *0 < y < 1000*. The random number generator is fed through the biasing function and then inserted into the URL each time a new image is fetched.

D. Biasing the Random Number Generator

An important part of JACI is the operation of the random number generator, herein known as RNG. A true RNG should achieve an even distribution of numbers within the domain. In the case of the Google Image Search, it is *assumed* that, like a traditional Google search, more relevant results are at the beginning. Simply reducing the RNG domain size was not an option as this also reduces the strength of Jaci against attack. Knowing this, a function was devised that would give a *biased* random number.

$$f(x) = \frac{x^b}{c^{b-1}}$$

Fig 4. Where x = RNG input, c = domain of RNG, b = order of bias. F(x) will then be used for the start location of the Google Image Search.

This function can be proved by letting x = c and x = 0. For this example, set c = 1000 and let the order of bias = 4.

$$f(x) = \frac{1000^4}{1000^{4-1}}$$

$$f(x) = \frac{1000^4}{1000^3}$$

$$f(x) = \frac{1000^1}{1000^0}$$

$$f(x) = 1000$$

Also prove when x = 0

$$f(x) = \frac{0^4}{1000^{4-1}}$$

$$f(x) = \frac{0^4}{1000^3}$$

$$f(x) = \frac{0^1}{1000^0}$$

$$f(x) = 0$$

The best way to visualise this function is to see it plotted. The graph in Fig. 5 shows this function plotted when c = 1000 and b = 2, 3 and 4.

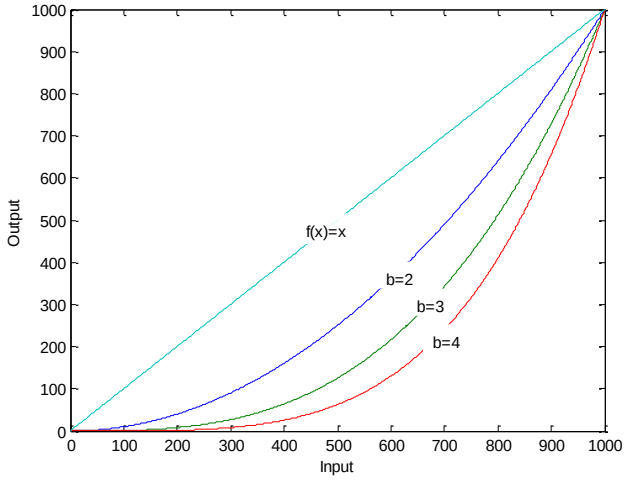

Fig 5. The plot shows that by altering b, we can change how biased random numbers will be.

The result of this function then undergoes the ceiling operation to make sure that the resultant is still an integer.

E. Accessibility and Functionality

One of JACI's strengths is its rich and accessible user experience. The drag and drop interface of JACI is logical to the user doing the test. As aforementioned, this functionality does not require a keyboard and only requires a pointer-like interface. This could be a mouse or touch screen.

Unlike distorted word CAPTCHA, Jaci does not require the user to be able to read or recognise words. This is useful for users that are dyslexic or have vision difficulties. Jaci does not suffer from a need for internationalisation. It does not deal with words and letters. Provided the selected list of subjects is general enough so that all cultures are able to identify the pictures, it can be used on a world-wide scale.

IV. Strength of Jaci

The strength against a brute force attack increases a at rate of *f(x) = 1/x!*, where x is the number of image pairs and *f(x) = 1* is success. Using 4 image pairs works out to a 4% chance of guessing the correct mapping of pairs which is currently what Jaci uses.

Number of Pairs	*1/x!*	*% chance of guess*
2	.5	50%
3	.16667	17%
4	.04167	4%
5	.00834	.8%
6	.00139	.1%
7	.0001984	.01%
8	.0000025	.0003%

Increasing the number of image pairs also increases the complexity of the test. It is recommended that no more than 6 image pairs be used in the test.

JACI should not be the only mechanism to protect websites against automated bots. It is important that there is also a limit in the number of retries before the user is then blocked from attempting the test again. This should be quite low, around 2 attempts would be sufficient for 4 pairs, and 3 attempts for 5 pairs. Without this protection, JACI will be broken for smaller image pairs quite easily.

V. Conclusion

JACI contributes a new and generic way of looking at CAPTCHA tests. It is not a perfect implementation but may promote more image-based authentication schemes to appear. If the images are relevant enough, an extremely broad range of candidates will be able to pass this CAPTCHA test. To users that are not as technically minded, it also makes more sense to "play" what appears as a game than to enter in distorted text which may or may not be correct. This user-friendly aspect of JACI is one of its major strengths.

Unfortunately JACI is not particularly strong against a brute force attack with low image pairs. Other security features need to be implemented in order to make Jaci secure against abuse.

Any kind of intelligent attack against JACI can be dismissed as a result of the current level of Artificial Intelligence. It is concluded that when a computer is able to successfully recognise images according to their subject matter, JACI would be redundant as would most other current CAPTCHA technologies currently used today.

References

[1] L.v. Ahn, M. Blum, J. Langford, "Telling Humans and Computers Apart Automatically", Communications of the ACM available http://www.CAPTCHA.net/CAPTCHA_cacm.pdf. February 2004.

[2] R. Datta, J. Li, and J.Z. Wang, "IMAGINATION: A Robust Image-based CAPTCHA Generation System," Proceedings of the ACM Multimedia Conference, pp. 331-334, Singapore, ACM, November 2005.

[3] J. Wang , "IMAGINATION – image based CAPTCHA authentication", available http://alipr.com/captcha/, 2008.

[4] K. Chellapilla, P. Y. Simard, "Using Machines Learning to Break Visual Human Interaction Proofs (HIPs)", available http://research.microsoft.com/~kumarc/pubs/chellapilla_nips04.pdf,

[5] L.v. Ahn, M. Blum, N. J. Hopper, J. Langford, "CAPTCHA: Using Hard AI Problems For Security", available http://www.captcha.net/captcha_crypt.pdf.

[6] R. Doyle, "JACI : Just Another Captcha Implementation" available http://ryandoyle.net/devel/jaci/ November 2008.