Protecting Websites with Reading-Based CAPTCHAs

Henry S. Baird
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
baird@parc.com www.parc.com/baird

Mark Luk
Computer Science Division
Univ. of California at Berkeley
Berkeley, CA 94720 USA
markluk@uclink.berkeley.edu

Abstract

Recent document image understanding R&D intended to protect Internet services against abuse by programs is summarized. The accelerating pace of introduction of working CAPTCHAs — completely automatic public Turing tests to tell computers and humans apart — is reported, and the ability of these CAPTCHAs to resist attack is critiqued. Attacks on PARC's 'BaffleText' CAPTCHA by the DIA, CV, and security communities are cordially invited by a PARC website www.parc.com/istl/projects/captcha.

Keywords: Web security, protecting e-commerce, Human Interactive Proofs (HIPs), Completely Automatic Public Turing test to tell Computers and Humans Apart, CAPTCHA, psychophysics of reading, optical character recognition (OCR), PessimalPrint, BaffleText, Turing tests

1. Introduction

The proliferation of publicly available services on the Internet has invited abuses by programs ('bots', 'spiders') designed to steal services and conduct fraudulent transactions. Some examples:

- Free online accounts are being automatically registered, many times, and then used to distribute stolen copyrighted material[10].

- Recommendation systems are vulnerable to artificial inflation or deflation of ratings. For example, E-bay, a high-traffic auction website, allows its users to rate buyers and sellers on the basis of how well they complete transactions[7]. Unscrupulous sellers rate themselves positively, thousands of times automatically, in order to hoodwink buyers into believing that they are trustworthy.

- Spammers register free e-mail accounts offered by such services as Hotmail in large numbers and use them to send unsolicited email[11].

These are just a few examples of actions which are tolerable when performed occasionally by individuals, but become abusive when executed many times automatically[4, 3].

2. CAPTCHAs

One practical defense against such abuse are CAPTCHAs: **C**ompletely **A**utomatic **P**ublic **T**uring tests to tell **C**omputers and **H**umans **A**part. Virtually all CAPTCHAs presently in commercial use exploit the ability of people to read images of text more reliably than optical character recognition (OCR) and other machine vision systems. Their challenges are typically created as follows: pick a word (or non-word character string), pick a typeface (or faces), render the word using the typeface into a raster image, and degrade the image. The choices of word, typeface, and degradation must be engineered to yield images which are easy for humans to recognize but baffling to all OCR systems now and, one hopes, for years to come. Then, if a subject can correctly transcribe (read and type in) the word in the image, the subject may be judged to be human, not a machine.

The first CAPTCHA was invented in 1997 by Andrei Broder and his colleagues [9], then at the DEC Systems Research Center, and was used to block the abusive automatic submission of URLs [1] to the AltaVista web-site. Yahoo! uses a CAPTCHA called EZ-Gimpy, developed at The School of Computer Science at Carnegie-Mellon University, to protect a variety of on-line services[4] including registering for free email accounts. Greg Mori and Jitendra Malik of the Computer Science Division at U.C. Berkeley describe an attack[12] on EZ-Gimpy, using lexical knowledge and 'generalized shape contexts,' which enjoyed a success rate of 83%. Another early example of a reading-based CAPTCHA is PARC/UCB's PessimalPrint[6]. For a survey of other reading-based CAPTCHAs, see [2].

3. The BaffleText CAPTCHA

BaffleText is a reading-based CAPTCHA developed at PARC by the first author and Monica Chew [5] that uses random masking to degrade images of non-English pronounceable character strings. Each BaffleText challenge is generated as follows:

1. generate a pronounceable English-like character string and ensure it is not in the English dictionary;
2. choose a font from among a large number;
3. render the character string using the font into an image (ideally, that is without physics-based degradations);
4. generate a mask image (described below);
5. choose a masking operation from among Boolean ‘union,’ ‘not-and,’ and ‘exclusive-or’; and
6. combine the character-string image and mask image using the masking operation.

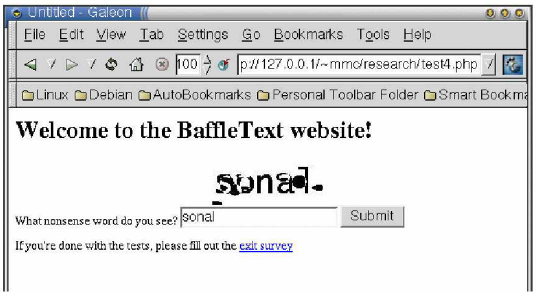
Parameters governing mask generation include:

1. Masking shape: any combination of circles, squares, and ellipses;
2. Minimum radius or radii (in pixels) of the masking shapes: for circles, this was the ordinary radius; for squares, this was the half of the minimum allowable side length; for ellipses, half the major axis or half the minor axis.
3. Maximum radius or radii (in pixels) of the masking shape, similar to minimum radius.
4. Density: the fraction of black pixels in the resulting mask.

Pronounceable character strings are generated by a character-trigram Markov model trained on the English-language Brown corpus[8, 13], in order to seem somewhat familiar to users. The strings contain only lowercase alphabetic characters and are between 5 and 8 letters long. They also are filtered so they do not appear in `/usr/share/dict/words` in order to deter known-lexicon-based attacks.

Examples of BaffleText challenges can be seen in Table 1. The P^2/A numbers are measures of ”image complexity,” defined and discussed in [5], which, it has been shown empirically, are well correlated with both objective and subjective difficulty experienced by human readers.

The Mori and Jitendra attack was unable to break BaffleText. To the best of our knowledge, BaffleText possesses the strongest defenses against automatic attack of any known CAPTCHA. But how can this conjecture be systematically verified?



Untitled - Galeon
File Edit View Tab Settings Go Bookmarks Tools Help
100
p://127.0.0.1/~mmc/research/test4.php
Linux Debian AutoBookmarks Personal Toolbar Folder Smart Bookma
Submit

Welcome to the BaffleText website!

What nonsense word do you see? sonal

If you're done with the tests, please fill out the exit survey

Figure 1. Screen shot of BaffleText challenge site.

4. A BaffleText Website Open for Use and Attack

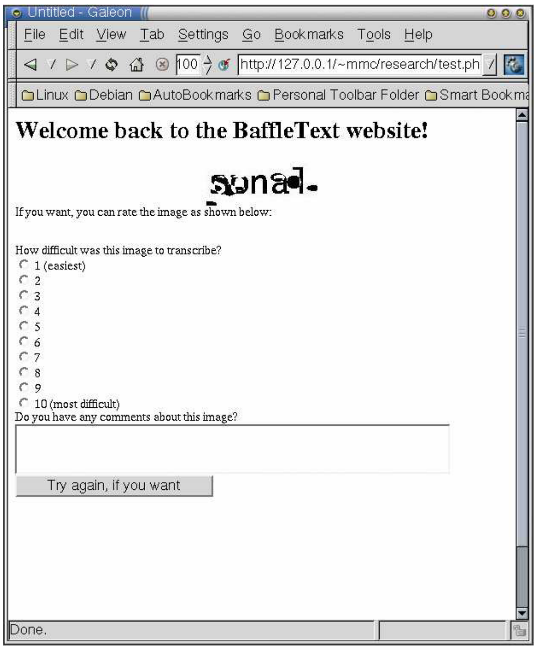
We have built a CAPTCHA website `www.parc.com/istl/projects/captcha` which serves BaffleText challenges and gives background information on CAPTCHAs, including links to the history of CAPTCHA and current articles. We invite human users — and programs designed to attack it — to visit the website.

Human users, especially, are invited to try to answer as many BaffleText challenges as they are willing to: their responses will be recorded and analyzed to help us evaluate our CAPTCHAs.

Currently, our website generates BaffleText images over an extremely wide range of difficulty: some are so easy that many OCR systems will be able to read them; others are so difficult that most human will get them wrong; and many and in-between, in the useful regime. Different applications of CAPTCHAs are likely, in general, to require differing difficulty regimes. To protect occasional, high-value transactions, such as signing up for free email accounts, difficult challenges may be appropriate and are likely to be well tolerated by users. By contrast, frequent low-value transactions, such as sending email, may be best served a low-difficulty regime.

For each answer to a BaffleText challenge, we record the user’s answer of course, plus the user’s estimate of difficulty, response time, and optional comments. Response time is measured at the server side, and so includes the round-trip network communication time between server and client machines. The user rating of the perceived ‘difficulty’ of each image is on a scale of 1-10 (10=hardest), which we record before revealing whether or not the user’s answer was correct. We will give a live demo of this website at the WDA workshop. Any comments on the design of the website will be highly appreciated.

We are eager to work with our DIA colleagues to enable automatic attacks on BaffleText. For this purpose,


Untitled - Galeon
File Edit View Tab Settings Go Bookmarks Tools Help
100
http://127.0.0.1/~mmc/research/test.ph
Linux Debian AutoBookmarks Personal Toolbar Folder Smart Bookma
Done.

Welcome back to the BaffleText website!

If you want, you can rate the image as shown below:

How difficult was this image to transcribe?
○ 1 (easiest)
○ 2
○ 3
○ 4
○ 5
○ 6
○ 7
○ 8
○ 9
○ 10 (most difficult)
Do you have any comments about this image?

Try again, if you want

Figure 2. Screen shot of BaffleText user-rating page.

we expect it may be good to redesign the website to serve batches of CAPTCHAs to automated attackers. It might be helpful in some cases for PARC to grade attackers' performance without revealing the correct answers. It might also be useful to offer versions of BaffleText tailorable to user-specified ranges of difficulty.

We will also, at the workshop, invite suggestions from colleagues about ways to include BaffleText and other reading-based CAPTCHAs in competitions run by the DIA R&D community.

References

[1] AltaVista's "Add-URL" site: `altavista.com/sites/addurl/newurl`, protected by the earliest known CAPTCHA.

[2] H. S. Baird and K. Popat. Human interactive proofs and document image analysis. In *Proc., IAPR 2002 Workshop on Document Analysis Systems*, Princeton, NJ, August 2002.

[3] D. P. Baron. eBay and database protection. Case No. P-33, Case Writing Office, Stanford Graduate School of Business, 518 Memorial Way, Stanford Univ., Stanford, CA 94305-5015, 2001.

[4] M. Blum, L. A. von Ahn, J. Langford, and N. Hopper. The CAPTCHA Project: Completely Automatic Public Turing test to tell Computers and Humans Apart. http://www.captcha.net, November 2000.

[5] M. Chew and H. S. Baird. Baffletext: a human interactive proof. In *Proc., 10th IS&T/SPIE Document Recognition & Retrieval Conf.*,, Santa Clara, CA, January 23–24 2003.

[6] A. L. Coates, H. S. Baird, and R. Fateman. Pessimal print: a reverse turing test. In *Proc., IAPR 6th Intl. Conf. on Document Analysis and Recognition*, pages 1154–1158, Seattle, WA, September 2001.

[7] eBay. ebay. http://www.ebay.com.

[8] W. Francis and H. Kucera. The brown corpus of standard american english, 1961.

[9] M. D. Lillibridge, M. Abadia, K. Bharat, and A. Z. Broder. Method for selectively restricting access to computer systems. U.S. Patent No. 6,195,698, Issued February 27, 2001).

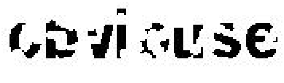
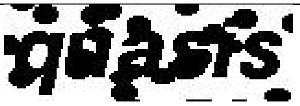
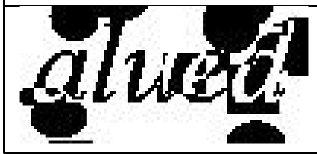
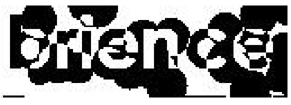
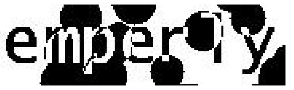
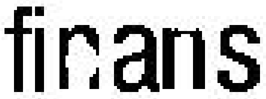
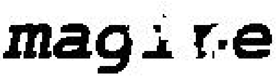
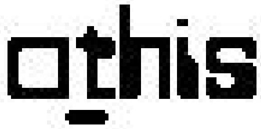
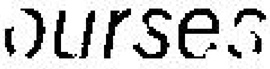
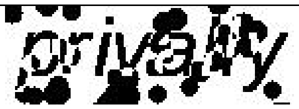
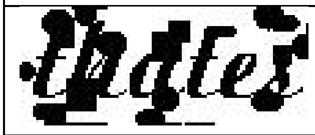
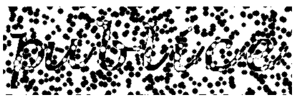
[10] U. Manber. Large scale internet security. Invited talk, IEEE Symposium on Security and Privacy, May 2002.

[11] Microsoft. Hotmail. http://www.hotmail.com.

[12] G. Mori and J. Malik. Breaking a visual captcha. 2002. In submission to Computer Vision and Pattern Recognition 2003.

[13] K. Popat, D. Bloomberg, and D. Greene. Adding linguistic constraints to document image decoding. In *Proc., 4th IAPR Workshop on Document Analysis Systems*, Rio de Janeiro, Brazil, December 2000.

Table 1. Examples of BaffleText

Image	word	Image	word
	obviouse, $P^2/A = 298$		quasis, $P^2/A = 280$
	alued, $P^2/A = 115$		brience, $P^2/A = 118$
	emperly, $P^2/A = 90$		finans, $P^2/A = 49$
	magine, $P^2/A = 113$		othis, $P^2/A = 14$
	ourses, $P^2/A = 113$		privally, $P^2/A = 178$
	thates, $P^2/A = 309$		publice, $P^2/A = 2900$