Video CAPTCHAs: Usability vs. Security

Kurt Alfred Kluever[1] and Richard Zanibbi[2]
Document and Pattern Recognition Lab
Department of Computer Science
Rochester Institute of Technology, Rochester, NY USA
kak@google.com[1], rlaz@cs.rit.edu[2]

September 26, 2008

1 Introduction

A Completely Automated Public Turing test to tell Computer and Humans Apart (*CAPTCHA*) is a variation of the Turing test, in which a challenge is used to distinguish humans from computers ('bots') on the internet. They are commonly used to prevent the abuse of online services; for example, malicious users have written automated programs that signup for thousands of free email accounts and send SPAM messages. A number of hard artificial intelligence problems including natural language processing, speech recognition, character recognition, and image understanding have been used as the basis for these tests, on the expectation that humans will outperform bots. The most common type of CAPTCHA requires a user to transcribe distorted characters displayed within a noisy image. Unfortunately, many users find existing character-recognition based CAPTCHAs frustrating and attack success rates as high as 60% have been reported for Microsoft's Hotmail CAPTCHA [8].

To address these problems, we present a first attempt at using content-based video labeling ('tagging') as a CAPTCHA task. We define correct responses using tags provided by the individual that posts a video to a public database (YouTube.com), along with tags on videos designated as being 'related' in the database. In an experiment involving 184 human participants, we were able to increase human pass rates on our video CAPTCHAs from roughly 70% to 90% while keeping the success of a frequency-based attack fixed at around 13%. Through a different parameterization of the challenge generation and tag matching algorithms, we were able to reduce the success rate of the same attack to 2%, while still increasing the human pass rate to 75% [5].

The frequency-based attack we consider is simple but logical for this type of CAPTCHA: the computer submits the three tags with the highest estimated frequencies below the rejection threshold, on the assumption that the tag frequency estimates used in creating the CAPTCHAs are publicly available.

A screenshot of our video-based CAPTCHA is shown in Figure 1. To pass the challenge, a user provides three words ('tags') describing the video. If one of the submitted tags belongs to the automatically generated ground truth tag set, the challenge is passed. This task is similar to the ESP game of von Ahn et al. [7], in which online users are randomly paired and presented with an image that they then submit tags to describe. Players cannot see each other's submitted tags until they agree on a common tag, at which point the round of the game ends. Our video CAPTCHA is similar to a game of ESP in which one player is online, while the other player's responses (the ground truth tags) are computed automatically.

2 Generating and Grading Challenges

Challenges are generated using a public video database (YouTube.com in our case). To select a video for generating a challenge, we use a modified version of a random walk through the videos in the database. First,

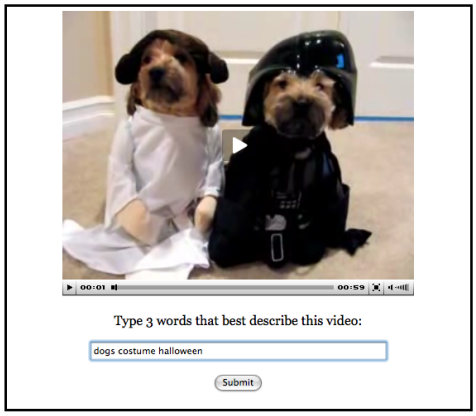

Figure 1: A Video CAPTCHA. The user watches a video and provides three tags. If one belongs to a set of ground truth tags, the challenge is passed.

we randomly select a word from an English dictionary and query the video database using it, and randomly select one of the returned videos. We then randomly select a tag from this video, query the database using the tag, and randomly select one of the returned videos. The process of selecting a tag, querying, and selecting a video is repeated for a number of steps randomly chosen between 1 and 100 in our experiments. Because of the database that we selected, a human was needed in the loop to insure that the selected video had appropriate content and contained English tags (due to the intended audience), but otherwise challenge generation is entirely automatic.

Once a video has been selected, we generate our challenges using a function with four parameters: the number of tags from related videos in the database to add n, the rejection threshold for tag frequencies t, and two boolean variables controlling whether word stemming s and approximate string matching l are used. In the simplest version of our CAPTCHA (i.e. the control condition in our experiments), no tags are added, no tags are rejected, and neither word stemming nor approximate matching are used. As can be seen in Table 1, people perform surprisingly well under this condition (69.7% pass rate in our experiment).

In our work, we used YouTube's related videos algorithm to obtain sources for additional ground truth tags. The workings of this algorithm are unpublished, but 'relatedness' seems to involve tag similarity and the number of views that a video has received. In our generation algorithm, we currently ignore the number of views for each video, and instead sort the returned related videos in decreasing order of cosine similarity for the tag sets. For a pair of videos, we represent their tag sets using binary vectors A and B, indicating which words in the union of the two tag sets are present for each video, and then compute their similarity as in the following:

$$\textsc{Sim}(A, B) = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|}$$

Up to n new, unique tags are then added to the ground truth set, as they appear in the sorted video list (selecting randomly if the last video observed has more unique tags than there are left to add). An interesting observation during our experiments was that this procedure often resulted in common misspellings of words such as 'balloon' (e.g. baloon) being added to the ground truth set.

The tag frequency rejection threshold t is used to increase security, by rejecting tags with an estimated frequency greater than or equal to this threshold. Tag frequencies were estimated using multiple random walks of the YouTube graph. The walking protocol is identical to that used to select challenge videos, except that all visited video identifiers and tags are stored in order to estimate frequencies [5]. The three most frequent tags found in our walks (over 86,368 videos) were 'music' (5.6%), 'video' (4.8%) and 'live' (3.4%). When we plotted the tags found in our walks in increasing order of frequency, the shape of the curve is exponential; a small number of words are used very frequently while most others are used very rarely.

Our video CAPTCHA challenge is passed if one of the submitted tags matches a ground truth tag. In the control condition, this is performed using exact matching (ignoring capitalization and punctuation). To increase usability, word stems produced using the Porter Stemming algorithm can be added to the submitted tag set (controlled by s), and approximate matching of submitted to ground truth tags using a length-normalized Levenshtein distance may be added (controlled by l).

3 Experiment

To evaluate the usability and security of our new CAPTCHA, we performed an experiment in which we compared human and frequency-based attack pass rates. Human pass rates were estimated using a sample of 20 videos from the YouTube.com database selected using the random walk procedure outlined in the previous section, while the attack pass rates were estimated using a separate sample of 5146 videos. In our experiment, the frequency-based attack submits the three most frequent tags occurring fewer times than the rejection threshold (t), selecting the tags to submit from the same frequency estimate used to generate the challenges. Results of the experiment are summarized in Table 1.

Table 1: Human and attack success rates. n is the number of tags added, t the tag frequency rejection threshold, s indicates if word stemming is used, and l indicates whether approximate matching of tags is used. $P_r(H)$ is the human success rate, $P_r(A)$ is the attack success rate, and Gap is the difference between the human and attack success rates

Condition	n	t	s	l	$P_r(H)$	$P_r(A)$	Gap
Control	0	1.0			0.6973	0.1286	0.5687
Most Usable	100	0.006			0.8828	0.1220	0.7608
Most Secure	30	0.002			0.7502	0.0239	0.7263
Largest Gap	45	0.006			0.8682	0.0750	0.7931
Most Usable	100	0.006	✓		0.8896	0.1226	0.7670
Most Secure	25	0.002	✓		0.7548	0.0209	0.7339
Largest Gap	45	0.006	✓		0.8755	0.0750	0.8005
Most Usable	100	0.006		✓	0.9000	0.1280	0.7719
Most Secure	15	0.003		✓	0.7671	0.0233	0.7438
Largest Gap	25	0.006		✓	0.8611	0.0526	0.8084
Most Usable	90	0.006	✓	✓	0.9019	0.1263	0.7755
Most Secure	15	0.003	✓	✓	0.7690	0.0237	0.7453
Largest Gap	25	0.006	✓	✓	0.8649	0.0526	0.8122

184 persons (primarily students, staff, and faculty in the College of Computing and Information Science at RIT) participated in the online experiment. Participants were recruited through email, fliers, and word-of-mouth. Each participant gave consent and completed a demographics questionnaire, labeled two practice videos, labeled an additional 20 videos (presented in random order), and then completed a brief exit survey.

As seen in Table 1, using stemming, approximate matching, and adding 90 related tags, we were able to increase human pass rates from 69.7% (the control) to 90.2% (our most usable condition). For this same condition, the attack success rate is kept at the same rate of the control, roughly 12.9% through pruning all tags with a frequency $\geq 0.6\%$. There is a tradeoff between usability and security; we can decrease the attack success rate to 2.1% but the human success rate will fall to 75.5%. In general, the parameters had the following effects: increasing the number of related tags added, increasing the pruning threshold, allowing stemming, or allowing approximate matching all increased the usability but decreased the security.

After completing 20 Video CAPTCHAs, only 20.1% of participants indicated that they found text-based CAPTCHAs more enjoyable, while 58.2% found our video-based alternative more enjoyable. However, 59.8% of participants indicated that they found text-based CAPTCHAs faster (the median challenge completion

time for a Video CAPTCHA was 17.1 seconds). Furthermore, our results indicate that Video CAPTCHAs have comparable usability and security to many existing CAPTCHAs (see Table 2). These results are encouraging and suggest that providing a Video CAPTCHA as an alternative to a text-based CAPTCHA may be a viable and user-friendly option.

Table 2: A comparison of human success rates ($P_r(H)$) and attack success rates ($P_r(A)$) for our video CAPTCHA (for our most usable condition) against several other well-known CAPTCHAs.

CAPTCHA Name	Type	$P_r(H)$	$P_r(A)$
Microsoft's CAPTCHAs [1]	Text-based	0.90 [1]	0.60 [8]
Baffletext [2]	Text-based	0.89 [2]	0.25 [2]
Handwritten CAPTCHAs [6]	Text-based	0.76 [6]	0.13 [6]
ASIRRA [3]	Image-based	0.99 [3]	0.10 [4]
Video CAPTCHAs [5]	Video	0.90 [5]	0.13 [5]

4 Future Work

The security of the Video CAPTCHA was only tested by a frequency-based attack. We acknowledge that other attacks may perform better. For example, computer vision could be used to located frames with text-segments in them, OCR them, and submit these as tags. Content-based Video Retrieval systems could be used to locate videos with similar content (and then submit their tags). Audio analysis might give an indication as to the content of the video.

The tag set expansion techniques presented are also an interesting avenue of future research. We can imagine other CAPTCHAs being developed which utilize social structure, perhaps one using Flickr images.

5 Acknowledgments

We gratefully acknowledge financial support from the Xerox Corporation through a University Affairs Committee (UAC) grant held with Bill Stumbo of Xerox Research Center Webster (XRCW).

References

[1] Kumar Chellapilla, Kevin Larson, Patrice Y. Simard, and Mary Czerwinski. Building Segmentation Based Human-friendly Human Interaction Proofs (HIPs). In *Proc. of HIP 2005*, pp. 1–26, Bethlehem, PA, May 2005.

[2] Monica Chew and Henry S. Baird. Baffletext: A Human Interactive Proof. In *Proc. of IST/SPIE Document Recognition and Retrieval X Conference 2003*, pp. 305–316, January 2003.

[3] John Douceur, Jeremy Elson, Jon Howell, and Jared Saul. ASIRRA: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In *Proc. of ACM CCS 2007*, pp. 366–374, New York, NY, October 2007.

[4] Philippe Golle. Machine Learning Attacks Against the ASIRRA CAPTCHA. To appear in *Proc. of ACM CCS 2008*, Alexandria, VA, October 2008.

[5] Kurt Alfred Kluever. *Evaluating the Usability and Security of a Video CAPTCHA*. Master's thesis, Rochester Institute of Technology, Rochester, NY, August 2008.

[6] Amalia Rusu. *Exploiting the Gap in Human and Machine Abilities in Handwriting Recognition for Web Security Applications*. PhD thesis, University of New York at Buffalo, Amherst, NY, August 2007.

[7] Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. In *Proc. of ACM CHI 2004*, pp. 319–326, New York, NY, April 2004.

[8] Jeff Yan and Ahmad Salah El Ahmad. A Low-cost Attack on a Microsoft CAPTCHA. To appear in *Proc. of ACM CCS 2008*, Alexandria, VA, October 2008.