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 sign up 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 challenges 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 CAPTCHAs based on character-recognition 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 the basis for a CAPTCHA. In our video CAPTCHA, a user must supply three tags about a video. We define the challenge answers (ground truth) using tags provided by the original uploader of the video as well as tags on videos designated as being 'related' in the database (YouTube.com). In an experiment involving 184 human participants, we were able to increase human success rates on our video CAPTCHAs from roughly 70% to 90%, while keeping the success rate of a tag frequency-based attack fixed at approximately 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 success rate to 75% [5].

A screenshot of our video CAPTCHA is shown in Figure 1. To complete a challenge, a user provides three words ('tags') that describe a video. If any of the submitted tags match any of the automatically generated ground truth tags, the user passes the challenge. 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 tag. Players cannot see each other's submitted tags until they have agreed 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

To select a video for generating a challenge, we use a modified version of a random walk through the videos in the database. First, we perform a query using a random word from the English dictionary and then randomly select one of the returned videos. Next, we randomly select a tag from this video, query the database using the tag, and then randomly select one of the returned videos. The process of selecting a tag, querying, and selecting a video is repeated for a random number of steps (between 1 and 100). A human

Figure 1: A video CAPTCHA. A challenge is passed if one of the three tags submitted by a user belongs to an automatically generated set of ground truth tags. Online demonstration: `http://sudbury.cs.rit.edu/`

was needed in the loop to ensure that the final 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 tag frequency rejection threshold (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 related tags are added, no ground truth 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% success rate in our experiment).

In our work, we used YouTube's 'related videos' algorithm to obtain 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 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 follows:

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

After sorting the videos, up to n new, unique tags are added to the ground truth set (randomly selecting a tag if the last video observed has more unique tags than there are left to add). An interesting observation during our experiments was that adding tags from related videos 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 YouTube.com. The random walk protocol is identical to that used to select challenge videos, except that tag counts are stored 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 was exponential; a small number of words are used very frequently while most others are used comparatively rarely [5].

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

3 Experiment

To evaluate the usability and security of our new CAPTCHA, we performed experiments to compare the human success rates and the tag frequency-based attack success rates. Human success rates were estimated using a random sample of 20 YouTube videos (selected using the random walk procedure outlined in the previous section), while the attack success rates were estimated using a separate sample of 5146 videos. 184 persons (primarily students and faculty at RIT) participated in our online experiment. The tag frequency-based attack submits the three most frequent tags directly below the rejection threshold (t), selecting the tags to submit from the same frequency estimate used to generate the challenges.

For both the human and attack experiments, the number of tags (n) was varied in increments of 5 between 0 and 200 tags (0, 5, 10, etc.), while the tag frequency rejection threshold (t) was varied in increments of 0.001 between 0.001 to 0.01 (the control, $t = 1.0$ was also computed). For the human experiment, n and l were set to false to test the most basic condition during the experiment. Human success rates for other parameter settings were computed afterwards [5]. Results are summarized in Table 1; shown are the conditions for which the most usable, most secure, and largest human/machine success rate differences ('gap') were observed, where the human success rate is no worse than the control, and the attack success rate is no better than the control.

Table 1: Human and attack success rates where 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	$P_r(H) : 20v$	$P_r(A) : 5146v$	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

Using stemming, approximate matching, adding 90 related tags, and pruning tags with a frequency $\geq 0.6\%$, we were able to increase human success rates from 69.7% (the control) to 90.2% while maintaining an attack success rate of approximately 13%. There is a tradeoff between usability and security; we were able to decrease the attack success rate to 2.1% but the human success rate dropped to 75.5%. In general, increasing the number of related tags, increasing the rejection threshold, allowing stemming, and approximate tag matching increases usability but decreases security. Different balances preferring usability, security, or maximum gap between the human success rate and attack success rate may be found in Table 1.

In an exit survey, 58.2% of the participants indicated that they found the video CAPTCHA more enjoyable than character-recognition based CAPTCHAs (only 20.1% preferred these to the video CAPTCHA). However, 59.8% of participants indicated that they found text-based CAPTCHAs faster (video CAPTCHA completion times in seconds were $\mu = 22.0, \sigma = 23.6$, median = 17.1). Our results also suggest that our

video CAPTCHA has comparable usability and security to existing techniques (see Table 2).

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 our CAPTCHA has only been tested with a tag frequency-based attack, and we acknowledge that other attacks may be more effective. 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), and 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 inherent social structure in tagged data, for example using images from Flickr.com.

5 Acknowledgments

We gratefully acknowledge financial support from 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.