

# Poster: Captchæcker – Automating Usability-Security Evaluation of Textual CAPTCHAs

Maliha Nazir, Youstra Javed,  
Muhammad Murtaza Khan, Syed Ali Khayam  
National University of Science & Technology (NUST)  
Islamabad, Pakistan  
{maliha.nazir, youstra.javed, muhammad.murtaza,  
ali.khayam}@seecs.nust.edu.pk

Shujun Li  
University of Konstanz  
Germany  
Shujun.Li@uni-konstanz.de

## 1. INTRODUCTION

CAPTCHAs are now deployed ubiquitously on the Internet to combat automated malicious programs. A major problem with many deployed CAPTCHA schemes is that they are either too weak in terms of security or unacceptable in terms of usability. Taking the CAPTCHA scheme used by Google Account as an example, all letters in the CAPTCHA image are often heavily distorted and connected with each other, which increases security but lowers usability. For instance, an average user (i.e., neither an expert nor an elderly person with limited computer exposure) will encounter significant difficulty in recognizing the Google CAPTCHA images shown in Figure 1.

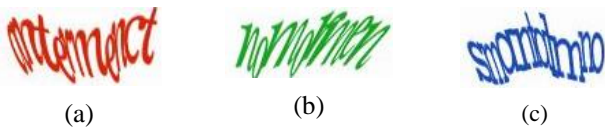


Figure 1: Three *hard* Google CAPTCHA images.

To circumvent hard CAPTCHA images, users often simply refresh the Web page until an easy CAPTCHA image comes. In other words, the (stronger) CAPTCHA scheme is reduced to a subset of weaker CAPTCHA images. Since it is very difficult to find a good balance between security and usability, many web sites choose to deploy more usable but less secure CAPTCHAs.

Balancing the delicate security-usability tradeoff of a CAPTCHA scheme remains an art rather than a science. This tradeoff can be more easily balanced if we can quantitatively evaluate the security and usability of CAPTCHAs in an automated manner.

In this poster, we present the first attempt (to the best of our knowledge) of automated usability-security evaluation of CAPTCHAs. The main goal is to automate the process of evaluating the hardness of different kinds of textual CAPTCHAs judged by an average user with normal eyesight, which is a direct metric of usability. This hardness measurement can also be an indirect metric of security since if a CAPTCHA is very hard for human users then it is likely even harder for automated programs. We base our automated evaluation on a number of geometric indicators that can be measured via simple image processing techniques. We name our system Captchæcker, meaning “Captcha

Checker”. We have used a set of 50 CAPTCHAs from Google, Yahoo! and Microsoft subjectively rated by 20 users for training a hardness classifier and a new set of 35 CAPTCHAs rated by 5 new users for testing the classifier. We show that Captchæcker can predict hardness of a CAPTCHA in the testing set with accuracy over 80%, thus allowing us to automatically judge how usable and secure a CAPTCHA is.

## 2. GEOMETRIC INDICATORS

We used the following geometric indicators in our Captchæcker system to capture different aspects of hard CAPTCHAs.

### 2.1 Shape Compactness

“Crowding characters together” [1] is one of the most widely-used approaches of enhancing security of CAPTCHAs, which has a side effect of reducing usability. We observed that the level of crowdedness or compactness ( $C_n$ ) of a CAPTCHA can be measured following the spirit of isoperimetric quotient of a shape with closed boundary [2]:  $C_n = \text{Perimeter}^2 / \text{Area}$ .

### 2.2 Euler’s Number

Crowding characters together in a CAPTCHA can create overlaps between adjacent characters resulting in larger fused areas and new holes between them. Generation of new holes and connected components results in a different Euler’s number (EN), thereby making it useful for our Captchæcker. We consider the CAPTCHA as the object of interest and define EN as follows:  
 $EN = \text{Number of Connected components} - \text{Number of holes}$ .

### 2.3 Thickness/Boldness

Thickness/boldness of the characters in a CAPTCHA is often linked to its hardness. We use the number of steps for morphologically eroding all characters in a CAPTCHA image as a measure of the thickness/boldness. This measure is called the number of Erosion Steps (ES). We have used a square-shaped structuring element of size  $2 \times 2$  pixels to calculate ES.

### 2.4 Compact-Length and Euler-Thickness

Two new indicators are defined based on the above ones:

- Compact-Length (CL): the ratio between compactness ( $C_n$ ) and the CAPTCHA text width ( $C_w$ );
- Euler-Thickness (ET): the ratio between Euler’s number (EN) and the number of Erosion Steps (ES).

These two indicators are used because a combination of them allows us to distinguish easy and hard CAPTCHAs with an acceptable accuracy. More details are given in the next section.

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.

Symposium On Usable Privacy and Security (SOUPS) 2011, July 20-22, 2011, Pittsburgh, PA, USA.









Google CAPTCHA		Google reCAPTCHA	
			
Cn=1122, EN=-8, ES=7, Cw=151 CL=7.43, ET=-1.14	Cn=1430, EN=-37, ES=6, Cw=122 CL=11.7, ET=-6.1	Cn=1525, EN=-4, ES=9, Cw=317 CL=4.8, ET=-0.44	Cn=1630, EN=-11, ES=6, Cw=318 CL=5.12, ET=-1.8
Microsoft CAPTCHA with Two Rows		Yahoo! CAPTCHA	
			
Cn=1070, EN=-7, ES=28, Cw=192 CL=5.57, ET=-0.25	Cn=1508, EN=-5, ES=12, Cw=171 CL=8.82, ET=-0.42	Cn=1412, EN=-7, ES=6, Cw=108 CL=13.1, ET=-1.16	Cn=1016, EN=-5, ES=8, Cw=158 CL=6.43, ET=-0.625

Figure 2: Objective hardness indicators of selected CAPTCHAs of four different CAPTCHA schemes.

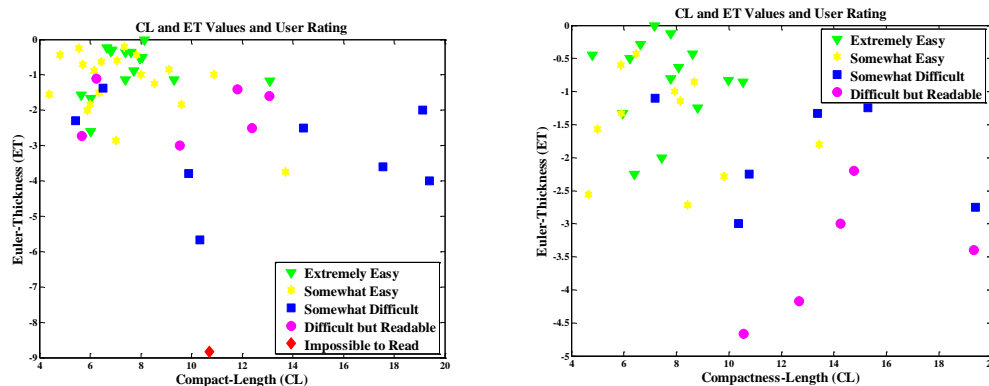


Figure 3: Usability study for training the classifier: training data (left) and testing data (right).

### 3. RESULTS

The CAPTCHA schemes involved in this work include Google CAPTCHA, Google reCAPTCHA, a Microsoft CAPTCHA scheme with two rows (one of several CAPTCHA schemes used by Microsoft) and Yahoo! CAPTCHA schemes. The geometric indicators of selected CAPTCHAs are shown in Figure 2. It can be inferred from the figure that higher CL and lower ET values are obtained for CAPTCHAs that are more compressed. The task of Captchæcker is to predict the hardness given the geometric indicators of a CAPTCHA. We consider this as a binary classification problem: given a feature vector formed by the geometric indicators, a CAPTCHA is classified as easy or hard. To train the classifier, we collected subjective hardness scores from 20 users on 50 CAPTCHAs. The users were asked to rate the CAPTCHAs on a 5-point scale so that we can easily define the boundary between easy and hard CAPTCHAs: 1 = extremely easy, 2 = somewhat easy, 3 = somewhat difficult, 4 = difficult but readable, 5 = impossible to read. For each CAPTCHA, we used the median score as the average user’s rating.

Figure 3 shows the scatter plot of the average user’s ratings of the 50 CAPTCHAs on the CL-ET plane. One can see that the upper left corner of the plane contains mainly easy CAPTCHAs (green and yellow markers), which implies that the 2-tuple (CL,ET) can be used to get a classifier with acceptable classification accuracy. Based on the training set, we trained a binary classifier NN. A 5-fold cross-validation scheme is used for training to avoid any bias

due to the random selection of the training and validation sets. We tested the classifier on a testing set with 38 new CAPTCHAs and five new users. We trained the NN approximately 30 times with different random partitions of the training set to test the stability of classification results. Average classification accuracy of the 5-fold cross-validation process is larger than 80% in all cases except one (76.8%) and with a high probability exceeds 85%.

### 4. CONCLUSIONS

This poster reports an automated evaluation system called Captchæcker, which is used to predict the hardness of CAPTCHAs. Automation of the CAPTCHA evaluation process can help CAPTCHA designers to judge automatically (i.e. without human intervention) how usable and secure a CAPTCHA scheme is and how it can be further enhanced. In our future work, we will try to build a larger database of subjective evaluation and develop better indicators to further improve the accuracy of Captchæcker.

### 5. REFERENCES

- [1] J. Yan and A. S. El Ahmad, “A Low-cost Attack on a Microsoft CAPTCHA”, in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS’08)*, pp. 543-554, ACM, 2008.
- [2] Hermann Kremer and Eric W. Weisstein, “Isoperimetric Quotient,” from MathWorld – a Wolfram web resource, <http://mathworld.wolfram.com/IsoperimetricQuotient.html>