The Robustness of “Connecting Characters Together”
CAPTCHAs

HAICHANG GAO, WEI WANG, YE FAN, JIAO QI AND XIYANG LIU
Software Engineering Institute
Xidian University
Xi’an, Shaanxi, 710071 P.R. China
E-mail: hchgao@xidian.edu.cn

CAPTCHA is now commonly used as standard security technology to tell computers and humans apart. The most widely deployed CAPTCHAs are text-based schemes. In this paper, we document how we have broken such a text-based scheme which uses the “connecting characters together (CCT)” principle. CAPTCHAs of this type can be classified into three types: CAPTCHA with overlap but no noise arcs; CAPTCHA with noise arcs but no overlap; and CAPTCHA without noise arcs and overlap. Yahoo!, Baidu CAPTCHA and reCAPTCHA were selected as representatives of the three types. The CCT CAPTCHA is effectively resistant to segmentation and recognition in the early attacks. In contrast to early works that recognized the text after segmentation, we combined recognition with segmentation to break the CCT CAPTCHA. Our method segments the text by extracting the recognized characters. The experiments show that our extraction and segmentation attack on Yahoo! CAPTCHA achieved a success rate of 78% and an overall (individual character recognition with OCR) success rate of 55%. The segmentation and recognition success rate of Baidu CAPTCHA was 34%. On reCAPTCHA we achieved 34% success rate.

Keywords: CAPTCHA, Yahoo!, Baidu, ReCAPTCHA, human interactive proof, segmentation attack, recognition attack

1. INTRODUCTION

The CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart), is now close to a standard security mechanism for defense against undesirable and malicious bot programs abusing resources on the Internet. Defense against malicious bots is important because of the scale of damage they can cause. For example, a malicious bot program can sign up thousands of free email accounts every minute and send out millions of junk emails. CAPTCHA, first introduced by Luis von Ahn et al. in 2003, is a program that generates and grades tests which most humans can pass but current computer programs cannot [1, 2]. This mechanism is widely adopted by commercial websites like Yahoo!, Google and Baidu.

Most commonly used CAPTCHAs are text-based CAPTCHAs which rely on the distortion of digits/letters and other visual effects added in the background image [3]. The popularity of such schemes is due to many advantages [4], for example, being intuitive to users world-wide (the user task requires character recognition), having few
localization issues (people in different countries all recognize Roman characters), can be generated rapidly when required and has good potential to provide strong security (e.g. if the scheme is properly designed the space a brute force attack has to search can be huge) [5].

It is widely accepted that a good CAPTCHA must be not only human friendly but also resistant to the computer programs trying to pass the challenge. The design goal was to prevent automated attacks achieving a success rate of higher than 0.01% [6].

It was demonstrated that computers are as good as or better than humans at single character recognition under all commonly used distortion and clutter scenarios by Human Interaction Proof (HIP) [7]. The success rate of recognizing typical distortion characters by computers is 95% [3]. Early research showed that a text-based CAPTCHA without the connecting characters can be segmented even though it is under different distortions and with noises [6]. Some other early works [8, 9] also suggested that CAPTCHA designers can no longer rely on the recognition problems to contribute effectively to the security of the CAPTCHA. As a result, whether attacking a CAPTCHA is successful or not is based on the task of segmenting the challenge text. For CAPTCHA programs which use the “connecting characters together” (CCT) principle, extracting the recognized characters is used to complete the segmentation.

Numerous websites adopted the CCT scheme to resist the segmentation method. The characters are connected with each other so that the early methods (recognizing the text after segmentation) [6] cannot extract any character from the image. The new scheme can be divided into three main types: CAPTCHA with overlap but no noise arcs; CAPTCHA with noise arcs but no overlap; and CAPTCHA without noise arcs and overlap. The representative of each CAPTCHA type selected in this paper is Yahoo! CAPTCHA, Baidu CAPTCHA and reCAPTCHA (see Fig. 1). The details are described in section 3.

(a) Yahoo!           (b) Baidu                   (c) reCAPTCHA
Fig. 1. The CCT CAPTCHA.

It is worth mentioning that reCAPTCHA is a free CAPTCHA service that helps to digitize books, newspapers and old time radio shows. It is applied to many famous websites, such as Facebook, Microsoft and Twitter. In reCAPTCHA, an unknown word and a control word are presented in random order. In Fig. 1 (c), the word underlined is the unknown word. The other one is the control word. The user is asked to read both words. If they solve the control word, the system assumes their answer is correct for the unknown word [10]. The weakness of this method is that even if the unknown word is not recognized correctly it does not influence the results. Therefore, as long as we break the control word, the reCAPTCHA will be broken at the same time. The same method is used for the unknown word and the control word.

If we know what and where one of characters is in the CCT CAPTCHA, this character can be extracted and erased. Then the challenge text may be segmented to two parts.
The more characters erased, the smaller the remaining parts are. If every part only contains one individual character, breaking the CCT CAPTCHA can be successfully reduced to a problem of individual character recognition.

In this paper, we report our techniques to attack the CCT CAPTCHA, which was carefully designed to be segmentation-resistant. The extraction and segmentation attack proposed in this paper has achieved 48% success rate on Yahoo!, 34% on Baidu and 34% on reCAPTCHA.

This paper is organized as follows. Section 2 discusses the related work. Section 3 displays the scheme of CAPTCHA using CCT principle. Sections 4, 5 and 6 describe our extraction and segmentation attacks on Yahoo!, Baidu and reCAPTCHA. Section 7 measures the success rate and then discusses limitations of our attack. Section 8 provides concluding remarks.

2. RELATED WORKS

Using sophisticated object recognition algorithms, Mori and Malik [11] successfully attacked the EZ-Gimpy and the Gimpy CAPTCHA with a 92% and 33% success rate respectively. Chellapilla and Patrice Simard attacked several CAPTCHA taken from the Web by using machine-learning algorithms, achieving success rates from 4.89% to 66.2% [12]. Moy and his colleagues developed distortion estimation techniques to break EZ-Gimpy with a success rate of 99% and four-letter Gimpy-r with a success rate of 78% [13]. Yan has broken most visual schemes provided at Captchaservice.org, a publicly available web service for CAPTCHA generation, with a success rate nearly 100% by simply counting the number of pixels of each segmented character, although these schemes were all resistant to the best OCR software on the market [8]. They also presented new character segmentation techniques of general value to attack a number of text-based CAPTCHA in 2008, including the earlier mechanisms designed and deployed by Microsoft, Yahoo! and Google, and have achieved a segmentation success rate of 92% against Microsoft CAPTCHA [6]. In 2011, applying a systematic evaluation methodology to 15 current CAPTCHA schemes from popular web sites, Elie Bursztein, Matthieu Martin, and John C. Mitchell demonstrated that 13 of them were vulnerable to automated attacks [14]. The lowest success rate was 5% on Baidu. The remaining two CAPTCHA schemes are Google and reCAPTCHA, with a success rate of 0%.

The new CAPTCHA scheme used by Yahoo!, Baidu and reCAPTCHA provides good security using distortion and the CCT principle as shown in Fig. 1. To date, we have seen no research indicating that these new schemes have been attacked effectively.

3. THE CAPTCHA SCHEME

3.1 Classification of the CCT CAPTCHA

For CAPTCHA using CCT principle, there are four key factors we should focus on:

- How many characters are used in a challenge? The more characters are used, the more difficulties we will face. Breaking a fixed-length challenge is easier than breaking a
random length challenge.

- Do noise arcs exist? In CAPTCHAs using CCT principle, the noise arc increases the difficulty of our break work, meanwhile it decreases the usability. For an attack on CAPTCHA using noise arcs, in preprocessing we attempt to eliminate the noise arcs.
- Do the neighbors overlap or just connect with each other? In our procedures, the recognized characters will be cut off. If the neighbors overlap with each other, the neighbor’s body will be randomly cut.
- Are the locations of the characters distributed according to a regulation rule? If the characters distribute according to their location, we can draw guidelines to help to recognize some characters or draw a line to segment the challenge (see Fig. 8).

A well designed CAPTCHA should not be solved by current computer programs or bots, but should be easily solved by humans. Noise arcs interrupt the recognition and the overlaps lead to incorrect segmentation for computer programs or bots. However, we did not find any current CAPTCHA using both noise arcs and overlap. The reason is that the usability level is decreased when these two characteristics are used in a CAPTCHA.

We consulted the Alexa list of most used websites [15] to analyze numerous CCT CAPTCHAs. According to the characteristics, we can classify the CCT CAPTCHA into three types: the CAPTCHA with overlap but no noise arcs (see Fig. 2 (a)); the CAPTCHA with noise arcs but no overlap (see Fig. 2 (b)); and the CAPTCHA without noise arcs and overlap (see Fig. 2 (c)).

(a) The CAPTCHA with overlap but no noise arcs (Yahoo!, Google).
(b) The CAPTCHA with noise arcs but no overlap (Baidu, Tencent).
(c) The CAPTCHA without noise arcs and overlap (Taobao, reCAPTCHA).

Fig. 2. The CAPTCHAs.

After analysis and comparison, three CAPTCHAs were selected to be representatives of the CAPTCHAs using CCT principle, which are Yahoo!, Baidu and reCAPTCHA.

The representative of the CAPTCHA with overlap but no noise arcs is Yahoo!. The characters are arranged with overlap in Yahoo! CAPTCHA. They are distributed according to a regulation requiring no blank space between neighbors. The representative of CAPTCHA with noise arcs is the Baidu CAPTCHA. The characters in Baidu are vertically placed with random rotation, size and location. Neighboring characters are connected, but with fewer connections when compared with Yahoo!. The representative of the CAPTCHA without noise arcs and overlap is reCAPTCHA. The characters are arranged more tightly but without overlap in reCAPTCHA.

Table 1 shows the features of the Yahoo! CAPTCHA, Baidu CAPTCHA and reCAPTCHA. The specific characteristics of each CAPTCHA will be described next.
Table 1. CAPTCHAs features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Yahoo!</th>
<th>Baidu</th>
<th>reCAPTCHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of the challenge</td>
<td>6-8</td>
<td>4</td>
<td>6-8</td>
</tr>
<tr>
<td>Noise arcs</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Overlap</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>With regulation</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

3.2 Characteristics of the CCT CAPTCHA

“Connecting characters together” is the process of merging characters horizontally. Humans perform well in segmenting connected characters, but this is still a problem for current computer algorithms [3]. In this section, the specific characteristics of the three CAPTCHAs are described. According to the analysis of the characteristics, we propose a method to segment the CCT CAPTCHA.

Specialized computer vision algorithms had success breaking the early Yahoo! CAPTCHAs. Yan broke the challenge texts in the early scheme by “even cut” [6]. A new advanced security mechanism was developed to resist attack. Fig. 1 (a) shows some examples now used by Yahoo!. Both distortion and connection help the Yahoo! CAPTCHA scheme provide strong segmentation-resistance.

As we have no access to the Yahoo! scheme codebase, so we collected 100 random samples from Yahoo!’s website. By analyzing the samples, we observed that the Yahoo! scheme (as deployed) has the following characteristics:

- Six to eight characters are used in each challenge.
- There is no noise arc exists.
- Only 9 upper cases, 12 lower cases and 7 digits are used in the challenges. The probability they appeared in the challenge varies from 1.84% to 5.52% with an average value of 3.57%. In order to guarantee usability, the remaining 33 alphanumeric characters are not used. For example, it is not easy for humans to distinguish I from 1.
- Characters typically connect or overlap with neighbors.
- Characters are placed in accordance with their location at the guide lines (see Fig. 7).

As with the Yahoo!, we collected 100 random samples from Baidu’s website and observed that the Baidu scheme has the following characteristics:

- Four characters are used in each challenge.
- There are noise arcs exist.
- Only 10 upper cases, 5 digits are used in the challenges. The probability they appeared in the challenge varies from 3.11% to 8.37% with an average of 5.72%. No lower case usage.
- Characters usually connect with neighbors.

A reCAPTCHA challenge contains two words. We call the word whose answer is already known “control word” and the new word “unknown word” [16]. We concentrated on the control word for the reCAPTCHA only verify the control word. Characteristics of the control word in reCAPTCHA are:
Six to eight characters are used in each challenge.

There is no noise arc exists.

Only 11 upper cases, 25 lower cases are used in the challenges. No digits are used. Table 2 shows the probability of each character in the control word of reCAPTCHA. Some characters occur more frequently, e.g. a, e, i, l, o, r, s, and t. The reason is that the words displayed in reCAPTCHA are taken from scanned text by OCR [16], so the probability of vowel selection is high. We will focus on attacking these common characters.

Characters commonly connect with but do not overlap with their neighbors.

Some characters occur more frequently, e.g. a, e, i, l, o, r, s, and t. The reason is that the words displayed in reCAPTCHA are taken from scanned text by OCR [16], so the probability of vowel selection is high. We will focus on attacking these common characters.

### Table 2. The probability of each character.

<table>
<thead>
<tr>
<th>Character</th>
<th>Probability(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>7.11</td>
</tr>
<tr>
<td>b</td>
<td>1.78</td>
</tr>
<tr>
<td>c</td>
<td>4.00</td>
</tr>
<tr>
<td>d</td>
<td>3.70</td>
</tr>
<tr>
<td>e</td>
<td>10.1</td>
</tr>
<tr>
<td>f</td>
<td>1.63</td>
</tr>
<tr>
<td>g</td>
<td>2.51</td>
</tr>
<tr>
<td>h</td>
<td>3.11</td>
</tr>
<tr>
<td>i</td>
<td>8.44</td>
</tr>
<tr>
<td>j</td>
<td>0.15</td>
</tr>
<tr>
<td>k</td>
<td>1.18</td>
</tr>
<tr>
<td>l</td>
<td>6.07</td>
</tr>
<tr>
<td>m</td>
<td>1.48</td>
</tr>
<tr>
<td>n</td>
<td>4.44</td>
</tr>
<tr>
<td>o</td>
<td>8.89</td>
</tr>
<tr>
<td>p</td>
<td>2.07</td>
</tr>
<tr>
<td>q</td>
<td>0.15</td>
</tr>
<tr>
<td>r</td>
<td>7.25</td>
</tr>
<tr>
<td>s</td>
<td>8.44</td>
</tr>
<tr>
<td>t</td>
<td>8.29</td>
</tr>
<tr>
<td>u</td>
<td>2.67</td>
</tr>
<tr>
<td>v</td>
<td>0.74</td>
</tr>
<tr>
<td>w</td>
<td>0.44</td>
</tr>
<tr>
<td>y</td>
<td>2.96</td>
</tr>
<tr>
<td>z</td>
<td>0.44</td>
</tr>
<tr>
<td>A</td>
<td>0.15</td>
</tr>
<tr>
<td>B</td>
<td>0.15</td>
</tr>
<tr>
<td>D</td>
<td>0.15</td>
</tr>
<tr>
<td>E</td>
<td>0.15</td>
</tr>
<tr>
<td>F</td>
<td>0.15</td>
</tr>
<tr>
<td>H</td>
<td>0.15</td>
</tr>
<tr>
<td>J</td>
<td>0.15</td>
</tr>
<tr>
<td>M</td>
<td>0.15</td>
</tr>
<tr>
<td>R</td>
<td>0.15</td>
</tr>
<tr>
<td>T</td>
<td>0.15</td>
</tr>
<tr>
<td>W</td>
<td>0.15</td>
</tr>
</tbody>
</table>

### 4. OUR ATTACK ON YAHOO!

In Yahoo! CAPTCHA, characters connect with each other horizontally. Utilizing the analysis noted in section 3, we have developed an extraction and segmentation attack that can effectively segment challenges generated by the Yahoo! CAPTCHA.

Our attack on Yahoo! can be divided into three steps. First, remove the noise pixels and get the binaryization image, then identify and erase some characters to divide the challenge, and finally, segment the remaining large chunks which contain two or more characters.

Our attack is built on observing and analyzing a sample set which contains 100 random samples. To check the effectiveness of our attack, 100 other random samples were also tested. This methodology follows the common practice in the computer vision and machine learning fields.

![Fig. 3. Pre-processing of the Yahoo! CAPTCHA.](a) ![Fig. 3. Pre-processing of the Yahoo! CAPTCHA.](b)

#### 4.1 Pre-processing

A few scattered pixels (marked by red pixels in Fig. 3 (a)) surround the characters in the Yahoo CAPTCHA. We cleared these scattered pixels using a threshold method: pix-
els with intensity higher than a threshold value are converted to white and pixels with lower than the threshold value are converted to black. The threshold, which is manually determined by analyzing the sample set, is set to (150, 150, 150) in RGB triplet. The same value is used for each image in both the sample and test sets. After this step, the binaryization image is illustrated in Fig. 3 (b).

### 4.2 Extraction and Segmentation

There are two steps in extraction. First, find what and where the characters are. Then erase these characters on the basis of their shape and size. In our attack, if the first step is accomplished, the location of the character is determined. Finally we can complete the erasure. The details of erasing the characters are described in section 4.3.

In this part, we use the following five methods to accomplish the extraction: Cut head and tail (CHT); guide lines; loop principle; projection with guide lines; and even cut. It is not necessary to use every method when we break the three CAPTHCAS. The appropriate methods can be chosen depending on the characteristics of each CAPTCHA. In our attack on Baidu, we used CHT. In the attack on reCAPTCHA, CHT and loop principle are both used. Each of these methods is described in detail in the following sections.

| Table 3. These characters can be recognized by the feature of projection only. |
|---|---|---|---|---|
| Characters | Example | 2 | 3 | 4 | 5 |
| Example | | 2N | 3F4 | 5H | |
| Characters | Example | | 7H | 8 | 1 | b |
| Example | | 7H | 8H | 1 | b |
| Characters | Example | J | p | s | T |
| Example | | Ju | pM | s | T |
| Characters | Example | V | w | y | z |
| Example | | VM | w | y | z |

#### 4.2.1 Cut head and tail (CHT)

As the first and the last characters both have one neighbor, and therefore no matter how sophisticated the distortion of the first and the last characters is, the horizontal projections of the same character in different challenge texts have the same characteristics. Therefore, we can recognize and erase them by analyzing the exposed part. We call this method ‘Cut head and tail’ (CHT). Horizontal projection and Area Color Filling (ACF, the directions and size of color filling are limited) are used in the CHT. The operation for the head character is taken as an example.

The process of CHT starts by mapping the horizontal projection. In the horizontal projection, the number of each row’s blue pixels represents the distance between the first black pixel in the row and the far left foreground pixel in the image. If the distance is
larger than 40 pixels, it is set to 40 for the width of a character is typically less than 40.

Table 3 shows the characters which can be recognized according to their distinct projection characteristics. The recognition process of character 3 and \( V \) are described as examples. The projection of 3 has three valleys and two peaks (see Fig. 4 (a)). They can be detected by line-by-line scanning. The valley is the inflection point where its projection value is smaller than neighbors. The peak refers to the inflection point where its projection is larger than neighbors.

The projection of \( V \) has a slope and the width between the top and bottom is larger than 10 pixels (Fig. 4 (b)). We can recognize the character \( V \) by the above characteristics. The other characters in Table 3 also have easily identifiable features.

![Fig. 4. Identify the first letter by its projection characteristics.](image)

**Table 3. The characters which can be recognized according to their distinct projection characteristics.**

<table>
<thead>
<tr>
<th>Characters</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>F, H, L and M</td>
<td>F, H, L, M</td>
</tr>
<tr>
<td>c, d, and e</td>
<td>c, d, e</td>
</tr>
<tr>
<td>n, u and r</td>
<td>n, u, r</td>
</tr>
<tr>
<td>6 and G</td>
<td>6, G</td>
</tr>
</tbody>
</table>

However, some characters with similar features cannot be distinguished using only projection (see Table 4).

We take the first group of Table 4 (F, H, L, and M) as an example. For this step, we distinguish F H L and M with the help of ACF. The algorithm consists of two parts: upper ACF and lower ACF. Fig. 5 provides examples. The color red is used to fill the upper area, and green to fill the lower area of the character. This means we can regard F as if the upper red area is besieged and the lower green area is not. If the red area overlaps the green area, and has an unlimited upper red area, it is L. Figs. 5 (b) and (d) has a similar characteristic that the red and green areas are both unlimited. However there is an obvious difference we can use to distinguish them. The lower boundary of red area is higher than the upper boundary of green area, therefore we regard it as \( H \). Otherwise it is M. We can recognize each of them relying on these characteristics and then remove these characters from the original images.

![Fig. 5. Examples of upper ACF and lower ACF.](image)
4.2.2 Guide lines principle

In Yahoo! CAPTCHA, the characters are placed according to their locations at guide lines in Fig. 6. Guide lines consist of four lines: ascender line, mean line, base line, and descender line. We draw the mean line and the base line to help break the challenge.

Sample analysis shows that the challenge texts are globally warped by cosine mapping with random amplitude. Therefore, the guide lines should also be warped to meet the text so that the challenge text can be divided into three layers, top, middle and bottom (see Fig. 7). The base line is the line close to the bottom edge of the challenge text. It includes the following characteristics:

- Located in the lower half of the CAPTCHA image.
- The base line is smooth. Violent ups or downs are not accepted.

According to these characteristics, the guide lines are drawn as the following descriptions. First, from the left-most foreground to the right-most foreground, we draw horizontal lines every fixed-length pixels. The vertical position of each line is determined by the lowest foreground pixel in the corresponding columns. The length of the horizontal lines is 7 pixels in our experiment (see Fig. 7 (a)). Second, assign a value to every horizontal line. The value is the sum of the differences between its coordinate values and coordinate values of its four nearest neighbors. The larger the value is the more protruding the line’s location is. Then draw a line to connect the lowest foreground pixels of every fixed-length horizontal line whose value is lower than an experiential threshold [17]. The remaining horizontal lines whose values are higher than the threshold are ignored. Finally, extend the left-most and right-most points to both ends horizontally. The original horizontal lines are modified to our expected base line (see Fig. 7 (b)). The mean line is drawn by lifting the base line with a given value, which is 4/5 of the largest vertical distance of the highest gray pixel in every column to the base line.
Fig. 7. Draw the base line and guide lines.

The challenge text is divided into three parts (top, middle and bottom layers) by the base line and mean line. In the Yahoo! CAPTCHAs, it is observed that upper cases and digits are located in the top and middle layers. $b$ and $d$ are also located in the top and middle layers. Lower case letters $p$ and $y$ are located in both middle and bottom layers, while other lower cases are in the middle layer. We defined these three categories by different distributions of the characters (see Table 5). Category I are the characters written in the top and middle layers. Category II are the letters written in the middle. Category III are the letters written in both the middle and bottom layers.

In Table 5, only $p$ and $y$ belong to the category III. The descender of $p$ and $y$ extend beyond the base line (see Fig. 8 (a, b)). We can find the location of $p$ and $y$ by detecting the gray pixels under the base line.

<table>
<thead>
<tr>
<th>Category</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Upper cases, digits, $b$, and $d$</td>
</tr>
<tr>
<td>II</td>
<td>$c$, $e$, $m$, $n$, $r$, $s$, $u$, $w$, $z$</td>
</tr>
<tr>
<td>III</td>
<td>$p$, $y$</td>
</tr>
</tbody>
</table>

Fig. 8. The instances of $p$ and $y$. The “loop” of $p$ is the key to distinguish $p$ from $y$. With the color filling algorithms, the loop is marked as shown in Fig. 8 (c). After the locations of $p$ and $y$ are identified, they can be extracted from the image. The method of extraction is described in section 4.3.

The guide lines model is the key point of our attack on Yahoo!. In the following two sections, Loop principle and Projection are carried out based on this model.

4.2.3 Loop principle

The CCT process for the Yahoo! CAPTCHA scheme creates numerous loops in the challenge text. With the location, size and shape of the loops we can judge whether the loops belong to an individual character or the gap between two characters.

For example, if two loops are located in a vertical direction and the projections of the two loops in a horizontal direction do not overlap, we consider the two loops to be-
long to an 8 (see Fig. 9 (a)). We find that A holds the same characteristics when we draw the base line on the image. Fig. 9 (b) shows an example. In Fig. 9 (b), if two adjacent triangle-loops appear in a horizontal direction and the bottoms of them are next to the base line, it is regarded as M.

![Fig. 9. Some characters can be recognized with the help of loops.](image)

Some loops occur in specific circumstances. For instance, when e, z and s are connected to their neighbors, two small loops will be formed in a vertical direction and both of them are located in middle part of the guide lines. After the extraction, OCR software is used to further distinguish them. We have summarized some typical cases, shown in Table 6.

In this step, not all the e, z and s characters are detected. Only those with the characteristics listed in Table 6 can be detected and segmented from the CAPTCHA.

With the above procedures, several connected characters will be separated. The challenge text can be segmented into several chunks. By observing the 100 challenges sample set, 9 challenges are separated into three chunks, 11 challenges are separated into four chunks, 30 are separated into five chunks, 22 are separated into six chunks, 20 challenges are separated into seven chunks and 8 challenges are separated into eight chunks. It is reasonable to say that after this step processing, many of challenges that been separated into six to eight chunks have been segmented completely.

<table>
<thead>
<tr>
<th>Image</th>
<th>Characters</th>
<th>Characteristics of the loop</th>
</tr>
</thead>
</table>
| ![Image](image) | e, z or s | • There are two loops where the projections intersect in a vertical direction  
• Both of the loops are located in middle layer of the guide lines |
| ![Image](image) | The gap between characters | • The maximum width is less than 7 pixels.  
• Located in the upper and middle layers. |
| ![Image](image) | 6 or b | • The width is between 10 and 30 pixels.  
• Located in the middle layers.  
• There are foreground pixels at the upper layer. |

### 4.2.4 Projection with guide lines

The projection technique in this phase is based on analyzing the projection characteristics of the CAPTCHAs. We projected the top and middle layers to the mean line and base line separately, as shown in Fig. 10. The upper blue area is the projection above the
mean line. The lower blue area expresses the projections of the middle layer between mean and base lines.

Next, two slide windows are used on the two projection layers. A red rectangle with the width of 5 pixels and height of 5 pixels slides on the upper projection layer and a green rectangle with the width of 5 pixels and height of 8 pixels slides on the lower projection layer (see Fig. 10). Move the two slide windows from left to right simultaneously. While there is no pixel in the red rectangle and all of the X-axis projections values are less than 8 in the green window, it is possible, but not certain that an $L$ can be identified. However, if the foreground pixels of this layer are all close to the base line, an $L$ is identified, seen by the pink line in Fig. 10.

If the projection values are between 0 and 5 pixels in the red window and less than 8 pixels in the green window, there may be a $G$. If the foreground pixels of this part are all close to the base line, a $G$ is detected, marked by the green line in Fig. 10.

The projection of $H$ also has characteristics similar to $L$ and $G$. However, with global warping some projections of $H$ do not possess an obvious valley so it is difficult to detect. To enhance the characteristic, the oblique-projection is used. We observed that some letters slant to the left or right in Yahoo! CAPTCHA. Therefore, we used the oblique line, at degrees either 25 or $-25$, to analyze the sample set to recognize the $H$ on the basis of the characteristic (see Fig. 11). Then the detected $H$ can be erased.

Analyzing the 100 challenges sample set, 5 challenges are separated into three chunks, 6 challenges are separated into four chunks, 16 are separated into five chunks, 31 are separated into six chunks, 29 challenges are separated into seven chunks and 13 challenges are separated into eight chunks.

4.2.5 Even cut

Although the extraction method erased some characters, there are still several remaining chunks (see Fig. 12). Consequently, our algorithm must estimate how many characters there are in each chunk and then divide the chunks which contain more than one character into individual characters.
We analyzed the characters in the remaining chunks and counted their pixels (see Table 7). The result shows that if the pixels are fewer than 282, it must be a single character. Between 282 and 350, there are three situations, an upper case or digit, two lower cases and a lower case with a J. With the help of guide lines and projection, the number of characters in a chunk whose pixels are between 282 and 350 can be identified (see Fig. 13). The single characters whose pixels are more than 350 are M, G, H, which have been extracted in the early steps. So the remaining chunks whose pixels are from 350 to 611 contain two characters. The chunks whose pixels are between 620 and 830 contain three characters and the chunks whose pixels are from 834 to 1090 contain four characters. Therefore the number of characters contained in each chunk can be confirmed and the even cut method is available to divide them into individual characters [6].

The results of even cut are shown in Fig. 14. The chunks containing two characters can be segmented.

Table 7. The probability and pixels of the remaining chunks.

<table>
<thead>
<tr>
<th>Characters in remaining chunks</th>
<th>Probability (%)</th>
<th>Minimum pixels</th>
<th>Maximum pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>63.6</td>
<td>123</td>
<td>350</td>
</tr>
<tr>
<td>2</td>
<td>26.1</td>
<td>282</td>
<td>611</td>
</tr>
<tr>
<td>3</td>
<td>6.1</td>
<td>620</td>
<td>830</td>
</tr>
<tr>
<td>4</td>
<td>4.2</td>
<td>834</td>
<td>1090</td>
</tr>
</tbody>
</table>

Fig. 13. (a) Two lower cases have no pixel above the mean line; (b) A lower case and a J have a few pixels above the mean line; (c) A digit or an upper case has more pixels above the mean line.

Fig. 14. Even cut to the remaining chunks.

4.3 Extraction

In this step, the recognized characters will be cut from the challenge according to their shapes. We take the p and H as examples, and other characters can also be cut with this method.
In Yahoo! CAPTCHA, the font and style of character varies. Some characters are bold and their strokes are wider than the regular style. Therefore, the width of the character should be determined first. After analyzing the samples, it was found that the stroke width of one character changes little. With this information we can cut the recognized characters on the basis of their shapes.

A black-and-white image is used in the recognized phase. But in the extraction phase, the pixels above the threshold are converted to white and others are kept intact because the edge of the stroke can be identified with the RGB triplet. In Yahoo! CAPTCHA, the edge of the stroke is composed of light gray pixels whose RGB triplet is more than (60, 60, 60).

The extraction procedure consists of three steps. First, find the stroke-width of the recognized character. Second, divide the recognized character into three parts: top, middle and bottom part. Finally, cut the foreground pixels of each part separately according to the character’s features.

- **Extract p**
  
  p is detected with help of guidelines, its stroke-width is two more than the number of pixels that intersect with the base line. In Fig. 15 (a), the yellow line is the intersection. The p is divided to three parts: top, middle and bottom layer. The top layer is the rows upon the third row of the loop. The bottom layer is the rows under the last three rows of the loop. The remaining rows are in the middle layer (see Fig. 15 (b)). We use different processes with different layers.

  ![Fig. 15. The process of extracting p.](image)

- **Extract H**
  
  After H is detected, along the slant line from the base line to the mean line, find the shortest stroke in the left and right of the slant line. In Fig. 16 (a), the yellow line is the shortest stroke. The stroke-width of this H is three more than the shortest stroke. Then H is divided into three layers. The top layer is the first three rows, the bottom is the last three rows on the base line. Other rows are the middle layer (see Fig. 16 (b)). Cut stroke-width foreground pixels on both sides of the slant line in the middle layer. For the trans-
verse line of $H$, cut it according to the row under the transverse line. The transverse line’s edge is considered the same as the row under the transverse line. The judgment and process method of overlap is similar to the methods described for extracting $p$.

The method of extracting other characters is similar to $p$ and $H$. Table 8 shows the process of extracting and segmenting the characters. Our process order is CHT, guide lines, loop principle and, projection with guide lines.

Table 8. The process of extracting and segmenting the characters.

<table>
<thead>
<tr>
<th>The original image</th>
<th>CHT</th>
<th>guide lines</th>
<th>loop principle</th>
<th>projection with guide lines</th>
<th>even cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>opGiH2</td>
<td>opGiH2</td>
<td>opGiH2</td>
<td>opGiH2</td>
<td>opGiH2</td>
<td>opGiH2</td>
</tr>
<tr>
<td>pAfiH5</td>
<td>pAfiH5</td>
<td>pAfiH5</td>
<td>pAfiH5</td>
<td>pAfiH5</td>
<td>pAfiH5</td>
</tr>
</tbody>
</table>

5. ATTACK ON Baidu

In Baidu CAPTCHA, noise pixels exist in the background of the image (see in Fig. 17 (b)). The first step is pre-processing to remove the noise pixels. In this step, the noise pixels are removed using a threshold set to (200, 200, 200) in RGB triplet. Pixels with higher than the threshold are converted to the background color. Fig. 17 (c) shows the image after pre-processing.

![Image](image1.png)

(a) The original image. (b) The noise pixels. (c) The image after pre-processing.

Baidu CAPTCHA adds noise arcs which must be removed prior to segmentation. Analysis showed that these noise arcs are dissimilar to the characters, which are surrounded by light grey pixels, (Fig. 18 (b)). On the contrary, there are no light gray pixels surrounding the noise line (Fig. 18 (c)).

![Image](image2.png)

(a) (b) (c)

Fig. 18. The difference between characters and the noise arc.

According to this characteristic, the noise arcs can be recognized. First, mark the
pixels which have no light gray pixels surrounding their neighbors with yellow pixels (Fig. 19 (a)). Then, find a linear path containing most of the marked-pixels chunks. Finally, clear the marked pixels in the linear path (Fig. 19 (b)). After experimenting on the 100 challenges sample set, the noise arcs of 86 challenges were eliminated successfully.

![Fig. 19. Remove the noise line.](a) ![Fig. 19. Remove the noise line.](b)

By analyzing the characteristics and horizontal projection of the first and last characters, CHT can also be used to break Baidu as Yahoo!. Y and 6 can be recognized and removed on the basis of their characteristics. After the first CHT, the remaining characters can be recognized by using the CHT once more. The specific method was introduced in the Yahoo! description.

In conclusion, Baidu CAPTCHA can be broken successfully with above methods. In the sample of 100 challenges, the first CHT achieved a 91% success rate. The success rate of the second CHT was 69%. The overall success rate is 34% \((0.91^2*0.69^2*0.86)\).

# 6. ATTACK ON RECAPTCHA

After analyzing the characteristics of the control word, the summary steps for reCAPTCHA attack are as follows:

- Separate the two words to create two pictures and process them separately.
- Find the gradient of the control word.
- Use CHT to identify the first and last characters and then remove them.
- Use “Loop principle” to identify and remove the characters with distinct characteristics.

A reCAPTCHA challenge contains two words. The first step is to separate the two words into two separate images. We then process the images using the threshold method (see in Fig. 20). The threshold is set to (200, 200, 200) in RGB triplet. Pixels with higher than 200 value are converted to the background color.

![Fig. 20. Separate the two words.](a) ![Fig. 20. Separate the two words.](b)

Analysis of 100 challenges showed that all of the characters in the reCAPTCHA challenge slant either right or left with a 20 to 40 degree angle. The gradient of each control word can be confirmed. First, draw the oblique lines for every four degree angle.
Second, project the foreground pixels in each oblique line direction and record their maximum projection values. Finally, compare the maximum projection values (Figs. 21 (a)-(c)) and find the highest value (Fig. 21 (b)). The angle which is associated with the highest value is the gradient of the control word.

![Fig. 21. Projection values for different angles.](image)

After finding the gradient of the control word, we identify the first and last characters of the control word using CHT, as we did with Yahoo! (see Fig. 22). Then the first and last characters are cut from the control word with the help of the oblique lines. The details will be described in following part. In the sample of 100 challenges, CHT achieved a 93% success rate.

Table 2 demonstrates that the probabilities of $a$, $e$, $i$, $o$, $s$ and $t$ are much higher than other letters, therefore we break them first due to their high appearance rate. The ‘Loop principle’ method can be used to identified $a$, $e$ and $o$. An $i$ has an independent point so that it can be recognized with high success rate.

![Fig. 22. Identify $a$, $e$, $o$ and $i$.](image)

Similarly, a character with one or more loops, such as $d$, $g$, $p$ and so on can also be recognized using the “Loop Principle” (see in Fig. 23). We test the method in the sample set. The success rate of recognizing the characters with loops is 95%.

After we confirm the gradient of the control word and recognize the characters, the recognized characters can be cut. When the control word is slanted left or right, we used the oblique line to segment the characters. According to the shape of the characters, the edges of the characters can be discovered. After observing the value of pixels in re-CAPTCHA, it was found that the color of the edge pixels stroke is light gray with an RGB triplet more than $(50, 50, 50)$. In the projection image (see Fig. 21 (b)), if one column is less than the neighbors with a sharp decrease it may be an edge. The edge is often on the right or left side of the column which has a sharp increase compared with its neighbors. According to the shape of the characters and with the help of projection, the edge of the character can be confirmed. In Fig. 23 (a), the cut point of $s$ and $b$ are con-
firmed with a red pixel. Then we draw the segmentation oblique line passing through the cut point (see Fig. 23).

By means of loop principle and CHT, characters which are previously recognized can be extracted with the oblique line segmentation. However, there are still a few characters left in the image. CHT can be sequentially applied to the first and last characters of the left image, using the same attack method as we used for Baidu. Table 9 shows the procedure of attack. reCAPTCHA is successfully broken with the above methods.

<table>
<thead>
<tr>
<th>The original image</th>
<th>The image after CHT</th>
<th>The image after identifying special characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>sessar</td>
<td>sessar</td>
<td>sessar</td>
</tr>
<tr>
<td>lenndm</td>
<td>lenndm</td>
<td>lenndm</td>
</tr>
<tr>
<td>exista</td>
<td>exista</td>
<td>exista</td>
</tr>
</tbody>
</table>

7. THE RESULTS AND DISCUSSIONS

We tested our algorithm on Yahoo!, Baidu and reCAPTCHAs, and compared the results with those of Stanford University, which proposed a method to attack many CAPTCHAs [14]. We also analyzed the attack failures in this section.

7.1 Success Rate

To examine our attack, we collected 100 random CAPTCHAs as the sample set and another 100 random CAPTCHAs as the test set for Yahoo!, Baidu and reCAPTCHA.

<table>
<thead>
<tr>
<th>CAPTCHA</th>
<th>Our attack</th>
<th>Stanford attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo!/ Google</td>
<td>55% (Yahoo)</td>
<td>0% (Google)</td>
</tr>
<tr>
<td>Baidu</td>
<td>34%</td>
<td>5%</td>
</tr>
<tr>
<td>reCAPTCHA</td>
<td>34%</td>
<td>0%</td>
</tr>
</tbody>
</table>

In Yahoo! CAPTCHA, our extraction and segmentation attack achieved an 80% success rate on the sample set. That is, 80 out of 100 challenges were extracted and segmented correctly. The overall (recognition with OCR software) success rate is 56% (0.8 * 0.95^7). Trial attack was used on another 100 CAPTCHAs to check the generality of our attack. These CAPTCHAs were not analyzed previously when we designed our attack. We gained the results of 78% extraction and segmentation success rate and a 55% overall success rate. For both the sample and test sets, the success rate is satisfactory.

In Baidu CAPTCHA, our attack achieved a 40% success rate on the sample set and
34% on the test set. In reCAPTCHA, the success rate is 36% on the sample set and 34% on the test set. Table 10 shows the comparison between our result and the result of Stanford University. In Table 10, Yahoo! and Google are similar, so we used them as comparison. We can see that our attack is more effective for the CCT CAPTCHA.

7.2 Extraction and Segmentation Failure Analysis

We analyzed all cases of the failures for our extraction and segmentation attack methods in both the sample and test sets, and found that four types of failure occurred as follows.

7.2.1 Failure of “Pre-processing”

In pre-processing, removing the noise pixels and arcs are fundamental for the next attack steps. Some examples are shown in Fig. 24. In Figs. 24 (a) and (b), $z$ and $d$ are broken after the pre-processing. They will not be accurately identified. In Baidu CAPTCHA, if the noise arcs were not eliminated completely, the characters may be identified incorrectly (see Fig. 24 (c)).

![Fig. 24. The failures in pre-processing.](image)

7.2.2 Failure of “CHT”

For characters which confuse humans, CHT cannot distinguish them either. For instance, it is difficult to distinguish the head character whether it is $r$ or $n$ (Fig. 25 (a)). In Fig. 25 (b), there are two loops of $S$ formed incorrectly. It could be identified as an 8.

![Fig. 25. Some confused character.](image)

7.2.3 Failure of “Loop principle”

This failure lies in incorrectly identifying loops. For example, if a 6 is connected to the neighbors, two loops may be detected. The example is shown in Fig. 26 (a). 6 may be incorrectly recognized as 8. In Fig. 26 (b), because a loop of $h$ is formed, it may be recognized as $b$.  

![Fig. 26. Failure of “Loop principle”.](image)
7.2.4 Failure of “draw the segmentation oblique line”

To draw the segmentation oblique line, the cut point and the angle are crucial. The failure is shown in Fig. 27. If the oblique angle of \( l \) is different from other characters, it will be segmented incorrectly.

7.2.5 Failure of “Even cut”

In this failure, “even cut” is not always effective in segmenting a remaining chunk containing more than two characters (see Fig. 28). Some characters are broken by even cut, with a proportion in the remaining chunks of approximately 10%. It will not seriously affect identification.

7.3 Principles for Designing a Secure CAPTCHA

The success of our “extraction and segmentation” methodology in identifying the CCT principle CAPTCHA suggests some new general principles for CAPTCHA design.

- **Resist the loop principle**: the loop can be used as a discriminator by the attacker. The location, shape, size and relative position of the loops can help the attacker identify particular characters. Breaking the loop (Fig. 24 (b)) and forming erroneous loops (Fig. 26) can strengthen the security of CAPTCHA to resist the loop principle. However, it will reduce the success rate for human identification in turn.
- **Resist the slant line cut**: all of the characters in a CAPTCHA should not slant in the same direction with the same angle. The use of projection attack makes this previously secure method now insecure.
- **Resist noise line detection**: the stroke width and the color of the noise line in a CAPTCHA should be as same as the character’s. However, this method will increase the solution difficulty for the user.

After many years of development, many varied attack methods have been used for text-based CAPTCHA. The CAPTCHA design faces serious challenges in keeping the balance between usability and security. Many current CAPTCHAs are resistant to auto attacks at the cost of usability. It may irritate users, resulting in users avoiding the website. Compared to text-based CAPTCHA, the image-based CAPTCHA is more human-friendly. Bin B. Zhu *et al.* proposed a new image-based CAPTCHA scheme, which disables machine-learning attacks [18]. The limitation of the scheme is that the generation time needs 122s per CAPTCHA. With the development of sophisticated processors and algorithms, the generation time may adapt to the web application. The image-based CAPTCHA may be the new standard CAPTCHA for networks.

8. CONCLUSION

In this paper, five main methods are used to extract and segment the CCT CAPTCHA. Yahoo!, Baidu, and reCAPTCHA are selected as the representatives. CHT, guide lines principle, loop principle, projection with guide lines, and even cut are used to extract the characters and segment the whole text into individual characters. Finally, each part is recognized by the OCR technique. The key step of the proposed approach is extraction. We achieved an extraction and segmentation success rate of 78% and an overall (recognition with OCR software) success rate of 55% on Yahoo!. When we attacked Baidu CAPTCHA, removing the noise line and CHT were the important steps. An 86% success rate of line noise removal was achieved, with a recognition success rate of 34%. For reCAPTCHA, the success rate is 34%.

REFERENCES


Haichang Gao (高海昌) received the Ph.D. degree in the School of Electronic and Information Engineering, Xi’an Jiaotong University in 2006. He is now an Associate Professor in the Software School, Xidian University, Xi’an, China. He has published more than 40 papers in international conferences and journals. Now he is in charge of a project of the National Nature Science Foundation of China. His current research interests include computer security and intelligent computing.
Wei Wang (王伟) received the B.S. degree in Computer Science from Xidian University in 2011. He is now a master candidate in Computer Science at Xidian University, Xi’an, China. His current research interest is the security and usability study of CAPTCHA.

Ye Fan (樊晔) received the B.S. degree in Software Engineering from Xidian University in 2011. She is now a master candidate in Software Engineering at Xidian University, Xi’an, China. Her current research interest is the security and usability study of CAPTCHA.

Jiao Qi (齐娇) received the B.S. degree in Computer Science from Xidian University in 2012. She is now a master candidate in Computer Science at Xidian University, Xi’an, China. Her current research interest is the security and usability study of CAPTCHA.

Xiyang Liu (刘西洋) is a Professor in Xidian University and a member of ACM and CCF. He has published more than 30 papers. Currently, he leads the Software Engineering Institute at Xidian University. His research interests include computer security and trustworthy computing.