1-1-2014

A CAPTCHA scheme based on the identification of character locations

Duc Vu Nguyen
University of Wollongong, dvn108@uowmail.edu.au

Yang-Wai Chow
University of Wollongong, caseyc@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

A CAPTCHA scheme based on the identification of character locations

Abstract

CAPTCHAs are a standard security mechanism used on many websites to protect online services against abuse by automated programs, or bots. The purpose of a CAPTCHA is to distinguish whether an online transaction is being carried out by a human or a bot. Unfortunately, to date many existing CAPTCHA schemes have been found to be vulnerable to automated attacks. It is widely accepted that state-of-the-art in text-based CAPTCHA design requires that a CAPTCHA be resistant against segmentation. In this paper, we examine CAPTCHA usability issues and current segmentation techniques that have been used to attack various CAPTCHA schemes. We then introduce the design of a new CAPTCHA scheme that was designed based on these usability and segmentation considerations. Our goal was to also design a text-based CAPTCHA scheme that can easily be used on increasingly pervasive touch-screen devices, without the need for keyboard input. This paper also examines the usability and robustness of the proposed CAPTCHA scheme.

Keywords

scheme, identi, cation, character, captcha, locations

Disciplines

Engineering | Science and Technology Studies

Publication Details

Nguyen, V. Duc., Chow, Y. & Susilo, W. (2014). A CAPTCHA scheme based on the identification of character locations. Lecture Notes in Computer Science, 8434 60-74.

A CAPTCHA Scheme based on the Identification of Character Locations

Vu Duc Nguyen, Yang-Wai Chow and Willy Susilo*

Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Australia
{vdn108, caseyc, wsusilo}@uow.edu.au

Abstract. CAPTCHAs are a standard security mechanism used on many websites to protect online services against abuse by automated programs, or bots. The purpose of a CAPTCHA is to distinguish whether an online transaction is being carried out by a human or a bot. Unfortunately, to date many existing CAPTCHA schemes have been found to be vulnerable to automated attacks. It is widely accepted that state-of-the-art in text-based CAPTCHA design requires that a CAPTCHA be resistant against segmentation. In this paper, we examine CAPTCHA usability issues and current segmentation techniques that have been used to attack various CAPTCHA schemes. We then introduce the design of a new CAPTCHA scheme that was designed based on these usability and segmentation considerations. Our goal was to also design a text-based CAPTCHA scheme that can easily be used on increasingly pervasive touch-screen devices, without the need for keyboard input. This paper also examines the usability and robustness of the proposed CAPTCHA scheme.

Keywords: text-based CAPTCHA, segmentation resistance, optical character recognition

1 Introduction

CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) are essentially automated reverse Turing tests that are commonly used by online services to distinguish whether an online transaction is being carried out by a human or an automated program, i.e. a bot [24]. Since its inception, many diverse CAPTCHA schemes have been proposed, and to date, CAPTCHAs have become a standard Internet security mechanism for deterring automated attacks by bots and other malicious programs. Of the different types of CAPTCHAs (e.g. text-based, image-based, audio-based) that are currently used in practice, text-based CAPTCHAs are the most prevalent form in use. Chellapilla et al. [12] attribute the popularity and pervasiveness of text-based

* This work is supported by ARC Future Fellowship FT0991397.

CAPTCHAs to its human friendliness, intuitiveness, ease of use, low implementation cost, etc. In general, a traditional text-based CAPTCHA challenge consists of a word or a random sequence of characters, which may consist of letters and/or digits, that are embedded within an image. The user's task is to solve the CAPTCHA challenge by entering the appropriate sequence of characters in the correct order.

Unfortunately, while there are numerous existing CAPTCHA schemes that are currently deployed on a vast number of websites, many of these schemes have been found to be insecure. The vulnerability of these schemes stem from various design flaws that can be exploited to break these CAPTCHAs. Over the years, researchers have documented many techniques that can be used to break a variety of CAPTCHA schemes at high success rates [1, 3, 4, 7, 13, 16, 19, 21, 20, 28, 29]. Furthermore, attacks against CAPTCHA schemes are not only limited to traditional text-based CAPTCHAs, as techniques to break other forms of CAPTCHAs have also been documented. These include techniques for breaking animated CAPTCHAs [23, 27], 3D-based CAPTCHAs [22], image-based CAPTCHAs [31], audio-based CAPTCHAs [5], etc. As such, the design of a CAPTCHA scheme that is robust against automated attacks is an important and open research problem. In addition, the challenge of designing a secure CAPTCHA scheme is further complicated by the fact that not only must the resulting CAPTCHA be secure against automated attacks, it must also be easily usable by a human.

This paper presents the design of a new CAPTCHA scheme along with a discussion on the security and usability of the proposed scheme. It is widely accepted that state-of-the-art in CAPTCHA design requires that a CAPTCHA be segmentation-resistant [1, 12], as once a CAPTCHA can be segmented into its constituting characters, the scheme is essentially deemed to be broken [11]. In this paper, we first examine CAPTCHA usability issues and current segmentation techniques that have been used to attack a variety of existing CAPTCHAs, in order to identify the various factors that must be considered when designing a robust CAPTCHA scheme. This will then be followed by a discussion on the design of our proposed scheme in relation to these usability and segmentation considerations. In addition, this paper presents the results of a user study that was conducted to ascertain the usability of the proposed CAPTCHA scheme, followed by an analysis on the robustness of the scheme.

Our Contributions. In this paper, we present and discuss the design of a new text-based CAPTCHA scheme that is robust against current segmentation techniques. The proposed CAPTCHA scheme is also usable on touch-screen interfaces, without the need to enter text via a physical or on-screen keyboard. Our proposed approach alters the traditional challenge posed by conventional text-based CAPTCHAs in which the user's task is to answer the question of "What is the text?", into a question of "Where is the text?". Hence, the user's task is to recognize and identify the locations of characters in the CAPTCHA challenge. Furthermore, this paper outlines and examines the various usability

and security issues that must be considered in the design of a robust CAPTCHA scheme.

2 Background

2.1 Usability versus Security

The fundamental requirement of a practical CAPTCHA scheme necessitates that humans must be able to solve the CAPTCHA challenges with a high degree of success, while the likelihood that a computer program can correctly solve them must be very small. This tradeoff between the usability and security of a CAPTCHA scheme is a hard act to balance. Security considerations push designers to increase the difficulty of the CAPTCHA scheme, while usability requirements compel them to make the scheme only as difficult as they need to be, but still be effective in deterring automated abuse. These conflicting demands have resulted in the ongoing arms race between CAPTCHA designers and those who try to break them [10, 15].

The design of a robust CAPTCHA must capitalize on the difference in natural human ability and the capabilities of current computer programs [10]. This is a challenging task because on one hand, computing technology and algorithms that can be used to solve CAPTCHAs are constantly evolving and improving (e.g. Optical Character Recognition (OCR) software), while on the other hand, humans must rely on their inherent abilities and are unlikely to get better at solving CAPTCHAs. In addition, it has been shown that several key features that are commonly employed to increase the usability of CAPTCHA schemes can easily be exploited by computer programs.

The use of color is a major factor that has to be considered in CAPTCHA design. Color is used in CAPTCHAs for a variety of reasons. From a usability perspective, color is a strong attention-getting mechanism, it is appealing and can make CAPTCHA challenges interesting, appropriate use of color can facilitate recognition and comprehension of a CAPTCHA, and so on [2]. However, it has been shown that the imprudent use of color can have a negative impact on both CAPTCHA usability and security [1, 30].

To aid usability, text-based CAPTCHA challenges that are based on dictionary words are intuitive and easier for humans to solve because humans find familiar text easier to perceive and read [25]. However, CAPTCHA challenges that are based on language models are susceptible to dictionary attacks. Rather of trying to recognize individual characters, which may be difficult if the characters are overly distorted and/or overlapping, researchers have successfully used holistic approaches to recognize entire words for CAPTCHA schemes that are based on language models [4, 21].

Instead of using actual dictionary words, it is possible to take advantage of text familiarity using "language-like" strings. Phonetic text or Markov dictionary strings are pronounceable strings that are not words of any language. Experiments have shown that humans perform better when solving CAPTCHAs with

pronounceable strings in contrast to CAPTCHAs which contain purely random characters [25]. Nevertheless, the disadvantage of using this approach is that certain characters (i.e. vowels) will appear at higher frequencies in pronounceable strings compared to other characters. The higher frequencies of certain characters makes the resulting CAPTCHA more vulnerable to attacks.

In addition, for usability purposes text-based CAPTCHAs should avoid the use of confusing digits and letters like the digit '0' and the letter 'O', the digit '1' and the letter 'l', etc. Confusing character combinations like 'W' and 'VV', 'm' and 'rn', etc. should also be avoided. Furthermore, if letter case is important, then confusing characters include upper and lower case pairs like 'S' and 's', 'Z' and 'z', etc. [30].

2.2 Segmentation Resistance

Chellapilla et al. [11, 13] demonstrated that machine learning algorithms can successfully be used to break a variety of different CAPTCHA schemes. In doing so, they also showed that computers can outperform humans at the task of recognizing individual characters. The task of solving a text-based CAPTCHA consists of two main challenges; namely, a segmentation challenge, followed by a recognition challenge. The segmentation challenge refers to the identification and separation of a sequence of characters into its constituting characters in the correct order, and the recognition challenge involves recognizing the individual characters. As such, it follows that once a computer program can adequately reduce a CAPTCHA to the problem of recognizing individual characters, the CAPTCHA is essentially broken. Hence, it is widely accepted that a secure CAPTCHA scheme must be designed to be segmentation-resistant [1, 10].

Broad classifications of three of the mainstream segmentation-resistant methods that are currently employed by a number of CAPTCHA schemes to deter segmentation, as defined by Bursztein et al. [7], are described as follows:

- **Background confusion**: CAPTCHA schemes that use this approach to prevent segmentation attempt to blend the CAPTCHA text with the background. There are three main ways of achieving this; namely, by using a complex background image, by using a background with very similar colors to the text, or by adding noise. Some CAPTCHA schemes employ a combination of these techniques.
- **Using lines**: In this approach, random line(s) that cross over multiple characters are drawn over the CAPTCHA text. This is done to help prevent segmentation because characters in the CAPTCHA challenge are connected together by the lines.
- **Collapsing**: This approach typically involves removing the space between characters, tilting characters and/or overlapping them, which in effect crowds the characters together. The notion behind this approach is to make segmentation difficult because the characters are either very close or joined together. While this is considered to be the most secure anti-segmentation mechanism, often design flaws in the CAPTCHA scheme allow attackers to exploit these

flaws in order to perform segmentation [7]. Some of these attacks are described in the next section.

2.3 CAPTCHA Segmentation Techniques

While it is widely accepted that a robust CAPTCHA scheme must be designed to be segmentation-resistant, many existing schemes that adopt anti-segmentation mechanisms have in fact been found to be insecure. This is mainly due to certain design flaws in the scheme that can be exploited by the attacker to segment the CAPTCHA. Over the years, researchers have documented a variety of different techniques that can be used to segment various CAPTCHA schemes. Among others, several key segmentation techniques are described as follows:

- **De-noising algorithms**: De-noising techniques are mainly used to remove random noise from a CAPTCHA. Of the various de-noising techniques that have been proposed over the years, the Markov Random Field technique, a.k.a. Gibbs algorithm [18], has been found to be very effective [7]. The algorithm works by computing the energy of each pixel based on its surroundings and removing pixels that have an energy below a certain threshold. This is performed iteratively until there is no more pixels to remove.
- **Histogram-based segmentation**: Histogram-based segmentation is a popular CAPTCHA segmentation technique that projects a CAPTCHA's pixels to their respective X or Y coordinates [3, 7, 19, 28, 29]. By producing a histogram of the number of pixels in the X or Y dimension, in general, sections that contain a large pixel count contain characters, while sections with a low pixel count are potential positions that can be used to segment the characters. For CAPTCHAs where the characters are only joined slightly or connected using small lines, this method is effective in segmenting the CAPTCHA. In other CAPTCHA attacking methods, this technique is efficient in separating groups of characters or potential groups prior to the use of other segmentation techniques.
- **Color Filling Segmentation (CFS)**: The basic idea behind this technique is identify a foreground color pixel (i.e. a pixel with a color associated with the text) and to trace all the neighboring pixels with the same color which are connected to this pixel, in effect performing a flood fill algorithm, to identify a chunk of connected pixels. This process is repeated until all chunks in a CAPTCHA have been identified [3, 29]. The end result of using this method is that an attacker can identify individual characters or groups of characters. This method is often used in conjunction with other segmentation techniques.
- **Opportunistic segmentation**: This technique relies on making educated guesses based on prior knowledge about the CAPTCHA scheme. The technique exploits regular and predictable features of a CAPTCHA scheme in order to approximate where the segmentation cuts should be. For example, CAPTCHA schemes that use a fixed number of characters per challenge,

where characters are usually placed at certain fixed locations, and all characters have roughly the same width, are susceptible to opportunistic segmentation. The reason for this is because it is easy to make an educated guess as to where the segmentation cuts are likely to occur [7, 10].

– **Segmentation based on patterns and shapes**: In this segmentation approach, attackers try to identify certain patterns and shapes that typically characterize some characters. For example, characters like 'a', 'b', 'd', 'e', 'g', 'o', 'p', 'q' all contain loops or circular regions, characters like 'i', 'j', 'l' typically consist of small vertical blocks of pixels, etc. [3, 16]. Once these patterns are determined, these particular features can be identified in the CAPTCHA which in turn allows the attacker to ascertain appropriate locations to segment the text.

The techniques described here are some generic methods that have been adopted to attack a number of different CAPTCHA schemes. There are also other specialized segmentation methods that have been used to attack specific CAPTCHA schemes [1, 28]. While to date there is no comprehensive segmentation solution that can be used to break all CAPTCHA schemes, many CAPTCHA segmentation attacks use a combination and/or variations of the techniques described above.

3 Design of the Proposed CAPTCHA Scheme

Many of the existing CAPTCHA schemes that adopt anti-segmentation mechanisms have actually been broken through the use of using various segmentation techniques, including those described in the previous section. As such, in designing a CAPTCHA scheme, it is imperative to examine the use of anti-segmentation mechanisms and to consider the resulting CAPTCHA's robustness against current segmentation techniques. Since a CAPTCHA scheme's robustness is determined by the cumulative effects of its design choices [10], we will discuss the reasons and security issues that were considered in the design of our proposed CAPTCHA scheme.

In general, the use of background confusion techniques as a security mechanism has been deemed to be insecure. For one thing, human usability considerations require that the text stand out from the rest of the background, otherwise a human will not be able to adequately solve the CAPTCHA, and for this reason it is likely that any background can be processed, filtered and removed. It has therefore been recommended that backgrounds only be used for cosmetic purposes [7]. In addition, using lines to connect characters as a segmentation-resistant technique has also been found to be inadequate in preventing the resulting CAPTCHA from being segmented. This is because there are a variety of techniques that can efficiently detect and/or remove lines, for example, through the use of line detection algorithms such as the Hough transform [17], erosion and dilation techniques [26], as well as histogram-based segmentation techniques.

Collapsing techniques such as crowding and overlapping characters together are considered to be the most secure anti-segmentation approach to date. How-

ever, an increasing number of CAPTCHA scheme that have been designed with these techniques have been successfully broken because attackers have managed to exploit design flaws in the various schemes [3, 7]. One of the reasons for this is due to the fact that current text-based CAPTCHAs crowd and overlap characters in the horizontal dimension only. This has allowed attackers to identify predictable features in the CAPTCHA schemes and to approximate where the segmentation cuts should occur.

In this paper, we propose the design of a text-based CAPTCHA scheme that is robust against current segmentation techniques. One of our goals was to also design a scheme that can easily be used on the increasingly popular and pervasive touch-screen devices, without having to rely on the need to input text using a physical or on-screen keyboard. As such, unlike conventional text-based CAPTCHAs which deal with the question of "What is the text?", our approach alters this into the question of "Where is the text?". Examples of our proposed CAPTCHA scheme are depicted in Figure 1.

(a) Colored version (b) Black and white version

Fig. 1. Examples of the proposed CAPTCHA scheme.

There are two slightly different versions of our proposed scheme. Figure 1(a) shows an example of a colored version of the proposed CAPTCHA, while Figure 1(b) shows a black and white version. Note that the use of color in our scheme is not for any security reasons, but rather it is primarily used for reasons of usability. This is in line with recommendations that color be used in CAPTCHAs for usability rather than for security [2]. The colored version of our scheme was implemented to ascertain whether or not it would facilitate human visual perception, by making it easier for humans to distinguish characters from the background, instead of having to solely rely on the outlines of characters. To help answer this question, a user study was conducted and the findings of the experiment are discussed in Section 4.1.

For each CAPTCHA in our proposed scheme, a user is provided with a challenge character set (the list of characters at the bottom of the CAPTCHA) and an image that contains these specific characters, along with a whole lot of other non-relevant characters. To solve the CAPTCHA, the user's task is to find the locations of the characters provided in the challenge character set in the image, then drag-and-drop each challenge character onto the correct character in the image. This can easily be done using a mouse, a pointing device or on a touch-screen. Figure 2(a) and Figure 2(b) show examples of an incorrect answer and a correct answer respectively[1]. Note that the correct characters in the CAPTCHA are only highlighted after the solution has been submitted. Drag-and-drop CAPTCHAs are not new and have previously been proposed, for example, for identifying images of 3D text objects [9]. Others have proposed clickable CAPTCHAs for mobile devices [14].

(a) An incorrect solution (b) A correct solution

Fig. 2. Example answers.

The design of the proposed CAPTCHA scheme mainly relies on the collapsing technique for preventing segmentation. However, unlike conventional text-based CAPTCHAs, it can be seen from the examples shown in Figures 1 and 2 that our approach not only crowds, tilts and overlaps characters in the horizontal dimension, it also does this in the vertical dimension. In addition, our approach uses many more characters than is actually contained in the challenge character set. For conventional CAPTCHAs that adopt the crowding characters together approach, each character only has a maximum of two neighboring left and right characters that it can overlap with. Our approach allows for center characters to overlap with a maximum of eight neighboring characters. Obviously, characters at the sides have less neighboring characters. As such, it can easily be seen that

[1] Animated examples depicting user interaction with the proposed scheme can be found at: http://www.uow.edu.au/~wsusilo/CAPTCHA/newCAPTCHA.html

our approach effectively prevents segmentation techniques like histogram-based segmentation, color filling segmentation or methods that try to identify character patterns and shapes to determine where the text should be segmented, because the text cannot simply be segmented using single lines.

The challenge character set in the proposed scheme consists of random characters instead of dictionary words, and is therefore not affected by dictionary attacks. Rather than using a fixed number of characters per challenge, the number of characters in the challenge set can be randomized. Furthermore, for all characters in the image, their rotation angles, positions and sizes, as well as the number of characters, can all be randomized within a certain range of values. This prevents other segmentation techniques like opportunistic segmentation, because the randomization makes it difficult to predict where individual characters are located. Also, since color in our scheme is not used for security, attackers cannot use color filtering to identify the locations of individual characters in the image.

From a usability perspective, the proposed CAPTCHA scheme is based on Gestalt principles of visual perception. By removing the outlines of characters wherever they overlap in the image to deter segmentation, a human can still solve the CAPTCHA because humans perceive objects as a whole and the visual system fills in the missing areas. Our implementation was programmed to avoid the use of confusing character combinations within the same CAPTCHA challenge. In our scheme, each unique character only occurs once per challenge. Also, although the characters are crowded together, unlike many conventional text-based CAPTCHAs which rely on character warping or distortion to deter automated attacks, the characters in our approach are not distorted. This is because the security mechanism of our approach does not rely on character warping or distortion, which in turn makes the task of recognizing undistorted characters easier for humans. Only upper case letters and digits are used in the current implementation.

4 Results and Discussion

4.1 User Study

User studies with human participants are the best method of establishing the human-friendliness of a CAPTCHA scheme [10]. As such, a pilot user study was conducted to determine the usability of the proposed CAPTCHA scheme. The study was also done to ascertain whether the use of color in one of the CAPTCHA versions would make a significant difference for a human. A total of 42 volunteers, 33 male and 9 female, took part in the experiment. Participants were aged between 18 and 58 (average $\sim$32.9, standard deviation $\sim$1.05). None of the participants had ever seen or had any prior knowledge about the proposed CAPTCHA scheme.

Method. For the study, a total of 30 CAPTCHA challenges were generated. Of this, 15 were the colored version and the other 15 were the black and white

version. For each version, the challenge character set consisted of 3 characters for 7 of the CAPTCHAs, while the challenge character set for the remaining 8 CAPTCHAs consisted of 4 characters. The order in which the CAPTCHAs were presented to the participants was randomized. In order to compare results, the same experimental conditions were maintained for all participants. Hence, each participant was required to solve the same set of 30 CAPTCHAs.

Before the experiment, each of the participants was given instructions about the experimental task and what they were required to do. Their task was simply to view each challenge and solve the CAPTCHA using a mouse. The duration of the experiment was designed to be short to avoid participants loosing concentration. The total time required by each participant to complete the experiment varied between individuals, but took no longer than 15 minutes. During the experiment, we recorded the time taken by each user to complete each CAPTCHA challenge as well as all their answers. Participants were not provided with any information regarding the correctness of their answers. At the end of the experiment, participants were also given a post-experiment questionnaire that contained questions about their subjective opinions in relation to the usability of the proposed CAPTCHA scheme.

Results. Table 1 shows the results of the experiment. It shows the difference in accuracy and average completion time between the colored version of the CAPTCHA and the black and white version. For good usability and to avoid users getting annoyed, Chellapilla et al. [10] state that the human success rate of a good CAPTCHA should approach 90%. It can be seen from the user study results that the success rate of both versions of our proposed CAPTCHA satisfies this benchmark.

Table 1. Average completion time and the success rates

	Accuracy (%)	**Average Time (s)**
Colored version	96.35	18.97
Black and white version	93.81	26.24

In addition, the experimental results suggest that the use of color has an effect on the overall usability of the CAPTCHA scheme. In Table 1, one can see that the accuracy for the colored version was higher than the black and white version. However, a Chi-square test did not reveal a significant difference. The results also show a difference in the average completion times. Upon further analysis, a t-test showed a significant difference in the average completion time for the black and white version ($M = 26.24$s, $SD = 18.26$s) and the colored version ($M = 18.97$s, $SD = 7.60$s); $t(841) = 9.22$, $p < 0.001$. This suggests that perceptually the use of color to distinguish characters from the background makes the CAPTCHA easier for a human to solve. Table 2 shows the breakdown

of average completion times based on the number of characters per challenge. Not surprisingly, the more challenge characters in a CAPTCHA, the longer it took participants to solve the CAPTCHA.

Table 2. Average completion time based on the number of challenge characters

	Average Time (s)	
	3 characters	**4 characters**
Colored version	17.70	21.52
Black and white version	20.67	31.12

It should be noted that overall the time taken by participants to complete our proposed CAPTCHA scheme appears to be longer than that required to solve other image CAPTCHAs. In a large scale evaluation study by Bursztein et al. [6] where they tested a variety of CAPTCHA schemes, they reported an average solving time of 9.8 seconds for image CAPTCHAs and 28.4 seconds for audio CAPTCHAs. The lengthy duration required to solve our proposed CAPTCHA, especially the black and white version, is probably due to two factors. First, users have to search the image in order to identify the locations of the appropriate characters. A task that appears to be more difficult in the case of the black and white version. Second, users may spend more time when trying to accurately drag-and-drop characters to the appropriate locations in the image, compared to traditional text-based CAPTCHAs where they would simply input text via a keyboard. Figure 3 is an example of a plot which shows the locations of where the participants' "dropped" each of the individual challenge characters when attempting to solve the CAPTCHA.

In one of the questions on the post-experiment questionnaire, participants were asked to rate the ease of use of the proposed CAPTCHA scheme using a 7-point Likert scale, with 1 being very difficult to use and 7 being very easy to use. The average participant response to this question was 5.14. When asked to rate the usability of the proposed CAPTCHA scheme as compared to other existing CAPTCHAs that they had used in the past, the average response was 4.65, where 1 was much harder to use and 7 was much easier to use. This indicates that in general, the majority of participants had a positive opinion about the usability of the proposed scheme.

4.2 Security

Figure 4 shows example images resulting from a number of typical techniques that are often used to attack CAPTCHA schemes. The CAPTCHA challenge itself is shown in Figure 4(a). Figure 4(b) shows the CAPTCHA image's skeleton. Skeletonization is a process that is used to thin a shape while preserving the general pattern of the shape. Skeleton images have been used to attack CAPTCHAs

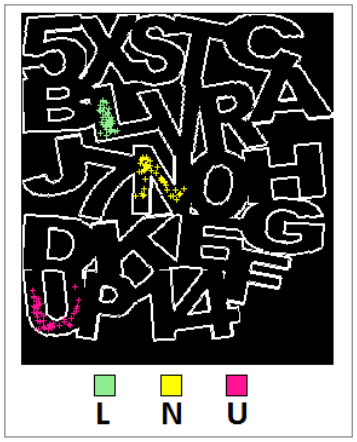


Fig. 3. Example showing the locations of where participants' "dropped" the respective challenge characters in the CAPTCHA.

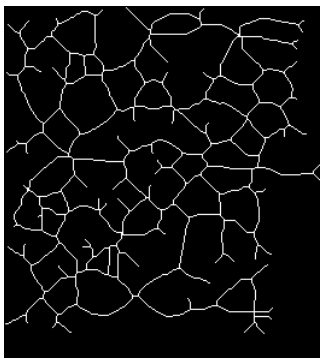
as the skeleton thins the characters to a single pixel thickness, which may potentially be used to identify geometric features of the characters [3]. It can be seen in the figure that because characters in the proposed CAPTCHA are overlapped in both the vertical and horizontal dimensions, the skeleton image does not result in useful information that can be used to segment or to identify individual characters. Figure 4(c) shows the results of processing the CAPTCHA using a Canny edge detection filter [8]. In the proposed CAPTCHA scheme, the edge detection filter merely highlights the outlines of the overlapping characters, but does not facilitate the task of separating the characters.

Histogram-based segmentation is a commonly used approach that projects CAPTCHA pixels in the X and Y dimensions in order to identify potential locations to segment the text. Figure 4(d) shows the results of histogram projections that project the pixels that form the outlines of the overlapping characters in the X and Y dimensions respectively. X and Y histogram projections of the internal regions of the characters are shown in Figure 4(e). It can be seen that both histogram projection approaches do not provide enough information that can be used to adequately segment the CAPTCHA challenge.

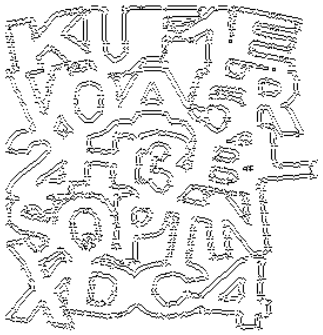
5 Conclusion

In this paper, we presented the design of a new CAPTCHA scheme that was developed to be segmentation-resistant. To achieve this, this paper first examined the various usability and security issues that have to be considered when designing a robust CAPTCHA scheme, then described how the proposed CAPTCHA scheme satisfied these issues. The CAPTCHA scheme introduced in this paper is based on the concept of identifying character locations, rather than merely recognizing characters, and can easily be used on touch-screen devices without the

(a) CAPTCHA challenge

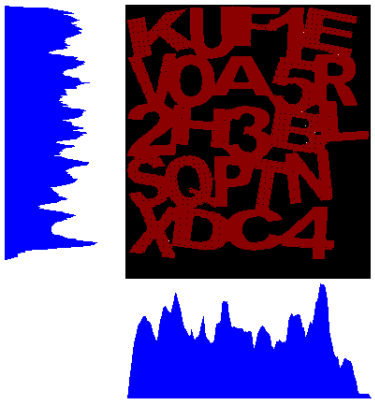


(b) Skeleton image



(c) Edge detection

(d) X and Y histogram projections of text outlines



(e) X and Y histogram projections of text regions

Fig. 4. Image processing results on a CAPTCHA challenge.

need for a keyboard. In addition, this paper also presented the results obtained from a user study that was conducted to ascertain the usability of the proposed CAPTCHA scheme.

References

1. A. S. E. Ahmad, J. Yan, and L. Marshall. The robustness of a new CAPTCHA. In *EUROSEC*, pages 36–41, 2010.
2. A. S. E. Ahmad, J. Yan, and W.-Y. Ng. CAPTCHA design: Color, usability, and security. *IEEE Internet Computing*, 16(2):44–51, 2012.
3. A. S. E. Ahmad, J. Yan, and M. Tayara. The robustness of Google CAPTCHAs. *University of Newcastle, UK, Technical Report*, 1278:1–15, 2011.
4. P. Baecher, N. Buscher, M. Fischlin, and B. Milde. Breaking reCAPTCHA: A holistic approach via shape recognition. In J. Camenisch, S. Fischer-Hubner, Y. Murayama, A. Portmann, and C. Rieder, editors, *Future Challenges in Security and Privacy for Academia and Industry*, volume 354 of *IFIP Advances in Information and Communication Technology*, pages 56–67, 2011.
5. E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. C. Mitchell. The failure of noise-based non-continuous audio CAPTCHAs. In *IEEE Symposium on Security and Privacy*, pages 19–31. IEEE Computer Society, 2011.
6. E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky. How good are humans at solving CAPTCHAs? a large scale evaluation. In *IEEE Symposium on Security and Privacy*, pages 399–413. IEEE Computer Society, 2010.
7. E. Bursztein, M. Martin, and J. C. Mitchell. Text-based CAPTCHA strengths and weaknesses. In Y. Chen, G. Danezis, and V. Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 125–138. ACM, 2011.
8. J. Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679 –698, 1986.
9. S. K. Chaudhari, A. R. Deshpande, S. B. Bendale, and R. V. Kotian. 3D drag-n-drop CAPTCHA enhanced security through CAPTCHA. In B. K. Mishra, editor, *ICWET*, pages 598–601. ACM, 2011.
10. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. Building segmentation based human-friendly Human Interaction Proofs (HIPs). In H. S. Baird and D. P. Lopresti, editors, *HIP*, volume 3517 of *Lecture Notes in Computer Science*, pages 1–26. Springer, 2005.
11. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. Computers beat humans at single character recognition in reading based Human Interaction Proofs (HIPs). In *CEAS*, 2005.
12. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. Designing human friendly Human Interaction Proofs (HIPs). In G. C. van der Veer and C. Gale, editors, *CHI*, pages 711–720. ACM, 2005.
13. K. Chellapilla and P. Y. Simard. Using machine learning to break visual Human Interaction Proofs (HIPs). In *NIPS*, 2004.
14. R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang. Making CAPTCHAs clickable. In M. Spasojevic and M. D. Corner, editors, *HotMobile*, pages 91–94. ACM, 2008.
15. Y.-W. Chow and W. Susilo. AniCAP: An animated 3D CAPTCHA scheme based on motion parallax. In D. Lin, G. Tsudik, and X. Wang, editors, *CANS*, volume 7092 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2011.

16. C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. A. Aquino, and L. Reyes-Cabrera. Breaking reCAPTCHAs with unpredictable collapse: Heuristic character segmentation and recognition. In J. A. Carrasco-Ochoa, J. F. M. Trinidad, J. A. Olvera-López, and K. L. Boyer, editors, *MCPR*, volume 7329 of *Lecture Notes in Computer Science*, pages 155–165. Springer, 2012.
17. R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Jan. 1972.
18. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.
19. S.-Y. Huang, Y.-K. Lee, G. Bell, and Z.-h. Ou. An efficient segmentation algorithm for CAPTCHAs with line cluttering and character warping. *Multimedia Tools and Applications*, 48(2):267–289, 2010.
20. P. Liu, J. Shi, L. Wang, and L. Guo. An efficient ellipse-shaped blobs detection algorithm for breaking Facebook CAPTCHA. In Y. Yuan, X. Wu, and Y. Lu, editors, *Trustworthy Computing and Services*, volume 320 of *Communications in Computer and Information Science*, pages 420–428. Springer, 2013.
21. G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *CVPR (1)*, pages 134–144, 2003.
22. V. D. Nguyen, Y.-W. Chow, and W. Susilo. Breaking a 3D-based CAPTCHA scheme. In H. Kim, editor, *ICISC*, volume 7259 of *Lecture Notes in Computer Science*, pages 391–405. Springer, 2011.
23. V. D. Nguyen, Y.-W. Chow, and W. Susilo. Breaking an animated CAPTCHA scheme. In F. Bao, P. Samarati, and J. Zhou, editors, *ACNS*, volume 7341 of *Lecture Notes in Computer Science*, pages 12–29. Springer, 2012.
24. L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In E. Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2003.
25. S.-Y. Wang, H. S. Baird, and J. L. Bentley. CAPTCHA challenge tradeoffs: Familiarity of strings versus degradation of images. In *ICPR (3)*, pages 164–167. IEEE Computer Society, 2006.
26. J. Wilkins. Strong CAPTCHA guidelines v1.2, 2009. http://www.bitland.net/captcha.pdf.
27. Y. Xu, G. Reynaga, S. Chiasson, J.-M. Frahm, F. Monrose, and P. Van Oorschot. Security and usability challenges of moving-object CAPTCHAs: decoding codewords in motion. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security'12, pages 4–4, Berkeley, CA, USA, 2012. USENIX Association.
28. J. Yan and A. S. E. Ahmad. Breaking visual CAPTCHAs with naive pattern recognition algorithms. In *ACSAC*, pages 279–291. IEEE Computer Society, 2007.
29. J. Yan and A. S. E. Ahmad. A low-cost attack on a Microsoft CAPTCHA. In P. Ning, P. F. Syverson, and S. Jha, editors, *ACM Conference on Computer and Communications Security*, pages 543–554. ACM, 2008.
30. J. Yan and A. S. E. Ahmad. Usability of CAPTCHAs or usability issues in CAPTCHA design. In L. F. Cranor, editor, *SOUPS*, ACM International Conference Proceeding Series, pages 44–52. ACM, 2008.
31. B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition CAPTCHAs. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 187–200. ACM, 2010.