ABSTRACT

CAPTCHA is an effective and widely used solution for preventing computer programs (i.e., bots) from performing automated but often malicious actions, such as registering thousands of free email accounts or posting advertisement on web blogs. To make CAPTCHAs robust to automatic character recognition techniques, the text in the tests are often distorted, blurred, and obscure. At the same time, those robust tests may prevent genuine users from telling the text easily and thus distribute the cost of crime prevention among all the users. Thus, we are facing a dilemma, that is, a CAPTCHA should be robust enough so that it cannot be broken by programs, but also needs to be easy enough so that users need not to repeatedly take tests because of wrong guesses.

In this paper, we attempt to resolve the dilemma by proposing a human computation game for quantifying the usability of CAPTCHAs. In our game, DevilTyper, players try to defeat the devils as many as possible by solving CAPTCHAs and player behavior in completing a CAPTCHA are recorded at the same time. Therefore, we can evaluate CAPTCHAs’ usability by analyzing collected player inputs. Since DevilTyper provides entertainment purposes itself, we conduct a large-scale study for CAPTCHAs’ usability without the resource overhead required by traditional survey-based studies. In addition, we propose a consistent and reliable metric for assessing usability. Our evaluation results show that DevilTyper provides a fun and efficient platform for CAPTCHA designers to assess their CAPTCHA usability and thus improve the CAPTCHA design.

Categories and Subject Descriptors
H.1.2 [Models and Principles]: User/Machine Systems—Human factors; K.8.0 [Personal Computing]: General—Games; H.5.2 [Information Interfaces And Presentation]: User Interfaces—User-centered design

General Terms
Design, Human Factors, Security

Keywords
Games with a purpose, GWAP, Human computation, Usability, Optical character recognition, Human perception

1. INTRODUCTION

Preventing computer programs from performing automated malicious tasks, such as registering thousands of free email accounts, has been one of the most challenging tasks of system administrators. For this issue, CAPTCHA (Completely Automated Public Turing test to tell Computer and Humans Apart) [12] is known as an effective and widely used solution. Although there are many types of CAPTCHAs [5,8], text-based CAPTCHA scheme, which asks users to recognize the distorted text, is the most widely used and is adopted by most commercial services, such as Google, Yahoo!, and Microsoft. For this reason, we focus on text-based CAPTCHA schemes in this work. Text-based CAPTCHAs are machine-generated images which contain obscure text. The common procedures to generate such images often include distortions, overlapping, clipping, and noise addition. These procedures are performed to make image recognition algorithms unable to resolve the text in the images. However, the distortion of the text should be controlled to a reasonable level so that human can still tell the text clearly.

As many CAPTCHAs have been broken by OCR (Optical Character Recognition) or other recognition techniques [1, 4, 9], it is necessary to enhance the complexity of CAPTCHAs to make them robust enough against such attacks. However, some enhancement procedures make the CAPTCHAs too difficult to be recognized by human, e.g., with too noisy background or too much text distortion. Therefore, we need to seek the balance in the tradeoff between the human usability and the computational recognition challenge when designing or employing a CAPTCHA test. Since there have been a great deal of works discussing the computational recognition difficulty of such tests in the field of pattern recognition, we focus on quantifying the usability of CAPTCHAs in this work.

The most intuitive way to assess the usability of CAPTCHAs is to ask numerous human subjects to solve assigned CAPTCHAs repeatedly. However, such surveys are cost-prohibitive if a large-scale study is required and the investigated CAPTCHAs are constantly updating. For example, investigating how different background noises affect the
user perception would require a large number of user inputs, which requires significant monetary investment to conduct user studies.

For these reasons, a human computation game [7] seems an efficient and effective approach to harness a huge amount of human resources for large-scale experiments. Several previous works, such as ESP game [13], Peekaboom [15], and KissKissBan [6], have successfully integrated the exploitation of human computation power and their own experiment purposes in games. From the perspective of players, they enjoy the games because they desire to be entertained; meanwhile, game designers obtain game logs and fulfill their research goals.

In this paper, we introduce DevilTyper, a web-based typing game for evaluating the human usability of CAPTCHAs. From the evaluation results, we demonstrate that our proposed approach is reliable for evaluating the usability of CAPTCHAs, and we also derive a rank list for CAPTCHAs selected in this work.

2. RELATED WORK

In recent years, research efforts on CAPTCHAs mainly focused on their robustness against automatic recognition techniques. Researches on computer vision [9] and pattern recognition [4] have shown the vulnerability of the CAPTCHA design. New methods for generating CAPTCHAs are also proposed to enhance the robustness of CAPTCHAs [5, 10].

At the same time, only a few works studied the usability issues of CAPTCHAs. In [17], the authors discussed a variety of factors that should be discussed when designing CAPTCHAs. [3] has examined the impact of common techniques used in CAPTCHA design, including distortion and background noise. Through conducting two user studies, which consist of 76 and 29 users respectively, they calculate users’ accuracy in solving CAPTCHAs to measure the tests’ usability under different distortion and noise settings. [2,16] discussed users’ familiarity of words and degradation of images in CAPTCHAs, and [3] evaluate the usability and computer vision techniques in single character recognition of CAPTCHAs.

The above works evaluated the usability of CAPTCHAs or the tradeoff between usability and robustness against computer recognition techniques based on small-scale user studies. In this work, we propose a more general platform to enable such studies in a cost-efficient way by hiding the mechanics of usability evaluation in computer games. The concept of using games to collect large-scale dataset is called Games With A Purpose (GWAP) [11]. In GWAP, users play games for fun and help game developers to achieve their tasks at the same time. The concept of GWAP has been applied to quite a few computationally hard problems, such as image annotation [13], locating object in images [15], and commonsense collection [14].

3. THE DEVILTYPER GAME

CAPTCHAs are designed to distinguish computer programs from human beings. While a CAPTCHA should be hard enough so that computer programs, such as bots, cannot break the test automatically, it should be easy enough for humans in order not to stop genuine users from using the services. In an extreme case, a human-unsolvable CAPTCHA would result in a totally unusable system.

In this work, we focus mainly on text-based CAPTCHAs, which require users to recognize distorted text in images, since they are widely used in most major websites, such as Google, Yahoo, and Microsoft. Text-based CAPTCHAs are favored for several reasons: 1) intuitive and few localization issues (users only need to type corresponding keystrokes), and 2) a large problem space and well-studied OCR techniques (i.e., providing strong security).

In text-based CAPTCHAs, usability is usually considered as readability. To assess readability, an intuitive way is to give users text-based CAPTCHAs and ask them to type the correct results as soon as possible. By collecting user inputs, we can calculate users’ input accuracy and response time in solving a CAPTCHA. However, this strategy would require significant monetary cost as large-scale user studies are normally required for sufficient amount of user inputs. For this reason, we adopt the concept of human computation, which takes “fun” as the incentive to engage users to complete specified tasks. We propose an interesting game, DevilTyper, to monitor and collect users’ behavior in solving a CAPTCHA.

3.1 Game Concepts

Currently, DevilTyper is designed as a single player game, in which players try to defeat devils appearing in the game by recognizing and typing the letters attached to each devil. When players enjoy the game, we collect each keystroke of the players along with the correctness of each keystroke for evaluating the readability of CAPTCHAs. In order to obtain as many participants as possible, the game should be interesting enough to attract people and encourage players continue their play. In the following sections, we describe the design and implementation details of DevilTyper.

![Image](389x303 to 484x370)

Figure 1: A CAPTCHA is attached to each devil. Players can defeat the devils by typing the distorted text in CAPTCHAs.

3.2 Game Description

In this game, the mission of a player is to destroy each devil as soon as possible. The only way to destroy a devil is to type the 3 to 5 character word attached to the devil, as shown in Figure 1. Therefore, players need to recognize the texts attached to the devils and finish the typing before devils come to hurt the player character.

3.2.1 Prior game

After entering a game, a player is asked to choose the skill level he would like to play. We provide three difficulty levels: Human, Hero, and Devil Buster, where the human level is the easiest level and the devil buster level is the most difficult one. From the easiest to the most tough level, the number of letters in each CAPTCHA increases from 3 to 5, and the maximum number of devils at any time
also increases. Furthermore, each level is comprised of 10 mini-missions, each of which introduces different numbers of devils that the player needs to destroy to finish the mission. As a player completes a mission, the number of devils he needs to destroy will increase in the next mission.

### 3.2.2 During game

As shown in Figure 2, devils move from the upside to the downside, and if a devil is not defeated by the player, it will keep moving down and get off of the screen, which indicates that the player get hurt by the devil. To target a devil, the player must type the first letter of its corresponding CAPTCHA. Once the player targets a devil, he is supposed to complete the remainder of the corresponding CAPTCHA to defeat it; otherwise, the player can release the target by pressing the SPACE key if he is unable to recognize the letters or want to destroy other devils first. In the game, the player character possesses two important attributes: the score and the HP (health points), where the HP is represented by a rectangle life bar and should be higher than zero anytime to keep playing. If a devil reaches the player character before being destroyed, the player is penalized by decreasing the HP value. On the other hand, the score accumulates if a devil is defeated. After defeating all the devils in a mission, the current mission is completed and the next mission will be started after a short time allowing the player to take a breath.

### 3.2.3 After game

To motivate the involvement of players, the name and score of players with high scores will be shown on the high score list, as shown in Figure 3. From the perspective of players, the ultimate goal of the game is to defeat as many devils as possible and therefore achieve a high ranking and score. Also, different skill levels introduces different levels of challenge so that users would continue the play with higher levels and feel accomplished. In [11], the authors mentioned that game features such as skill levels, score keeping, and high score list significantly increase the joyfulness of game play. As a result, we can achieve our study of CAPTCHAs’ usability while having players entertained through our game.

### 3.3 Implementation

To facilitate the game deployment, we implement DevilTyper using Adobe Flash. The game is now publicly available on the Internet\(^1\). Currently we provide six different CAPTCHA schemes and a plain-text CAPTCHA as a baseline for readability evaluation. The word associates with each devil is randomly selected from a dictionary and rendered using a random CAPTCHA scheme. We log all players’ actions, including each key pressed by the player, the time of keystrokes, the correctness of each keystroke, and the aiming and release of a devil, to provide researchers for statistical analysis.

### 3.4 Data Collection

In DevilTyper, we record all the actions performed by players and therefore can derive various performance metrics as follows:

- **Finish time**: the total time to solve a CAPTCHA.
- **Rate of typing error**: the number of CAPTCHAs with wrong keystrokes divided by the number of all CAPTCHAs.
- **Rate of timeout**: the number of timeout CAPTCHAs divided by the number of all CAPTCHAs. A CAPTCHA is regarded as “timeout” if it is targeted but the player did not have any further action in 3 seconds.
- **Rate of giving up**: the number of CAPTCHAs given up divided by the number of all CAPTCHAs. After targeting a devil (and it’s corresponding CAPTCHA), players can give up the devil by pressing SPACE key before timeout.
- **Rate of repeat typing**: the number of repeat keystrokes divided by the number of all keystrokes. In the game, players may type the same character repeatedly to avoid timeout.

1. \[http://deviltyper.iis.sinica.edu.tw/\]
4. DEVILTYPER DATA VALIDATION

In this section, we conduct experiments to validate whether the data collected by DevilTyper are consistent with those collected using traditional approaches.

4.1 Experiment Setup

Our experiments comprise two parts: contributions from DevilTyper game players, and validation data from Amazon Mechanical Turk.

We announced DevilTyper at a popular social network service PTT and held a four-week campaign. In each week, we awarded a certain amount of virtual currency to the top 5 players (at each difficulty level) attending the campaign, and also awarded 5 randomly chosen players who completed all the missions. Each of the players was awarded 1,000 to 10,000 PTT currency, which approximately corresponds to US$ 0.1 to US$ 1. The total cost for holding the four-week campaign is around US$ 30. During this period, the DevilTyper game has been played 6,500 times, and recorded players’ detailed behavior in solving 1,407,055 CAPTCHAs.

4.2 Used CAPTCHAs

We choose six CAPTCHAs in the experiments and use a plain-text CAPTCHA as a baseline. As shown in Figure 4, the CAPTCHAs we provide are AuthImage, Captcher, Kiranjv, SecurImage, CoolCAPTCHA, and TgCAPTCHA.

These CAPTCHA schemes use various strategies, such as background patterns, distortions, and noises, to generate images. In addition to usability evaluation of different CAPTCHA schemes, we also explore how the design factors, e.g., noise and distortion, affect those CAPTCHAs’ usability. To do so, we generate CAPTCHAs in CoolCAPTCHA and TgCAPTCHA with different levels of distortion and noise and record how users’ behavior changes. The analysis results will be presented in Section 5.

4.3 CAPTCHA Usability

In this subsection, we explore how different CAPTCHA schemes affect the usability by observing the different performance metrics players exhibited in solving the tests. The results are shown in Figure 5. It seems that the relative usability of the six CAPTCHAs remains consistent regardless of the performance metric used. Therefore, we can easily rank and cluster the CAPTCHAs by their usability. Among the CAPTCHAs, SecurImage and Captcher took players the longest time to solve the tests, as their rates of typing error, timeout, giving up, and repeat typing are generally the highest. Followed by these two CAPTCHAs, the usability of AuthImage and TgCAPTCHA are similar and moderate, which is followed by CoolCAPTCHA. Surprisingly, the usability of Kiranjv and plain text are similar.

4.4 Consistency between Performance Metrics

To further demonstrate the consistency among different performance metrics, we normalize each performance metric within the range 0 and 1. The normalized metrics are plotted in Figure 6. From the graph, we can see that different metrics are highly correlated with each other, which indicates that the usability of CAPTCHAs can be simply captured by either one metric. Therefore, we shall use “Rate of Typing Error” hereafter as the evaluation metric for CAPTCHA usability in the following discussion.

4.5 Comparison of Crowdsourcing and DevilTyper

To evaluate the trustworthiness of the data collected by DevilTyper, we also conduct a usability evaluation of the CAPTCHAs by crowdsourcing the experiments. We publish a human computation task, which requires users to solve 15 CAPTCHA tests, on a micro-task crowdsourcing platform, Amazon Mechanical Turk. On Mechanical Turk, any one can publish tasks and pay small amount of money, e.g., $ 0.01 to get his tasks done. Potential workers, who are anonymous Internet users, would try to complete the tasks with reasonable reward and difficulty. In our task, workers were asked to complete 15 random CAPTCHAs used in DevilTyper, where the workers’ accuracy and elapsed time

---

2PTT can be reached via telnet://ptt.cc, which is one of the largest social network services in Taiwan, with an average of 800,000 daily logins.

3http://mturk.com/
in solving each CAPTCHA were recorded. The task was accomplished 500 times by 44 workers, and 7,500 CAPTCHAs solving tasks were performed.

Figure 5: CAPTCHA comparisons. The CAPTCHAs are coded by A: AuthImage, B: Captcher, C: Kinravj, D: SecurImage, E: Plain text, F: CoolCAPTCHA, and G: TgCAPTCHA.

Figure 7: A comparison of CAPTCHA solving error rates on Mechanical Turk and DevilTyper.

We compute the typing error rate and average finish time for each CAPTCHA scheme and compare the results with those obtained from DevilTyper. The normalized error rates of CAPTCHA solving on Mechanical Turk and DevilTyper are shown in Figure 7. From the graph, we can see that the user performance are generally consistent no matter the users are playing the DevilTyper game or performing crowdsourcing tasks on Amazon Mechanical Turk. The results indicate that DevilTyper provides trustworthy user performance data for CAPTCHA usability evaluation and researchers can now use DevilTyper for such studies with a much lower or even none monetary cost.

5. DESIGN FACTOR ANALYSIS

In this section, we analyze the effect of various design factors on the usability of CAPTCHAs. We first investigate whether the choice of characters affects users’ performance in solving CAPTCHAs. We then analyze the usability of CAPTCHAs when two common text obscuration procedures, distortion and noise, are applied.

5.1 Effect of Characters in CAPTCHA

Each CAPTCHA scheme has its own obscuration algorithm to distort the text, which may have different impacts on the recognition difficulty of different characters. For example, as shown in Figure 8, “i” is hardly recognizable in TgCAPTCHA since players may be confused to identify whether it is a character or a random noise line. On the other hand, in CoolCAPTCHA, “i” is much easier to recognize. Understanding the effects of characters in different CAPTCHAs can help designers avoid the confusing characters and improve the CAPTCHA usability.

To evaluate the usability of characters with different CAPTCHA schemes, we first show how players react for characters in plain text in Figure 9. As the graph shows, players have different typing error rate even though the words are not distorted at all. This result may also be affected by the design of the keyboard layout. To eliminate such factors, we take users’ performance with the plain-text CAPTCHA as the baseline and examine the relative performance with each CAPTCHA scheme. Assuming the error rate for a character \( c \) with plain text is \( E_c \), and the error rate for character \( c \) with CAPTCHA type \( X \) is \( E'_c \), then the performance degradation rate is defined by \( \frac{E'_c - E_c}{E_c} \). The relationship between the performance degradation rate and CAPTCHA schemes...
on the choice of characters are presented in Figure 10.

From the graph, we can see that the effects of CAPTCHA scheme may be different for different characters. For example, the characters ‘Q’ and ‘V’ with the Captcher scheme and the characters ‘C’ and ‘T’ with the SecurImage scheme are particularly difficult to recognize. In addition, there are some characters always robust to the obscuration procedures (e.g., distortion and noise addition) of CAPTCHAs. For example, the characters ‘S’ and ‘X’ are more robust than other characters when being distorted, which looks reasonable because both characters have relatively unique typographical shapes. We believe such results provide helpful information when designing and applying CAPTCHAs. One obvious application is that, if a user happens to correctly solve all the characters beside a ‘C’ character with the SecurImage scheme, we may allow the user pass the test as the ‘C’ character is really difficult to recognize with that scheme.

5.2 Effect of Text Distortion

To understand the effect of text distortion on CAPTCHA usability, we choose CoolCAPTCHA, which is similar to the CAPTCHA scheme used by Google, to demonstrate how such analysis is done by using the traces produced by DevilTyper. As shown in Figure 11, CoolCAPTCHA provides three strategies for text distortion in addition to a plain raw text strategy:

Character Distance

Character distance stands for the distance between characters. In our experiment, we randomly set the character distances between 0.8 and 1.3, where a larger value corresponds to a tighter character arrangement. As shown in Figure 12(a), this parameter significantly influences the CAPTCHA usability especially when the distances between characters are small.

X-Axis Wave

X-axis wave controls the degree of sine-wave distortions of characters along the x-axis. In the experiment, this parameter is randomly set within the range from 0.5 to 1.2, where a larger magnitude corresponds to stronger distortion. According to Figure 12(b), the x-axis wave distortion does not make a systematic influence on users’ error rate, which implies that this type of distortion does not harm the CAPTCHA’s usability.

Y-Axis Wave

Similar to the x-axis wave parameter, the y-axis wave controls the degree of sine-wave distortions of characters along the y-axis, which we set within the range of 0.5 and 1.2 in our experiments. The results, as shown in Figure 12(c), indicates that the y-axis distortions lead to a much more significant impact on CAPTCHA usability than x-axis distortions. Therefore, CAPTCHA designers should be careful in choosing the appropriate degree for this type of distortions when adopting such CAPTCHAs in real use.

5.3 Effect of Noise Addition

To understand the effect of noise addition on CAPTCHA usability, we take TgCAPTCHA, which is similar to the previous Microsoft CAPTCHA scheme, to demonstrate how such analysis is done by using the traces produced by DevilTyper. As shown in Figure 13, TgCAPTCHA provides three strategies for noise addition:
Figure 12: Users’ average error rates with CAPTCHA rendered using different distortion strategies.

(a) Character Distance vs. Error rate

(b) X-Axis Wave vs. Error rate

(c) Y-Axis Wave vs. Error rate

Figure 13: TgCAPTCHA examples generated using different noise addition strategies.

(a) Long arcs noise

(b) Short arcs noise

(c) Short lines noise

6. CONCLUSION

In this paper, we have proposed a human computation game, DevilTyper, to facilitate the evaluation of CAPTCHA usability. DevilTyper is fun to play—we had over 6,500
game play rounds within four weeks, and collected users’ detailed behavior in solving more than 1,400,000 CAPTCHAAs. More importantly, such ample amount of data enable researchers quantify the usability of various CAPTCHAAs and their design factors in a rigorous statistical approach ever the interaction between factors is considered.

DevilTyper is now publicly available at http://deviltyper.iis.sinica.edu.tw. CAPTCHA designers are welcome to incorporate their CAPTCHAAs on the platform and obtain the users’ behavior traces for their own usability studies. We believe DevilTyper, as an open platform, can provide a testbed for CAPTCHA designers to assess their CAPTCHA usability and help develop more user-friendly CAPTCHAAs.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under the grants NSC99-2631-H-001-018 and NSC99-2631-H-001-020. It was also supported in part by the National Science Council of Taiwan under the grants NSC98-2221-E-001-017.

7. REFERENCES


Figure 9: Player typing time for characters in plain text.

Figure 10: Single character analysis.