# DevilTyper: A Game for CAPTCHA Usability Evaluation

CHIEN-JU HO, Academia Sinica
CHEN-CHI WU, National Taiwan University
KUAN-TA CHEN, Academia Sinica
CHIN-LAUNG LEI, National Taiwan University

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 article, 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 as many devils as possible by solving CAPTCHAs, and player behavior in completing a CAPTCHA is recorded at the same time. Therefore, we can evaluate CAPTCHAs' usability by analyzing collected player inputs. Since DevilTyper provides entertainment 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 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

Additional Key Words and Phrases: Games with a purpose (GWAP), human computation, human perception, optical character recognition, readability

**ACM Reference Format:**
Ho, C.-J., Wu, C.-C., Chen, K.-T., and Lei, C.-L. 2011. DevilTyper: A game for CAPTCHA usability evaluation. ACM Comput. Entertain. 9, 1, Article 3 (April 2011), 14 pages.
DOI = 10.1145/1953005.1953008 http://doi.acm.org/10.1145/1953005.1953008

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.

Authors' addresses: C.-J. Ho, Computer Science Department, University of California, Los Angeles; C.-C. Wu, Department of Electrical Engineering, National Taiwan University; K.-T. Chen (corresponding author), Institute of Information Science, Academia Sinica; email: ktchen@iis.sinica.edu.tw; C.-L. Lei, Department of Electrical Engineering, National Taiwan University.

DOI 10.1145/1953005.1953008 http://doi.acm.org/10.1145/1953005.1953008

## 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) [von Ahn et al. 2003] is known as an effective and widely used solution. Although there are many types of CAPTCHAs [Kochanski et al. 2002; Datta et al. 2005], the 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 humans can still tell the text clearly.

As many CAPTCHAs have been broken by OCR (Optical Character Recognition) or other recognition techniques [Mori and Malik 2003; Chellapilla and Simard 2004; pwn 2011], 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, for example, with too noisy background or too much text distortion. Therefore, we need to seek the balance in the trade-off 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 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 [Ho and Chen 2009] 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 [von Ahn and Dabbish 2004], Peekaboom [von Ahn et al. 2006b], and KissKissBan [Ho et al. 2009], 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 article, 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 [Mori and Malik 2003] and pattern recognition [Chellapilla and Simard 2004] have shown the vulnerability of the CAPTCHA design. New methods for generating CAPTCHAs are also proposed to enhance the robustness of CAPTCHAs [Datta et al. 2005; Rusu and Govindaraju 2004].

At the same time, only a few works studied the usability issues of CAPTCHAs. In Yan and El Ahmad [2008], the authors discussed a variety of factors that should be discussed when designing CAPTCHAs. Chellapilla et al. [2005] have 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 calculated users' accuracy in solving CAPTCHAs to measure the tests' usability under different distortion and noise settings. Wang and Bentley [2006] and Baird and Riopka [2005] discussed users' familiarity of words and degradation of images in CAPTCHAs, and Chellapilla et al. [2005] evaluated the usability and computer vision techniques in single character recognition of CAPTCHAs.

The just mentioned works evaluated the usability of CAPTCHAs or the trade-off 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 datasets is called Games With A Purpose (GWAP) [von Ahn 2006]. 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 [von Ahn and Dabbish 2004], locating objects in images [von Ahn et al. 2006b], and commonsense collection [von Ahn et al. 2006a].

## 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 Web sites, 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 a 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.

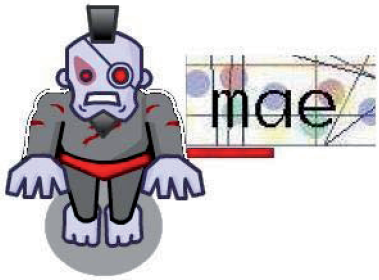


Fig. 1. A CAPTCHA is attached to each devil. Players can defeat the devils by typing the distorted text in CAPTCHAs.

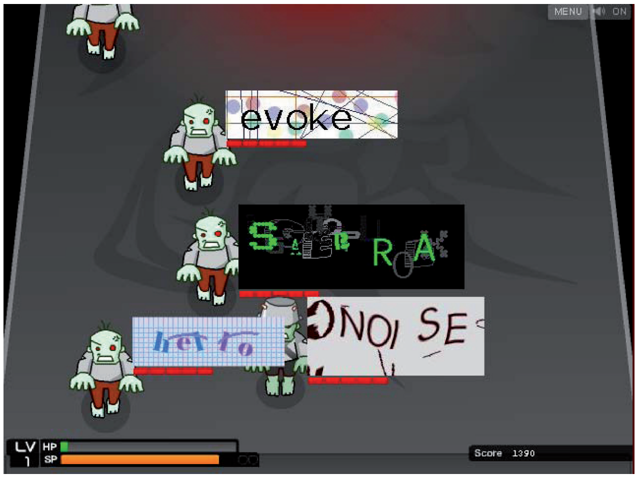


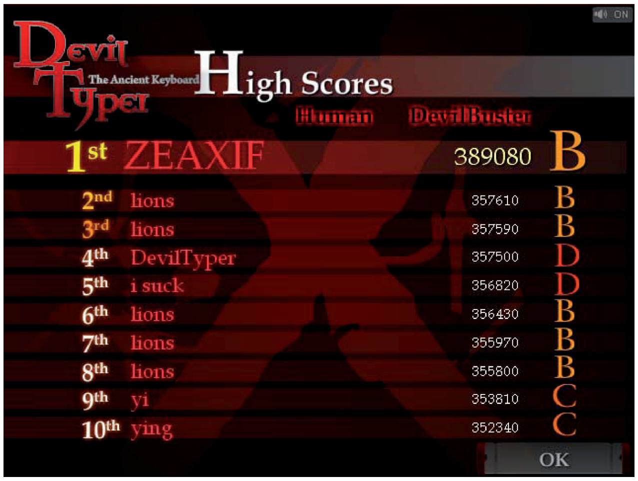
Fig. 2. The screenshot of the game. The devils move from the upside to the downside. Players will lose life points if the devils reach the bottom of the game screen.

### 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 got hurt by the devil. To target a devil, the player must type the first letter of its corresponding CAPTCHA. Once the player targets a



| | | | |
|---|---|---|---|
| 1st | ZEAXIF | 389080 | B |
| 2nd | lions | 357610 | B |
| 3rd | lions | 357590 | B |
| 4th | DevilTyper | 357500 | D |
| 5th | i suck | 356820 | D |
| 6th | lions | 356430 | B |
| 7th | lions | 355970 | B |
| 8th | lions | 355800 | B |
| 9th | yi | 353810 | C |
| 10th | ying | 352340 | C |

Fig. 3. The screenshot of the score board of DevilTyper.

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 wants 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 introduce different levels of challenge so that users will continue the play with higher levels and feel accomplished. In von Ahn [2006], the author 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 associated 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.

[1] http://deviltyper.iis.sinica.edu.tw/

### 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 its 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.

## 4. DEVILTYPER DATA VALIDATION

In this section, we conduct experiments to validate whether the data collected by DevilTyper is consistent with that 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[2] 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. Selected CAPTCHAs

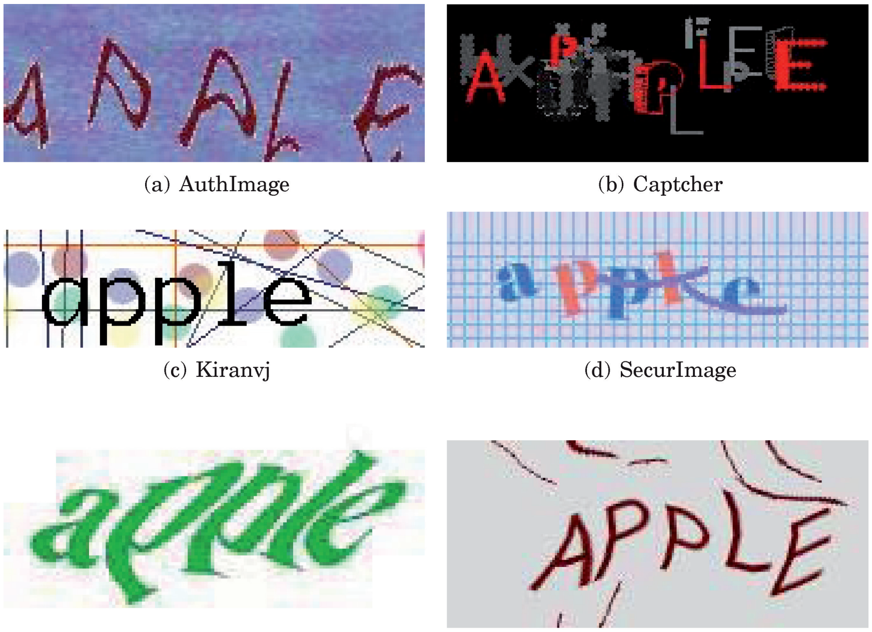
We choose six CAPTCHAs in the experiments and use a plain-text CAPTCHA as a baseline. As shown in Figure 4, the CAPTCHAs we provide are AuthImage, Captcher, Kiranvj, SecurImage, CoolCAPTCHA, and TgCAPTCHA.

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, for example, 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 usability by observing the different performance metrics players exhibited in solving the tests. The results are shown in Figure 5. It shows that the relative usability of the six CAPTCHAs remains consistent regardless of the performance metric used. Therefore, we can easily

[2] PTT 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.



(a) AuthImage (b) Captcher

(c) Kiranvj (d) SecurImage

(e) CoolCAPTCHA (f) TgCAPTCHA

Fig. 4. CAPTCHA schemes used in our experiments.

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 Kiranvj and plain text are similar.

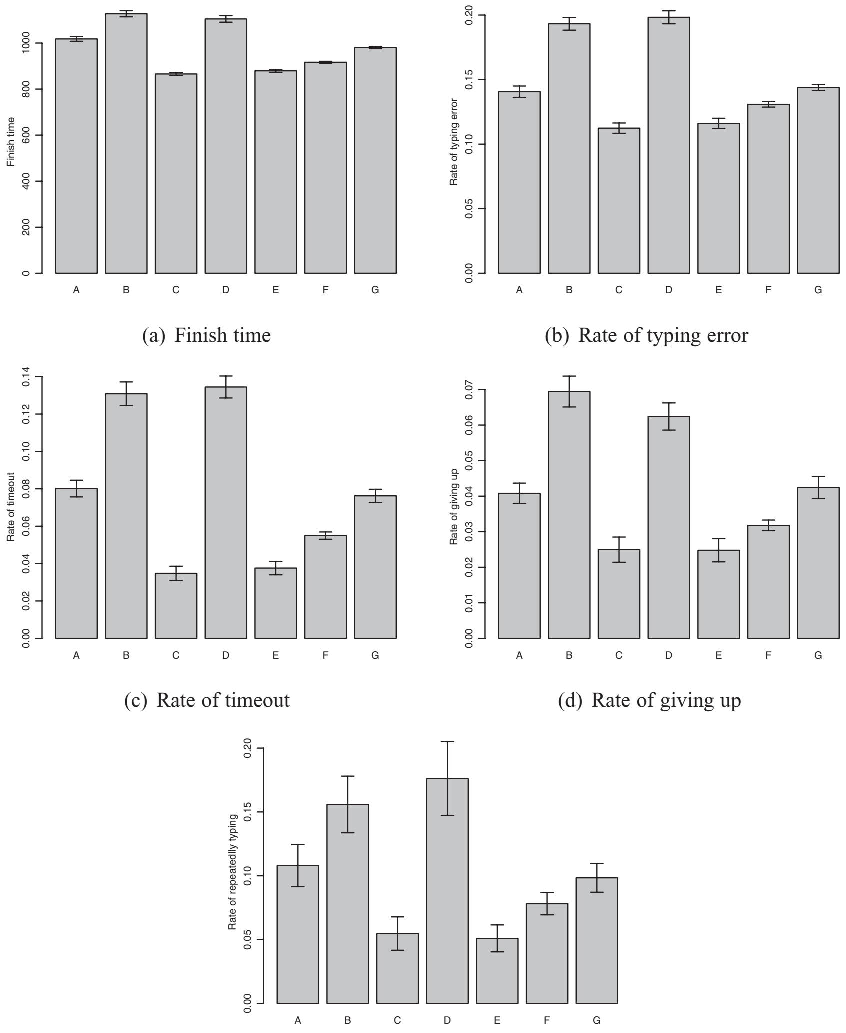
### 4.4. Consistency between Performance Metrics

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 microtask crowdsourcing platform, Amazon Mechanical Turk[3]. On Mechanical Turk, anyone can publish tasks and pay small amount of money, for example, $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 for a reward of $0.03, where the workers’ accuracy and elapsed time in solving each CAPTCHA were recorded. The task was accomplished

[3] http://mturk.com/



(a) Finish time

(b) Rate of typing error

(c) Rate of timeout

(d) Rate of giving up

(e) Rate of repeat typing

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

500 times by 44 workers, and 7,500 CAPTCHAs solving tasks were performed. Comparing DevilTyper and MechanicalTurk in terms of the number of CAPTCHAs per U.S. dollar, the ratio was 46900 versus 500, approximately 95 versus 1, which proves the economic advantage of DevilTyper in data collection.

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 user performance is generally consistent no matter whether 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 no monetary cost.

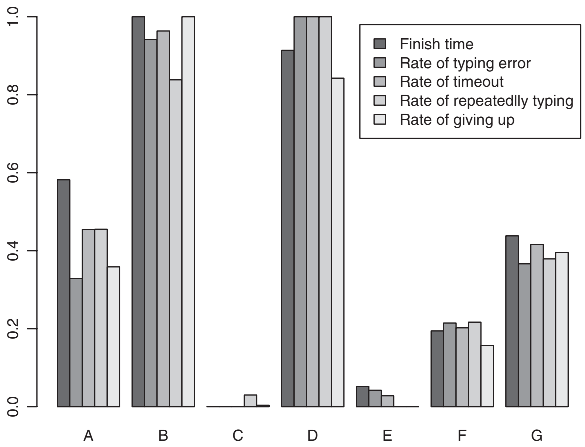


Fig. 6. A comparison of normalized performance metrics which capture how players solve CAPTCHAs.

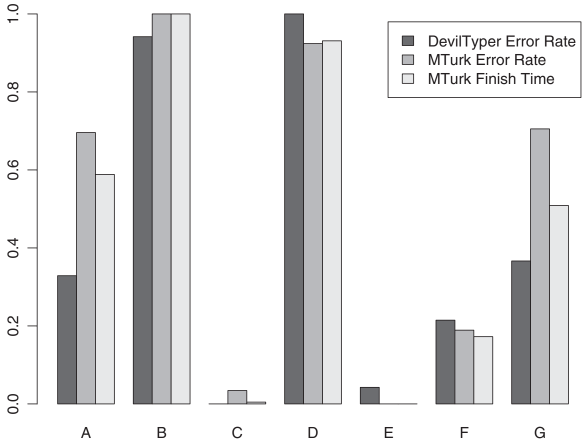


Fig. 7. A comparison of CAPTCHA solving error rates on Mechanical Turk and DevilTyper.

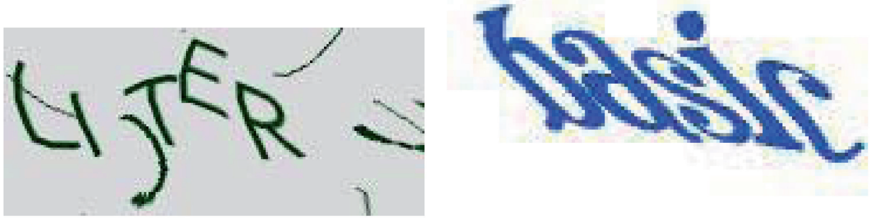
## 5. DESIGN FACTOR ANALYSIS

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 confusing characters and improve 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,



(a) “LITER” with the TgCAPTCHA scheme (b) “BASIC” with the CoolCAPTCHA scheme

Fig. 8. Examples illustrating the combined interaction of the CAPTCHA design and characters. While “i” is clearly recognizable in CoolCAPTCHA, it is often confused with the short lines in background noise in TgCAPTCHA.

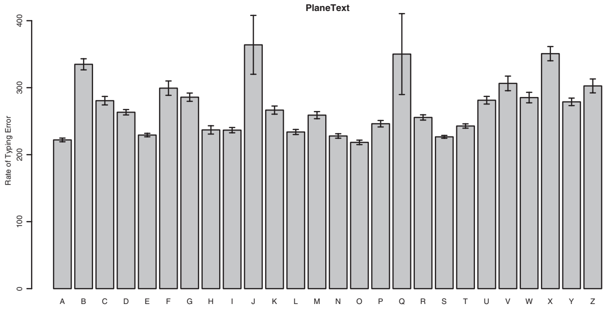


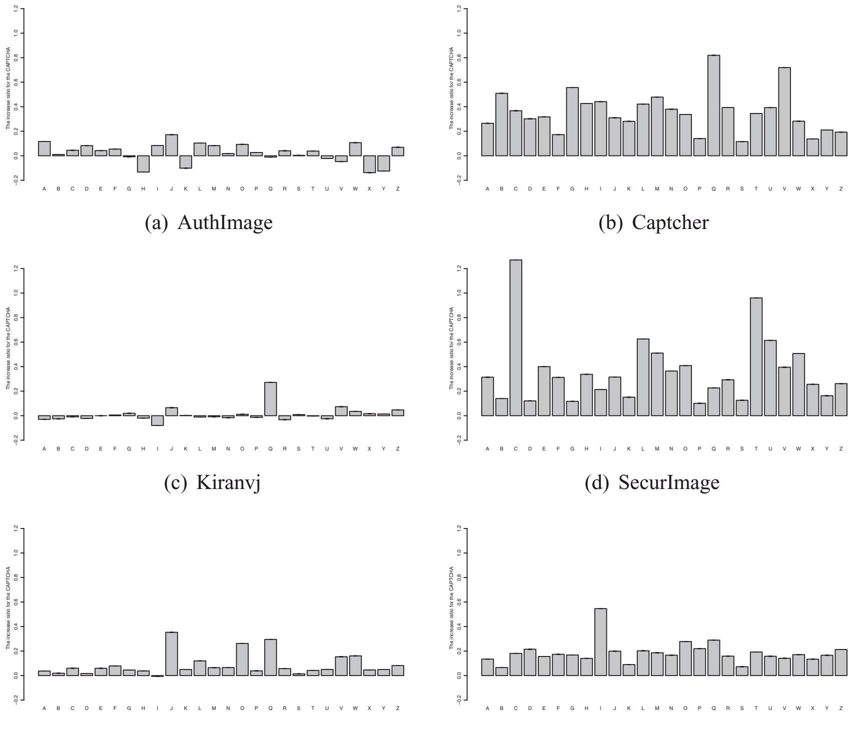
Fig. 9. Player typing time for characters in plain text.

players have different typing error rates 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 is presented in Figure 10.

From the graph, we can see that the effects of the 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 that are 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 to 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



(a) AuthImage

(b) Captcher

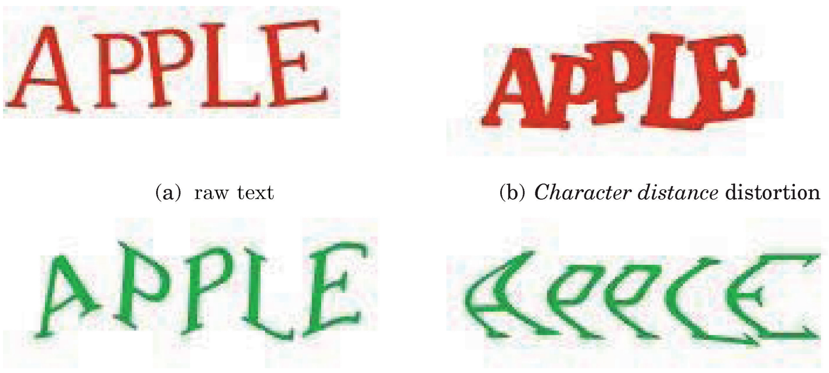
(c) Kiranvj

(d) SecurImage

(e) CoolCAPTCHA

(f) TgCAPTCHA

Fig. 10. Single character analysis.



(a) raw text

(b) *Character distance* distortion

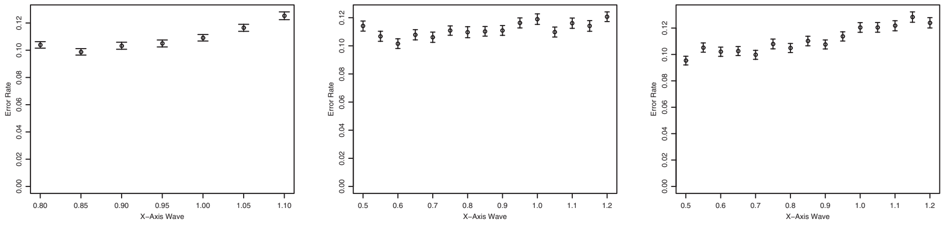
(c) *X-axis wave* distortion

(d) *Y-axis wave* distortion

Fig. 11. CoolCAPTCHA examples generated using different distortion strategies.

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

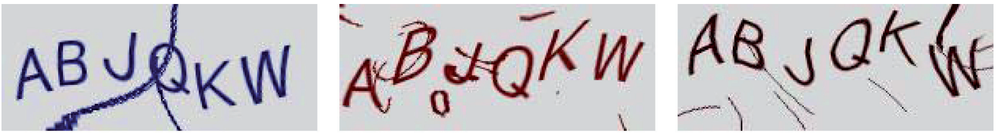


(a) character distance vs. error rate

(b) x-axis wave vs. error rate

(c) y-axis wave vs. error rate

Fig. 12. Users' average error rates with CAPTCHAs rendered using different distortion strategies.



(a) *long arcs* noise

(b) *short arcs* noise

(c) *short lines* noise

Fig. 13. TgCAPTCHA examples generated using different noise addition strategies.

Figure 12(a), this parameter significantly influences 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.

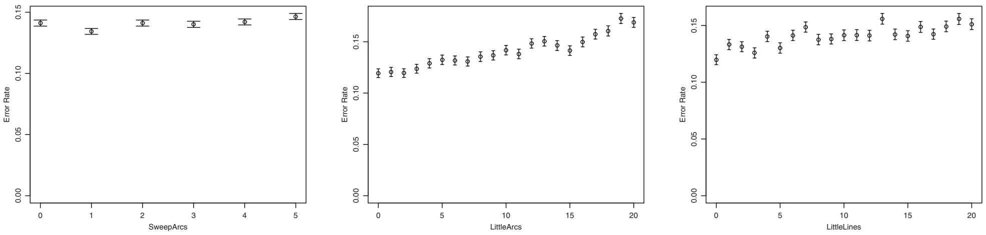
*Y-Axis Wave.* Similar to the x-axis wave parameter, the *y-axis wave* controls the degree of sine-wave distortions of characters along the y-axis, which we set within the range of 0.5 and 1.2 in our experiments. The results, as shown in Figure 12(c), indicate that the y-axis distortions lead to a much 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 Microsft 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.

*Long Arcs.* The *long arcs* parameter controls the number of long arcs overlaid on the image, where the position, length, and curvature of the arcs are randomly chosen. In our experiment, we set this parameter between 0 and 5. From Figure 13(a), we can see that the long arcs do not influence the usability of the CAPTCHAs significantly even when 5 long arcs were added.

*Short Arcs.* Similar to long arcs, the *short arcs* parameter controls the number of short arcs overlaid on the image. In our experiment, the number of short arcs are randomly drawn from the range 0 to 20. Interestingly, while long arcs do not impact the CAPTCHA's usability, short arcs do, as shown in Figure 13(b). We believe this is



(a) long arcs vs. error rate (b) short arcs vs. error rate (c) short lines vs. error rate

Fig. 14. Users' average error rates with CAPTCHAs rendered using different noise addition strategies.

due to the length of short arcs, which are similar to that of the character strokes so that short arcs are more likely to interfere with distorted text and increase the difficulty of text recognition.

*Short Lines.* The *short lines* parameter controls the number of short lines overlaid on the rendered CAPTCHA. As with long and short arcs, the position, length, and direction of each segment is randomly decided. Our results show that users' average error rates slightly but steadily increase with more short lines, as shown in Figure 13(c). However, the impact of short lines is slightly less than that of short arcs, which is reasonable because arcs are more like the strokes of distorted text and therefore more interference on readers' recognition is induced.

## 6. CONCLUSION

In this article, 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 CAPTCHAs. More importantly, such an ample amount of data enables researchers to quantify the usability of various CAPTCHAs and their design factors in a rigorous statistical approach even when 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 CAPTCHAs 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 CAPTCHAs.

## REFERENCES

Baird, H. S. and Riopka, T. 2005. ScatterType: A reading CAPTCHA resistant to segmentation attack. In *Proceedings of SPIE Document Recognition and Retrieval Conference*. 16–20.

Chellapilla, K., Larson, K., Simard, P., and Czerwinski, M. 2005. Designing human friendly human interaction proofs (HIPs). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*. 711–720.

Chellapilla, K. and Simard, P. 2004. Using machine learning to break visual human interaction proofs (HIPs). In *Proceedings of Advances in Neural Information Processing Systems 17 (NIPS'04)*. MIT Press.

Datta, R., Li, J., and Wang, J. Z. 2005. IMAGINATION: A robust image-based CAPTCHA generation system. In *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA'05)*. 331–334.

Ho, C.-J., Chang, T.-H., Lee, J.-C., Hsu, J. Y.-j., and Chen, K.-T. 2009. KissKissBan: A competitive human computation game for image annotation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP'09)*. 11–14.

Ho, C.-J. and Chen, K.-T. 2009. On formal models for social verification. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP'09)*. 62–69.

KOCHANSKI, G., LOPRESTI, D., AND SHIH, C. 2002. A reverse turing test using speech. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)*. 1357–1360.

MORI, G. AND MALIK, J. 2003. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *Proceedings of Computer Vision and Pattern Recognition Conference*. 134–141.

PWNtcha. 2011. CAPTCHA decoder. http://libcaca.zoy.org/wiki/PWNtcha/.

RUSU, A. AND GOVINDARAJU, V. 2004. Handwritten CAPTCHA: Using the difference in the abilities of humans and machines in reading handwritten words. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*. 226–231.

VON AHN, L. 2006. Games with a purpose. *Comput. 39,* 6, 92–94.

VON AHN, L., BLUM, M., HOPPER, N. J., AND LANGFORD, J. 2003. CAPTCHA: Using hard ai problems for security. In *Proceedings of the Eurocrypt Conference*. 294–311.

VON AHN, L. AND DABBISH, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*. 319–326.

VON AHN, L., KEDIA, M., AND BLUM, M. 2006a. Verbosity: A game for collecting common-sense facts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*. ACM Press, 75–78.

VON AHN, L., LIU, R., AND BLUM, M. 2006b. Peekaboom: A game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*. ACM, New York, 55–64.

WANG, S.-Y. AND BENTLEY, J. L. 2006. CAPTCHA challenge tradeoffs: Familiarity of strings versus degradation of images. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*. 164–167.

YAN, J. AND EL AHMAD, A. S. 2008. Usability of CAPTCHAs or usability issues in CAPTCHA design. In *Proceedings of the 4th Symposium on Usable Privacy and Security (SOUPS'08)*. ACM, New York, 44–52.

Received September 2010; accepted October 2010