

Designing Human Friendly Human Interaction Proofs (HIPs)

Kumar Chellapilla, Kevin Larson, Patrice Simard and Mary Czerwinski

Microsoft Research

1 Microsoft Way, Redmond, WA, USA 98052

{kumarc, kevlar, patrice, marycz}@microsoft.com

ABSTRACT

HIPs, or Human Interactive Proofs, are challenges meant to be easily solved by humans, while remaining too hard to be economically solved by computers. HIPs are increasingly used to protect services against automatic script attacks. To be effective, a HIP must be difficult enough to discourage script attacks by raising the computation and/or development cost of breaking the HIP to an unprofitable level. At the same time, the HIP must be easy enough to solve in order to not discourage humans from using the service. Early HIP designs have successfully met these criteria [1]. However, the growing sophistication of attackers and correspondingly increasing profit incentives have rendered most of the currently deployed HIPs vulnerable to attack [2,7,12]. Yet, most companies have been reluctant to increase the difficulty of their HIPs for fear of making them too complex or unappealing to humans. The purpose of this study is to find the visual distortions that are most effective at foiling computer attacks without hindering humans. The contribution of this research is that we discovered that 1) automatically generating HIPs by varying particular distortion parameters renders HIPs that are too easy for computer hackers to break, yet humans still have difficulty recognizing them, and 2) it is possible to build segmentation-based HIPs that are extremely difficult and expensive for computers to solve, while remaining relatively easy for humans.

Author Keywords

Visual letter recognition, Evaluation, Human Perception, Human Interaction Proofs (HIPs), Completely Automated Public Turing Tests to Tell Computers and Humans Apart

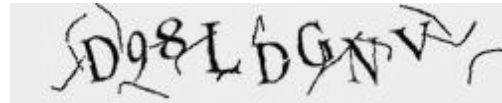


Figure 1: An example character based HIP

(CAPTCHAs), Computer vision.

ACM Classification Keywords

H.5.2f. Information interfaces and presentation (HCI): Graphical user interfaces.

INTRODUCTION

HIPs, or Human Interactive Proofs, are challenges meant to be easily solved by humans while remaining too hard to be solved economically by computers (see Figure 1). For instance, a HIP challenge (or HIP) could be a pixel image of distorted characters, and the proper response would be the ASCII string of corresponding characters (in this case, D98LDGNV).

HIPs are increasingly used to protect services against automatic script attacks. Examples of such services include email (spam), online registrations (fraud, denial of service, or DoS), ticket/event reservations (DoS), online voting (stuffing), login (DoS), chat rooms, weblogs, etc. Many companies such as Yahoo, Microsoft, TicketMaster, Register.com, and Google, are currently using HIPs to protect their online services. To be effective, a HIP must be difficult enough to discourage script attacks by raising the computation and/or development costs of breaking the HIP to an unprofitable level. At the same time, the HIP must be easy enough to not discourage humans from using the service. Early HIP designs have successfully met these criteria [1]. For instance, when MSN Hotmail deployed its first HIP, hotmail registrations dropped by 19% without impacting customer support inquiries. A study of the data revealed that the drop corresponded to mail accounts acquired by scripts for the purpose of spamming. However, the growing sophistication of attackers and increasing profit incentives have rendered most of the currently deployed HIPs vulnerable to attacks [2,12]. Yet, most companies have been reluctant to increase the difficulty of their HIPs

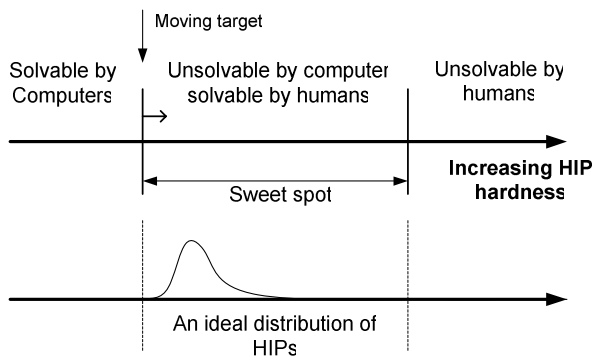


Figure 2: Regions of feasibility as a function of HIP difficulty for humans and computer algorithms.

for fear of making them too complex or unappealing to humans. This has raised an important question: Is it possible to design human-friendly HIPs that are easy for humans but difficult for computers?

Work on distinguishing computers from humans traces back to the original Turing test [3] which asks that a human distinguish between another human and a machine by asking questions of both. In contrast, we are interested in building a computer program designed to distinguish between another computer program and a human [4]. Such programs have been called reverse Turing tests, HIPs, or CAPTCHAs (Completely Automated Public Turing Tests to Tell Computer and Human Apart) [6]. An overview of this work can be found in [5]. Construction of HIPs of practical value is difficult because it is not sufficient to develop challenges to which humans are somewhat more successful than machines. This is because the cost of failure from using machines to solve the puzzles may be very small. In practice, if one wants to block automated scripts, a challenge at which humans are about 90% successful and machines are 1% successful, may not be sufficient, especially when the cost of failure and repetition is low for the machine [2,7,12]. At the same time, the identical challenge must not put too much burden on the human in order to avoid discouraging the use of the service. This is summarized in Figure 2.

The figure shows an ideal distribution of HIPs. The sweet spot, where the HIPs are easy for humans to recognize but difficult for hackers to crack, is not guaranteed to actually exist. Furthermore, automatically generated HIPs, being random in nature, will have a distribution of difficulty, with some particular instances extending beyond the hypothesized sweet spot. Depending on the cost of the attack and the value of the service, automatic scripts should not be more successful than 1 in 10,000 (0.01%) and the human success rate should approach at least 90%. While the latter is a common requirement for reducing the number of retries a human user has to endure, the former is obtained by analyzing the cost of hiring humans to solve HIPs. For example, requiring a signup HIP for creating an e-mail account only imposes a maximal cost

of about .002 cents per message, while the minimum estimate for the costs/potential revenue from sending spam are around .0025 cents, with many spammers charging or earning 5 to 10 times that [12]. The sweet spot will decrease in size over time as computers get faster, attackers get more sophisticated, and HIPs are specifically targeted. Unfortunately, humans are unlikely to get better at solving HIPs in the same timeframe [10,11].

We have come across dozens of proposals for HIP designs, ranging from counting objects in a picture, segmenting faces, recognizing animations, identifying words in audio, etc. [6]. Among visual challenges, character identification is the most obvious favorite because 1) OCR (optical character recognition) is a well studied field and the state of the art is well known, 2) characters were designed by humans for humans and humans have been trained at the task since childhood, 3) each character has a corresponding key on the keyboard and 8 keystrokes span a space of over 1000 billion solutions, 4) the task is easily understood by users without much instruction, and 5) character-based HIPs can be generated quickly (300 8-character HIPs per second on a 3GHz P4 [2,12]). Owing to these merits, character-based HIPs have been adopted by several companies to protect various services on the web. A few examples are presented below:

Mailblocks: While signing up for free email service with mailblocks (www.mailblocks.com), one will find HIP challenges of the type: shown in Figure 3(a).

MSN: Starting in July 2004, MSN introduced their second generation HIP. While signing up for free e-mail with MSN Hotmail (www.hotmail.com), one will find HIP challenges of the type shown in Figure 3(b).

Register.com: While requesting a whois lookup for a domain at www.register.com, one will find HIP challenges of the type shown in Figure 3(c).

EZ-Gimpy (CMU): While signing up for free e-mail service with Yahoo! (www.yahoo.com) before August 2004, one received HIP challenges of the type shown in Figure 3(d).

Yahoo!: Starting in August 2004, Yahoo! introduced their second generation HIP. Two examples are presented in Figure 3(e).

Ticketmaster: While looking for concert tickets at www.ticketmaster.com, one will receive HIP challenges of the type shown in Figure 3(f).

Google/Gmail: While signing up for free e-mail with Gmail at www.google.com, one will receive HIP challenges of the type shown in Figure 3(g).

So, while solutions to Yahoo HIPs (before August '04) are common English words, those for Ticketmaster and Google do not necessarily belong to the English



Figure 3(a): Mailblocks HIP samples.



Figure 3(b): MSN HIP samples.

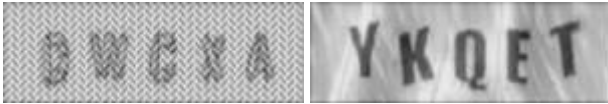


Figure 3(c): Register.com HIP samples.

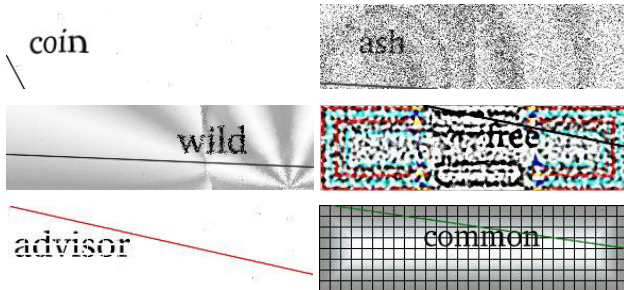


Figure 3(d): EZ-Gimpy (CMU) HIP samples.



Figure 3(e): Yahoo! HIP samples.

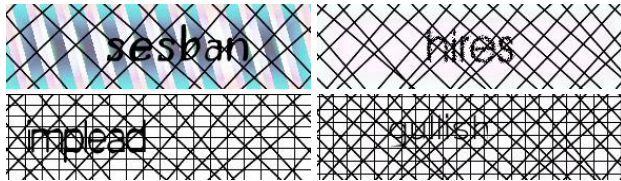


Figure 3(f): Ticketmaster HIP samples.



Figure 3(g): Google HIP samples.

dictionary. They appear to have been created using a phonetic generator [8].

Recently, each of these HIPs has been systematically broken, with a success rate of 5% or greater at a rate of 300 attempts per second [2,12]. In each case, the attack was based on a dedicated segmentation attack, followed by a generic recognition attack based on machine learning. The need for much harder HIPs has brought the existence of a sweet spot into question, and has forced us

to examine the computer versus human side of the problem.

Computers are very good at OCR. In fact, the state-of-the-art algorithms are very close to human performance on single printed character recognition and some commercial OCR systems can achieve speeds of 1000 recognized characters per second. Recognizing letters such as the one below



can be done with a 95%+ success rate using machine learning, given letters with similar distortions [9].

The story is quite different, however, when the location of the character(s) is not known a-priori. The problem of detection, or segmentation, has remained a challenging problem in the fields of handwriting recognition, speech recognition and computer vision for the last two decades. For example, in the following image,



finding where the letter is (segmentation) and which letter it is (recognition) can yield several false positives. Furthermore, the recognition rate falls to single digit numbers.

Segmentation is intrinsically difficult for both computers and humans because 1) it is computationally expensive, since every position must be tested for a potential candidate, 2) the space of input is very large since it includes all the non-valid characters, and 3) identifying the valid characters in the image is a combinatorially difficult problem. For instance, correctly identifying which 8 characters among 20 candidates (assuming 12 false positives), has a 1 in 125,970 (20 choose 8) chance of success by random guessing.

In light of the results from [2,12], successful HIPs will have to rely on segmentation problems that are difficult for computers, rather than simpler character recognition problems where the location of each character is easy to infer. One of the goals of this paper is to better understand human segmentation and recognition capabilities, for the purpose of designing a segmentation-based HIP that is easy for humans but difficult for computers.

The fact that we do not fully understand how humans do segmentation is important. Notably, despite a geometric growth in computational power, segmentation tasks such as cursive handwriting, continuous speech and vision have only made marginal progress in the last 10 years. We can conjecture that without a scientific breakthrough, even a 100X speedup in computers would only yield marginal improvements in current segmentation

algorithms. In all likelihood, humans will remain much better at segmentation than computers for a few more years. Our hope is therefore to make our HIPs “segmentation complete”.

Character distortions

Character-based HIPs employ a set of character distortions to make them hard to OCR using computers. The basic character transformations include translation, rotation (clockwise or counterclockwise), and scaling. Rotation is usually less than 45 degrees to avoid converting a 6 into a 9, an M into a W etc. Both computers and humans find HIPs, using these three transformations, easy to solve (see Figures 5, 6, and 7). To increase the difficulty of computer-based OCR, we introduced two kind of warping:

1. **Global Warp:** The global warp produces character-level, elastic deformations. It is obtained by generating a random displacement field followed by a low pass filter. The resulting displacement field is then applied to the image with interpolation. These appear to bend and stretch the given characters.



The purpose of these elastic deformations is to foil template matching algorithms.

2. **Local Warp:** Local warp is intended to produce small ripples, waves, and elastic deformations along the pixels of the character, i.e., at the scale of the thickness of the characters, rather than the scale of the width and height of the character. The local warp deformations are generated in the same manner as the global warp deformations, by changing the low pass filter cut-off to a higher frequency. The purpose of the local warp is to foil feature-based algorithms which may use character thickness or serif features to detect and recognize characters.



We have verified that the warp distortions are effective at breaking commercial OCR (e.g. Scansoft’s OCR). However, HIPs generated using random characters and these five distortions (assuming characters do not touch or overlap) can still easily be solved by computers using a sophisticated training algorithm and a sufficient amount of data [9]. As a result they are very vulnerable to computer OCR attacks, and indeed all the HIPs deployed by major companies have been broken [2,12]. To pose a much more difficult computer segmentation problem, clutter is introduced into the HIP. Crisscrossing straight lines and arcs, background textures, and meshes in foreground and background colors are commonly introduced as clutter. Increasing the clutter density typically increases the segmentation difficulty. In this

paper, we use random arcs of different thicknesses as clutter.

USER STUDY I

We carried out a series of studies in an effort to examine the effects that a number of parameters would have on human observers of the resultant HIPs. The studies were designed to be run electronically, allowing participants to do the HIP recognition tasks from the comfort of their own offices. 76 users were recruited to participate in the first set of experiments. All were employees at a large software company. Average age of the participants was 35.2 (range of 22-54 years of age), 17 were female, and all but 12 had normal or corrected-to-normal vision. In addition, 42 wore glasses or contacts of some kind. All but six of the participants had at least an undergraduate education (six responded “other” which could have included a PhD).


For the first study, it was important that we obtained some baseline measures of some of the parameters independent of each other prior to examining their difficulty for users in combination. For that reason, we chose the dimensions of translation, scaling, rotation and global warping of the HIP characters. The experiment was designed to present several difficulty levels of these four factors independently to users and determine the users’ accuracy at solving the resulting HIPs. Accuracy was defined as the percentage of characters correctly recognized. For example, for 8 character HIPs, getting on average 7 characters correct would imply an accuracy of 87.5 percent. In another set of studies, combinations of these variables, as used today in actual HIP generation, were presented to users to see where human observers begin to have problems. The hypothesis was that human accuracy would be almost 100%, even with increasing difficulty values for the individually observed parameters. Additionally, it is well known that these initial parameters are quite easily solved by computers using standard recognition techniques [2,12].

Because we expected human performance to be at a very high level for the individual parameters, a non-zero baseline combination of parameters for translation, scaling, rotation, and global warp was chosen for inclusion for a small number of trials in the first study. This “baseline” combination was included because it was expected to be more difficult for the humans and computers. The baseline combination was then added to one of the parameters, local warping, as it was our hypothesis that this set of parameters might be hard for humans.

Since we had no benchmark performance data to base any of these hypotheses on, we had to come up with difficulty levels based on our own piloting of the HIPs at the various levels. Two experimenters ran through a series of trials at all levels of HIP difficulty both for those varying on only one parameter and then for combinations of the



Figure 4: Example of Plain Text (M7F47VWC)



5MS9FVLL
3R2YAZ9X
C7AXBZZR

Figure 5: Example of Translated Text, levels 10 (5MS9FVLL), 25 (3R2YAZ9X), and 40 (C7AXBZZR)



PWVDYLVH
B5P4MMLB
GSB5776E

Figure 6: Example of Rotation Text, levels 15 (PWVDYLVH), 30 (B5PYMMLB), and 45 (GSB5776E)

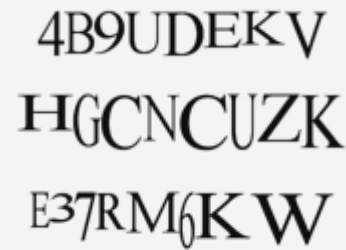
parameters. This pilot data was then used to drive the settings for the parameters used in the HIP creation in the studies so that they ranged from very easy to very difficult trials (if possible) for each individual variable, and then finally for the initial combinations we chose to study.

The pilot study then led to our including the following settings for the first set of experiments. Only one HIP was created at each parameter level, and each participant saw that same exact HIP in a predetermined, randomized order. The seven parameters tested in the first user study were plain (or undistorted) text, translated text, rotated text, scaled text, global warping, local warping, and local warping combined with all the previous parameters.

In summary, participants were provided with a website for viewing 68 HIPs and for entering what they thought the solutions to the puzzles were. If the HIP was deemed to be unreadable by the participants, they could enter “unreadable” by pressing a button provided on the website for that trial. Each response provided by the participant was recorded, as was the response time prior to completing each HIP. Total time to complete the experiment was approximately 15 minutes. If desired, participants were encouraged to enter some contact information in order to receive a lunch/dinner coupon for the local cafeteria.

Plain Text

Plain text is text that has not been altered with any distortions, as shown in Figure 4. Participants were very accurate at identifying the plain text letters. 73



4B9UDEK V
HGCNCU ZK
E37RM6K W

Figure 7: Example of Scaled Text, levels 20, 35, and 50

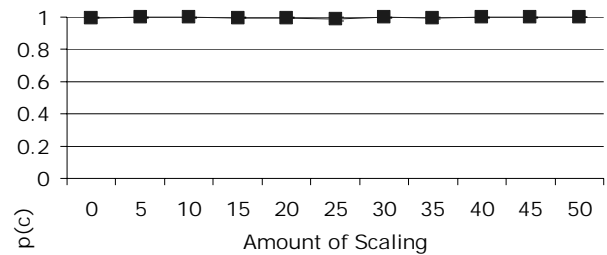


Figure 8: Accuracy rate for scaled text

participants recognized all letters perfectly, while 3 participants missed a single letter.

Translated Text

Translated text is moved either up or down and left or right by an amount, as shown in Figure 5. We increased the amount of translation in nine steps from 0% to 40% of the character size. Participants had a very high accuracy rate with translated text. The accuracy rate was 99% or above for all levels.

Rotated Text

Rotated text is text that is turned either in a clockwise or counterclockwise direction as shown in Figure 6. We rotated text in ten incremental steps from 0 degrees to 45 degrees. Participants had a very high accuracy rate with rotated text. The accuracy rate was 99% or above for all levels.

Scaled Text

Scaled text is text that is stretched or compressed in the x-direction and stretched or compressed in the y-direction, as shown in Figure 7. We scaled the text in eleven incremental steps from 0% to 50%, as shown in figure 8. Participants had a very high accuracy rate with scaled text, as shown in Figure 8. The accuracy rate was 98% or above for all levels.

Global Warp Text

Global warp is a warping field that covers an entire eight character HIP, as shown in Figure 9. Each character's twists and turns are dependent on the twists and turns of nearby characters. We increased the amount of global warping in 11 incremental steps from 0 to 390, as shown



Figure 9: Example of Global Warp Text, levels 180 (UHYE8VBL), 270 (B3277UHF), and 360 (GLX45BMS)

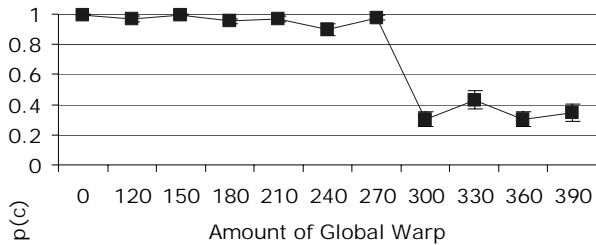


Figure 10: Accuracy rate for global warp text

in Figure 10. The global warp value indicates the magnitude of the global warp field and is proportional to the average movement of ink pixels in the HIP.

Participants had a very high accuracy rate with levels of global warp up to level 270. Accuracy drops off dramatically with more global warp, as shown in Figure 10. A One-Way ANOVA shows that accuracy is reliably different for levels of global warp, $F(10,65) = 73.08$, $p < .001$. Post-hoc tests show that the 0-270 levels of global warp are reliably different from the 300-390 levels of global warp at the $p < .05$ level, using Bonferroni corrections for multiple tests in this and all following post-hocs.

Local Warp Text

Local warp is a warping field that is independent for each character within a HIP, as shown in Figure 11. The twists and turns on one character have no bearing on the twists and turns of the next character. The local warp was incremented in 16 steps from 0 to 90, as shown in Figure 12. The local warp value indicates the magnitude of the local warp field and is proportional to the average movement of ink pixels in the HIP. Participants had a very high accuracy rate with levels of local warp up to level 45, and very poor accuracy at level 70 and above, as shown in Figure 12. A One-Way ANOVA shows that accuracy is reliably different for levels of local warp, $F(15,60) = 120.24$, $p < .001$. Post-hoc tests indicate that levels 0-60 are reliably different from levels 65-90.

Local Warp plus Baseline Text

This parameter is a complex interaction between the conditions that have already been used. The amount of translation, rotation, scaling, and global warp is held constant throughout these conditions and is called the

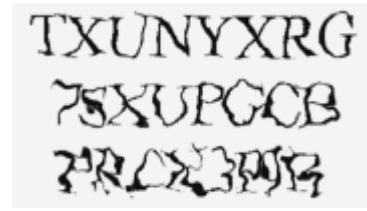


Figure 11: Example of Local Warp Text, levels 30 (TXUNYXRG), 55 (7SXUPGCB), and 80 (PRCK3P9R)

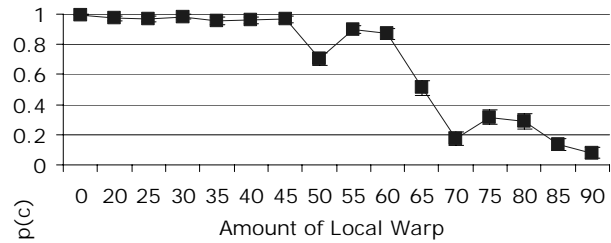


Figure 12: Accuracy rate for local warp text

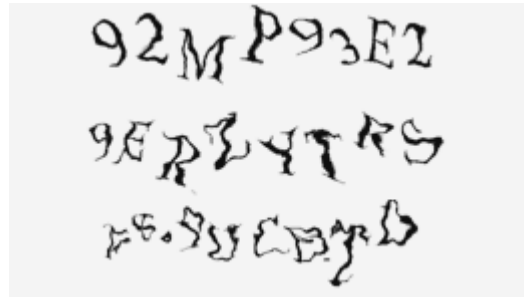


Figure 13: Example of Local Warp plus Baseline Text, levels 30 (92MP93E2), 55 (93RZYTRS), and 80 (F8PUCBTD)

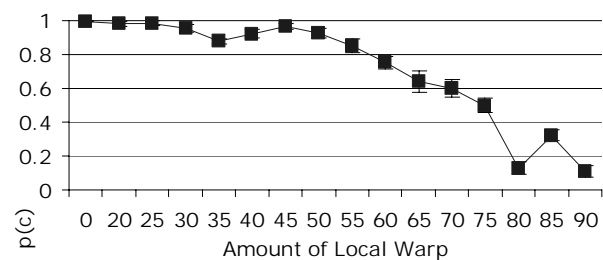


Figure 14: Accuracy rate for local warp plus baseline text

“baseline”. We used level 20 of translation, level 20 of rotation, level 20 of scaling, and level 75 of global warp. See Figure 13 for three examples of this condition. Local warp is manipulated with the same levels used in the previous parameter, as shown in Figure 14.

Participants had a high accuracy rate with local warp plus baseline up to level 55 of local warp. After level 50, accuracy decreased in gradual steps, as is shown in Figure 14. A One-Way ANOVA shows that accuracy is reliably different for levels of local warp plus baseline, $F(15,60) = 98.08$, $p < .001$. Post-hoc tests show that levels 0-55 are reliably difference from levels 70-90.

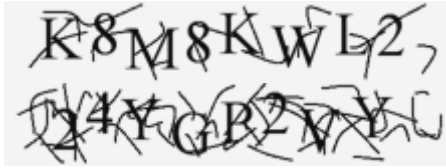


Figure 15: Example of Thin Arcs that Intersect, levels 18 (K8M8KWL2), and 36 (24YGP2VY)

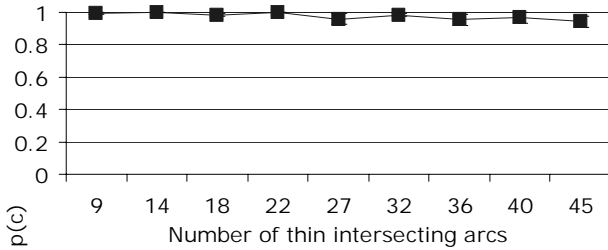


Figure 16: Accuracy rate for thin arcs that intersect

USER STUDY II

Twenty-nine more users from the same large software company were recruited for the second set of experiments. Average age of the participants was 35.2 (range of 26-54 years of age), 10 were female, and 23/29 had normal or corrected-to-normal vision. In addition, 19 wore glasses or contacts of some kind. All but six of the participants had at least an undergraduate education (once again 6 responded “other” which could have included a PhD). Despite the similarities in the profiles between participants in studies 1 and 2, only one participant participated in both studies.

The goals of the second study included examining the users’ performance on the baseline we had chosen for study 1, in addition to evaluating our next set of parameters that we thought humans might be able to handle quite easily. This set of HIP parameters, however, was a set that computers recognize much more poorly than the first set [2,12]. The hypothesis going into the second study was that it should be possible to find difficulty settings in some of these parameters (either alone or in combination with others) that are easy for humans to solve, but are very difficult for computers to “break”, at least using today’s recognition algorithms. Other than the new HIP examples, all of details of the study were identical to Study 1.

Thin Arcs that Intersect

In this condition, a HIP with a small amount of constant translation is manipulated with nine levels of thin squiggly lines – or *arcs*, as is shown in Figure 15. The arcs in this condition cross over the HIP’s characters. There are 9 levels of arcs ranging from 9 to 45 arcs across the HIP, as shown in Figure 16.

Participants had a very high accuracy rate with thin arcs that intersect. The accuracy rate was 94% or above for all

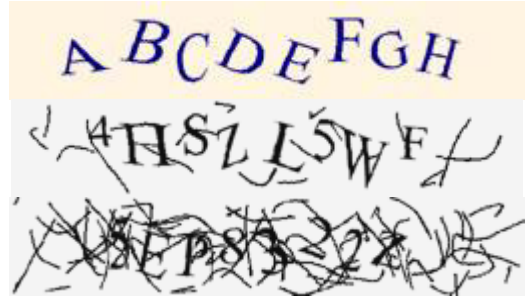


Figure 17: Example of Thin Arcs that Intersect plus baseline, levels 0(ABCDEFGH), 18 (4HSZL5WF), and 36 (5EP8322Z)

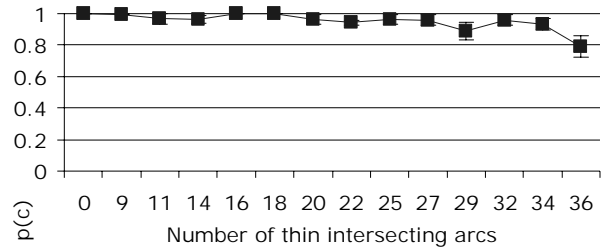


Figure 18: Accuracy rate for thin arcs that Intersect plus Baseline

levels, as shown in Figure 16. The differences between the numbers of arcs is not reliably different, $F(8,21) = 0.84, p > .05$.

Thin Arcs that Intersect plus Baseline

In this condition, a HIP with a small amount of constant translation, rotation, scaling and global warp is included with the manipulation of number of thin arcs that intersect. The arcs in this condition will cross over the HIP’s characters, as is shown in Figure 17. There are 14 levels of arcs ranging from 0 to 36 arcs across the HIP, as shown in Figure 18.

Participants had a high accuracy rate with thin arcs that intersect plus baseline, with accuracy above 90% for all but the highest number of arcs examined, as shown in Figure 18. A One-Way ANOVA shows that accuracy is reliably different for levels of thin arcs that intersect plus baseline, $F(13,16) = 2.70, p < .01$. Despite reliable main effects, post-hoc tests found no reliable differences between any two conditions.

Thick Arcs that Intersect

In this condition, a HIP with a small amount of constant translation is manipulated with nine levels of thick arcs that cross over the HIP characters, as shown in Figure 19. The number of arcs used range from 9 to 45, as shown in Figure 20.

Participants had more difficulty with thick arcs that intersect than with any other parameter in these studies. Accuracy stayed reasonably high through level 22 before dropping off considerably, as shown in Figure 20. A One-



Figure 19: Example of Thick Arcs that Intersect, levels 18 (FN2VMZ85), and 36 (SMZCWTTXX)

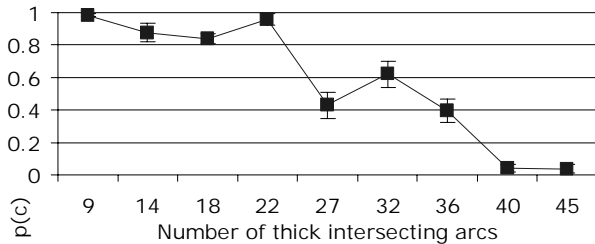


Figure 20: Accuracy rate for Thick Arcs that Intersect

Way ANOVA shows that accuracy is reliably different, $F(8,21) = 50.66$, $p < .001$. Post-hoc tests show that levels 9-22 are reliably different from levels 36-45.

Thick Arcs that Intersect plus Baseline

In this condition, a HIP with a small amount of constant translation, rotation, scaling, and global warp is combined with 14 levels of thick arcs that cross over the HIP characters, as shown in Figure 21. The number of arcs used ranged from 0 to 36, as shown in Figure 22.

Not surprisingly, thick arcs that intersect are also difficult for participants when the baseline distortions are also incorporated, as shown in Figure 22. A One-Way ANOVA shows that accuracy is reliably different, $F(13,16) = 49.27$, $p < .001$. Post-hoc tests show that levels 0-22 are reliably different from levels 27-36.

Thick Arcs that Don't Intersect

In this condition, a HIP with a small amount of constant translation is manipulated with nine levels of thick arcs that do not cross over the HIP characters, a few examples of which are shown in Figure 23. The number of arcs used ranges from 9 to 45, as shown in Figure 24.

Participants had a very high accuracy rate with thick arcs that don't intersect. The accuracy rate was 96% or above for all levels, as can be seen in Figure 24. The differences between levels of arcs is not reliably different, $F(8,21) = 0.96$, $p > .05$.

Thick Arcs that Don't Intersect Plus Baseline

In this condition, a HIP with a small amount of constant translation, rotation, scaling, and global warp is combined with 14 levels of thick arcs that do not cross over the HIP characters. A few examples are shown in Figure 25. The number of arcs used ranged from 0 to 36 (Figure 26).



Figure 21: Example of Thick Arcs that Intersect plus Baseline, levels 0 (ABCDEFGH), 18 (4L6PSPZL), and 36 (ZFG4N4ME)

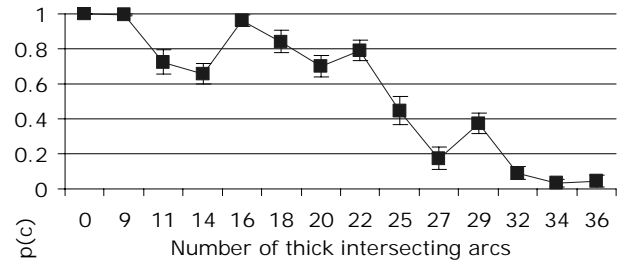


Figure 22: Accuracy rate for Thick Arcs the Intersect plus Baseline

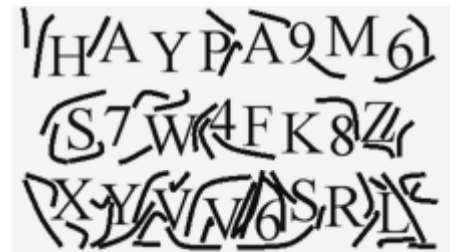


Figure 23: Example of Thick Arcs that don't intersect, levels 9 (HAYPA9M6), 18 (S7W4FK8Z), and 45 (XYVV6SRL)

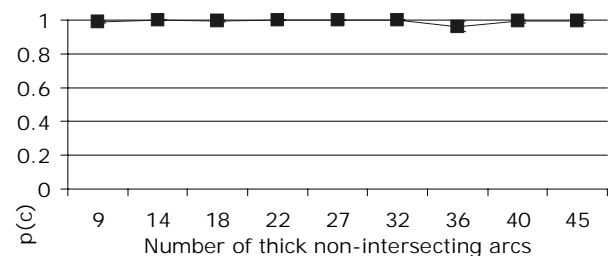


Figure 24: Accuracy rate for Thick Arcs that don't Intersect

Participants had a very high accuracy rate with thick arcs that don't intersect plus baseline distortions. The accuracy rate was 92% or above for all levels, as shown in Figure 26. The differences between levels of arcs was reliably different, $F(13,16) = 2.12$, $p < .05$. Despite the reliable main effect, post-hoc tests did not find any reliable differences between any two conditions.



Figure 25: Example of Thick Arcs that don't Intersect plus Baseline, levels 0 (ABCDEFGH), 18 (VCM9DNXS), and 36 (HPTX7YNX)

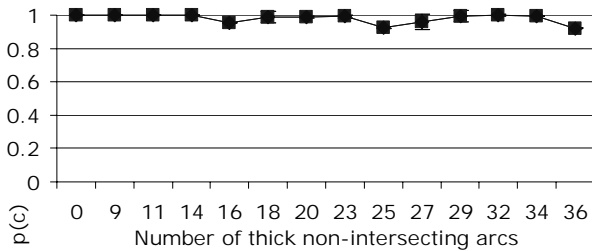


Figure 26: Accuracy rate for Thick Arcs that don't Intersect plus Baseline

DISCUSSION

We ran two user studies to understand where the human perceptual limits with regard to solving HIPs were for a variety of lone and combined parameters typically used to automatically generate HIPs that are difficult for a computer hacker to break. In the first study, it was hypothesized that humans would easily solve HIPs that only varied on one parameter of distortion, and in fact that is what the data revealed. For the parameter levels tested on plain, translated, rotated or scaled text HIPs, users were at 99% correct or higher. It would be difficult to imagine how to make these unidimensional HIPs difficult for users to solve, given today's typical web page size constraints, screen resolutions, and viewing distances. Study 1 also showed that for global warping, local warping and local warping plus a combination of other, distorting parameters, there is a significant decrease in human HIP solution accuracy. This was unexpected for all but the combined parameter HIP condition. It would appear that today's recognition algorithms and training methods outperform humans with sufficient levels of text warping. In effect, the human results from study 1 and the computer results reported in [2,12] indicate that if the positions of the characters are known (pure recognition task), there are no sweet spots for HIPs.

This prompted us to move to a segmentation task, in which additional clutter misleads computer attacks as to where the real characters are located. The second study examined the addition of arcs to the HIP designs and showed that human recognition performance was quite

Distortion (parameter range)	Computer accuracy range
Rotation (-45° to 45°)	0.00% - 0.00%
Global warp (120-360)	0.04% - 8.08%
Local warp (20-80)	0.01% - 3.51%
Local warp (20-80) + Baseline	0.01% - 5.23%
Thin arcs (9 - 45)	0.04% - 3.07%
Thick arcs (9 - 45)	0.27% - 34.04%
Thick non-intersecting arcs (9 - 45)	0.16% - 0.30%

Table 1. Preliminary computer accuracy results for single character recognition.

good for the baseline combination of parameters that had been used in Study 1 (but in that study local warping had been added). Accuracy was also quite high across all levels of HIP recognition with thin arcs in the foreground. Adding thin arcs to the baseline distortion increased error rates only at the highest difficulty levels significantly. Adding intersecting thick arcs, either with or without the baseline distortion, caused significant performance decrements, but non-intersecting thick arcs did not.

In summary, study 2 showed that adding clutter via thin arcs or non-intersecting thick arcs gives us two additional dimensions at which humans perform well. This is particularly significant because both of these dimensions can be used to design a segmentation-based HIP, knowing that segmentation is the Achilles' heel of machine-learning based computer attacks [2,12].

LESSONS LEARNED

The need to protect services from automatic script attack has created a market for HIPs. The design of good HIPs is turning out to be much more difficult than previously thought, pitting human against computer, in a dynamic and complex economic environment. This is the first set of studies on the human side of the equation.

Study 1 indicated that there are probably no sweet spots in which humans are significantly better than computers for HIPs where character location is easy to guess. Study 2 pointed to two directions for building segmentation-based HIPs, which are more likely to be harder for computers while remaining easily solvable by humans.

User data is necessary to drive HIP design in a direction that will not cause discomfort for most humans but will give hackers trouble for some time.

FUTURE WORK

The first user study examined individual distortions, while the second user study made a preliminary attempt at studying combinations of these features through the use of a baseline (obtained from the first study). Real-world

HIPs (see Introduction) employ more parameters and a much larger combination of these parameters. Examples include intersecting and non-intersecting arcs of different thicknesses, arcs in the background color, random meshes, background textures, etc. Some HIPs such as the Ticketmaster and Google HIPs, use phonetic generators to make the characters in these HIPs more readable to humans.

We do not consider the user time to respond to a HIP to be a major critical factor at this stage in our research. Participants took 10-15 seconds to respond to all of the HIPs reported here. Only the most difficult HIPs took an extra second or two longer than the easiest HIPs. In addition, the character misrecognitions are fascinating, but upon reflection, not very surprising. For example, the HIP for local warp, level 80 (Figure 11) has the stem of a K angled backwards. Only one participant correctly identified the warped letter while all other participants identified it as an X or unreadable. Still, we are currently focusing on character-based recognition in our next line of studies of this technology.

As part of another study, we conducted computer experiments with character distortions addressed in this paper. At this time, we have some preliminary results on single character recognition results. A brief summary of these results is presented in Table 1. As in the case of humans, computer accuracy is very high under rotation (Figure 6) and in the presence of non-intersecting arcs (Figure 23). Both had near zero percent errors. However, with character warp distortions (local and global) and intersecting arcs (thin and thick), we note that though computer and human accuracies were similar under very low distortion parameters, computer accuracies were significantly better under high distortion. Overall these preliminary results appear to indicate that for single character recognition, computers do much better than humans. Part of our future research is to study and compare human and computer recognition accuracies on single- and multi-character HIPs in greater detail.

Several lines of future research are possible that explore interactions between these parameters to produce a human friendly HIP. With time, computers will become incrementally better at computer vision and hackers will exploit these improvements. A good understanding of HIP design from a human perspective is inevitable in order to stay one step ahead of the hackers.

CONCLUSION

HIPs with thick foreground arcs are easily recognized at certain levels for humans, and yet these conditions remain extremely difficult for computer hackers to solve. The contribution of this research is to continue to drive our HIP design from a user-centered perspective, wherein we try to design for a “sweet spot” that maximizes the comfort of human solvers while minimizing the ease of the code being broken through machine learning.

REFERENCES

1. Simard PY, Szeliski R, Benaloh J, Couvreur J, and Calinov I (2003), “Using Character Recognition and Segmentation to Tell Computers from Humans,” *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE Computer Society, pp. 418-423, 2003.
2. Chellapilla K., and Simard P., “Using Machine Learning to Break Visual Human Interaction Proofs (HIPs),” *Advances in Neural Information Processing Systems 17*, Neural Information Processing Systems (NIPS’2004), MIT Press.
3. Turing AM (1950), “Computing Machinery and Intelligence,” *Mind*, vol. 59, no. 236, pp. 433-460.
4. Von Ahn L, Blum M, and Langford J. (2004) “Telling Computers and Humans Apart (Automatically) or How Lazy Cryptographers do AI.” *Comm. of the ACM*, 47(2):56-60.
5. *First Workshop on Human Interactive Proofs*, Palo Alto, CA, January 2002.
6. Von Ahn L, Blum M, and Langford J, *The Captcha Project*. <http://www.captcha.net>
7. Mori G, Malik J (2003), “Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA,” *Proceedings of the Computer Vision and Pattern Recognition (CVPR) Conference*, IEEE Computer Society, vol.1, pages:I-134 - I-141, June 18-20, 2003
8. Chew, M. and Baird, H. S. (2003), “BaffleText: a Human Interactive Proof,” *Proc., 10th IS&T/SPIE Document Recognition & Retrieval Conf.*, Santa Clara, CA, Jan. 22.
9. Simard, P.,Y., Steinkraus, D., Platt, J. (2003) “Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis,” *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE Computer Society, Los Alamitos, pp. 958-962, 2003.
10. Selfridge, O.G. (1959). Pandemonium: A paradigm for learning. In *Symposium in the mechanization of thought process* (pp.513-526). London: HM Stationery Office.
11. Pelli, D. G., Burns, C. W., Farrell, B., & Moore, D. C., “Identifying letters.” (accepted) *Vision Research*.
12. Goodman J. and Rounthwaite R., “Stopping Outgoing Spam,” *Proc. of the 5th ACM conf. on Electronic commerce*, New York, NY. 2004.