Usability of CAPTCHAs
Or usability issues in CAPTCHA design

Jeff Yan
School of Computing Science
Newcastle University, UK
Jeff.Yan@ncl.ac.uk

Ahmad Salah El Ahmad
School of Computing Science
Newcastle University, UK
Ahmad.Salah-El-Ahmad@ncl.ac.uk

ABSTRACT
CAPTCHA is now almost a standard security technology, and has found widespread application in commercial websites. Usability and robustness are two fundamental issues with CAPTCHA, and they often interconnect with each other. This paper discusses usability issues that should be considered and addressed in the design of CAPTCHAs. Some of these issues are intuitive, but some others have subtle implications for robustness (or security). A simple but novel framework for examining CAPTCHA usability is also proposed.

Categories and Subject Descriptors
D.4.6 Security and Protection, H.1.2 User/Machine Systems.

General Terms
Security, Human Factors, Design.

Keywords
CAPTCHA, security, usability.

1. INTRODUCTION
A CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) is a program that generates and grades tests that are human solvable, but beyond the capabilities of current computer programs [1]. This technology is now almost a standard security mechanism for addressing undesirable or malicious Internet bot programs (such as those spreading junk emails and grabbing thousands of free email accounts instantly) and has found widespread application on numerous commercial web sites including Google, Yahoo, and Microsoft's MSN.

It is widely accepted that a good CAPTCHA must be both robust and usable. The robustness of a CAPTCHA is its strength in resisting adversarial attacks, and this has attracted considerable attention in the research community (e.g. [12, 15, 16, 27]).

However, it is strikingly surprising that there has been little study of the usability aspects of CAPTCHA, although by definition, a CAPTCHA that is unusable for human should have no reason to exist. All related work known to us is as follows. A W3C Working Group report highlighted that CAPTCHAs can pose a major accessibility problem to "users who are blind, have low vision, or have a learning disability such as dyslexia", and discussed potential alternatives to human verifications [14]. However, it did not discuss how to improve the usability of CAPTCHAs. The only work concentrated on addressing the usability aspect of CAPTCHA design known to us [4, 5] recognised that CAPTCHA should be "human friendly", and it examined the impact of different text distortion techniques on the usability of a CAPTCHA designed by Microsoft. In addition, some usability issues of CAPTCHAs were touched in [3, 6, 7, 9].

In this paper, we aim to understand what kind of issues should be addressed to make CAPTCHAs usable in the contexts where this technology has been widely deployed. Solving issues of poor accessibility caused by CAPTCHAs, e.g. by exploring CAPTCHA alternatives, is important and of practical relevance, but beyond the scope of this paper.

Specifically, we will propose a novel framework for examining the usability of CAPTCHAs, and then under this framework, discuss issues that should be addressed in the design of a CAPTCHA to improve its usability. Many of the issues are novel – they are lessons that we have learnt both by breaking widely deployed CAPTCHAs and by designing our own. Some others were identified by peer researchers but scattered in the literature. This paper is a first attempt towards a systematic analysis of usability issues that should be considered and addressed in the design of robust and usable CAPTCHAs, although we do not claim the issues we have identified represent a complete list.

So far, there are the following three main types of CAPTCHAs:

- **Text-based schemes** – they typically rely on sophisticated distortion of text images rendering them unrecognisable to the state of the art of pattern recognition programs but recognisable to human eyes.
- **Sound-based schemes** (or *audio* schemes): - they typically require users to solve a speech recognition task.
- **Image-based schemes** - they typically require users to perform an image recognition task.

In this paper, our discussion will largely focus on text-based CAPTCHAs, for the following reasons.

First, text-based CAPTCHAs have been the most widely deployed schemes. Major web sites such as Google, Yahoo and Microsoft all have their own text-based CAPTCHAs deployed for years.

Second, text-based CAPTCHAs have many advantages compared to other types of schemes [4], for example, being intuitive to users world-wide (the user task performed being just character recognition), having few localization issues, and having good potential to provide strong security (e.g. the space a brute force attack has to search can be huge, if properly designed).

Symposium On Usable Privacy and Security (SOUPS) 2008, July 23-25, 2008, Pittsburgh, PA, USA.

Third, it can have a large and positive impact for the society to improve the usability of such popular and well-claimed CAPTCHAs by identifying issues that should be addressed in these schemes.

Lastly, although our discussions are focused on text-based schemes, they can also be relevant to other types of CAPTCHAs.

The rest of this paper is organised as follows. Section 2 presents our simple framework, which is inspired by text CAPTCHAs but applicable to other different types of schemes. Section 3 examines specific issues with text-based schemes using the framework. Section 4 briefly discusses usability issues with sound-based schemes under the same framework. Section 5 concludes the paper.

2. A SIMPLE FRAMEWORK

Quoted from Jakob Nielsen [13], usability is defined by the following five quality components: "

- *Learnability*: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- *Efficiency*: Once users have learned the design, how quickly can they perform tasks?
- *Memorability*: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- *Errors*: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- *Satisfaction*: How pleasant is it to use the design? "

Typically, the basic task that a CAPTCHA imposes to users is intuitive, easy to understand and easy to remember. Thus, CAPTCHA has a relatively good learnability and memorability. Therefore, in this paper, we will only consider the other three quality components.

The nature of CAPTCHAs determines that the following usability criteria are applicable to address efficiency, errors and satisfaction:

- *Accuracy*: how accurately can a user pass a CAPTCHA challenge? For example, how many times she has to try in order to pass a test?
- *Response time*: how long does it take for a user to pass the test?
- *Perceived difficulty/satisfaction of using a scheme*. How difficult to use do people perceive a CAPTCHA is? Are users subjectively satisfied and would they be willing to use such a scheme?

This set of criteria can be key for (quantitatively) evaluating the usability of CAPTCHAs. However, this set offers little specific guidance on *how* to improve accuracy, response time or perceived difficulty/satisfaction.

Instead, we propose the following three-dimensional framework for examining the usability of CAPTCHAs.

- *Distortion*. This dimension examines the form of distortions employed by a CAPTCHA and their impact on usability.
- *Content*. This dimension examines contents embedded in CAPTCHA challenges (or tests) and their impact on usability. For example, how should the content be organised, and is the content appropriate?
- *Presentation*. This dimension examines the way that CAPTCHA challenges are presented and its impact on usability.

With this framework, specific elements of a CAPTCHA can be pinpointed and improved so as to enhance the usability of the scheme as a whole.

This framework is applicable to text-based and sound-based CAPTCHAs, in which distortion, content and presentation typically are all concerned. It is also applicable to image-based schemes (e.g. IMAGINATION [24], PIX [1] and the scheme proposed in [26]). However, distortion is absent in some image-based schemes (e.g. Assira [25] and Bongo [1]) - for these schemes, only the dimensions of content and presentation matter.

3. USABILITY ISSUES OF TEXT-BASED CAPTCHAS

In this section, we discuss usability issues in text-based CAPTCHAs under the framework proposed in Section 2. Table 1 summarizes all the issues that will be discussed in the following sections.

Table 1. Usability issues with text-based CAPTCHAs

Category	Usability issue	
Distortion	Distortion method and level	
	Confusing characters	
	Friendly to foreigners?	
Content	Character set	
	String length	How long?
		Predictable or not?
	Random string or dictionary word?	
	Offensive word	
Presentation	Font type and size	
	Image size	
	Use of colour	
	Integration with web pages	

3.1 Distortion related issues

Distortion has a clear impact on the usability of CAPTCHAs, since human users would find it difficult or impossible to recognise over-distorted characters. To cope with usability problems caused by distortion, a system will have to allow multiple attempts for each user. Typically a new challenge is used for each attempt. This will not only annoy users, but also lowers the security of the system by a factor of the number of allowed attempts.

Distortion method and level. The most intuitive usability concern for a text-based CAPTCHA is its readability, which can be largely determined by what distortion methods are used and how much distortion is applied to texts. A Microsoft team [4] examined the following common distortion methods, among

others, and empirically determined the level of distortion for each method that will not make it difficult for human users to recognise distorted texts.

- *Translation*: moving characters either up or down and left or right by an amount
- *Rotation*: turning characters either in a clockwise or counter clockwise direction
- *Scaling*: stretching or compressing characters in the x-direction and y-direction
- *Warp*: elastic deformation of CAPTCHA images at different scales

This study led to valuable results, which guided the design of a Microsoft CAPTCHA that has been deployed for years in their online services such as MSN, Hotmail and Windows Live (We will refer to this CAPTCHA as the MSN scheme in this paper). These results are also applicable to the design of other text-based CAPTCHAs.

Confusing characters. Distortion often creates ambiguous characters, where users cannot be sure what they are. Although some characters have very different shapes, after distortion, they become hard to tell apart from each other. This problem is common in most schemes that we have studied. We list common confusing character pairs as follows.

- *Letter vs digits*: hard to tell distorted O from 0, 6 from G and b, 5 from S/s, 2 from Z/z, 1 from l.
- *Digit vs digit*: 5 is hard to tell apart from 6, 7 is written differently in different countries and often what looks like a 7 may in fact be a 1, and 8 can look like 6 or 9.
- *Letter vs letters*: Under some distortion, "vv" can resemble "w"; "cl" can resemble "d"; "nn" can could resemble "m"; "rn" can resemble "m" ; "rm" can resemble "nn"; "cm" can resemble "an". Table 2 shows some such confusing examples that we observed in the Google CAPTCHA (used for its Gmail service). We observed that about 6% of challenges generated by this Google scheme contained such characters.
- *Characters vs clutters*: In CAPTCHAs such as the MSN schemes, random arcs are introduced as clutters. Confusion between arcs and characters is often observed in this Microsoft scheme. For example, it is difficult to tell an arc from characters such as 'J', '7' and 'L' in Figure 1. In particular, the confusion between an arc and 'J' was observed regularly in this scheme (typically at the beginning or end of a challenge, more examples see Figure 1(d)).

Note: characters that look similar in one typeface can look differently in another typeface. So typeface is another related usability issue.

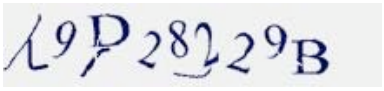
Friendly to foreigners? In theory, text-based CAPTCHAs are intuitive to world-wide users and have little localization issues – these were recognised by many researchers (e.g. [5]) as major advantages of text-based CAPTCHAs over other schemes. However, in a small scale test carried out with 20 students in the first author's class in October 2007, we observed that many foreign students whose mother tongue does not use the Latin alphabet performed much worse than those whose first language is based on Latin alphabet (e.g. native English speakers), when asked to recognise distorted challenges generated by BaffleText [6], an early text-based scheme. The former found it hard to recognise (or even guess) distorted letters in the scheme.

Table 2. Confusing characters in the Google CAPTCHA

Image	Confusing characters
	Is the middle part 'd" or connected "cl"?
	Another case of "cl" or "d" confusion.
	Another case of "cl" or "d" confusion.
	Is the starting part 'm' or connected 'rn"?
	The 2nd and the 3rd character could be confused with "w".
	A real headache: is the first part "m" or "rn", the middle part "inv" or "nw"?

(a)

(b)



(c)

(d)

Figure 1. Microsoft CAPTCHA: the 1st object in (a), (b) and (c) looks like 'J', '7' and 'L' respectively. The last object in each image in (d) looks like 'J'.

To the best of our knowledge, this is the first experiment examining the correlation between people's first languages and their performance in decoding distorted Latin alphabets in CAPTCHAs.

At the time of preparing the camera-ready version of the present paper, we became aware of a user study on the relevance of the language spoken by experiment participants to their speed of solving CAPTCHA [23]. In this study, it was observed that the average time for solving challenges generated by the Google CAPTCHA was similar for subjects familiar with English and those not familiar with English. This appears to contradict to our experimental result. However, this discrepancy can be easily explained: the CAPTCHA used in our study was much more distorted than the Google scheme.

On the other hand, our observation was (loosely) confirmed by Luis von Ahn in his world-wide deployed reCAPTCHA system [2]. He observed an average success rate of around 97% and 93% for passing reCAPTCHA tests in daytime and at night (both US time), respectively. According to IP addresses of service requests that reCAPTCHA has received, more users from outside of the US (e.g. those in Asia) access this service at night than in the daytime (both US time) – typically evening time in the US is daytime in Asia. This suggests to some extent that people with different first languages do perform differently in decoding distorted Roman characters. This is easy to explain - just imagine how easy it would be for someone (e.g. English) to decipher handwritten texts in a foreign language (e.g. Chinese).

The performance difference between foreigners and natives does not appear to be large in the case of reCAPTCHA. However, given the size of population using this service (hundreds of thousands websites serving millions of people at least, for example, popular sites such as Facebook and Twitter are amongst subscribers of this service), this "being friendly to foreigners" issue can be a serious usability concern. Moreover, for schemes whose designers were unaware of this issue, usability problems caused can be even worse.

3.2 Content related issues

The choice of content materials used in each CAPTCHA challenge can also have significant impact on usability.

Character set. The size of the character set used in a CAPTCHA matters for security. Typically, the larger the character set, the higher resistance to random guessing attacks each challenge can have. However, a larger character set can also imply a higher number of characters that look similar after distortion, causing confusion.

String length. The length of the text string used in each challenge also matters for security. If both the character set size and the string length are small, random guessing would have a high chance of passing the CAPTCHA. Typically, the longer the string is used in a challenge, the more secure is the result. For example, assume that the state of the art techniques can achieve an individual character recognition rate of r (<1), the chance of recognising the whole challenge of n characters can be r^n, which decreases as n grows.

String length has interesting usability implications. If random strings are used in a scheme, then the longer the string is, the more difficult the scheme is to use. The reason is that it is more demanding for users to decode and enter their answers correctly. For example, users might tend to make recognition mistakes, e.g. due to distorted characters looking like each other. However, it is not necessarily the case in schemes where English words are used. For example, it was observed for the reCAPTCHA scheme [18] that the longer the string is, the higher pass rate the users have [2]. A likely explanation is that the longer the word is, the more information people can gather, and thus Gestalt psychology (i.e., humans are good at inferring whole pictures from only partial information) effectively helps people to decode the word correctly. However, from short words that are too distorted to recognise, users would not be able to gather enough information to decode them correctly.

Whether the length of strings used in a scheme is predictable or not is another design issue. Some schemes choose to use a fixed length. For example, in the MSN scheme, each challenge uses 8 characters. In some other schemes such as Google's CAPTCHA, the string length is variable: each challenge uses a different number of characters, and the string length for each challenge is unpredictable. This design issue turns out to have implications for both security and usability.

For example, the use of a fixed string length in the MSN scheme has a negative impact on its security. The knowledge of how many characters can be expected in a challenge was used for locating connected characters and estimating the number of such characters in the challenge, which is a crucial step in our highly successful segmentation attack on the MSN scheme [16]. In this attack, our segmentation success rate was higher than 92%, which could lead to an overall (both segmentation and recognition) success rate of higher than 60%.However, on the other hand, such a design choice contributes to improving the scheme's usability. For example, knowledge of the string length can ensure that users know the first object in each challenge in Figure 1 (a)-(c) is a random arc, rather than a character 'J', '7' or 'L'. Therefore, the use of a predictable length of string, as well as an indication on how many characters a user is expected to enter (as shown in Figure 2), is good for usability.

Figure 2. The MSN scheme: the text length is fixed and indicated in the interface.

On the contrary, if the MSN scheme used a varied, unpredictable string length for each challenge, it would be much harder or even impossible for users to recognise that the above-mentioned objects are indeed arcs. With this disadvantage in usability, however, this design choice would make it much harder or even impossible to perform an automatic segmentation attack similar to ours [16].

The security of Google's CAPTCHA has not been rigorously tested yet. But we conjecture that its design choice of using unpredictable string length makes it harder to break this or achieve a high success rate, since length information can play an important role in segmenting a challenge image. Such a design choice has some usability concerns. For example, we have observed many confusing characters in this scheme, as discussed earlier (see Table 2). This kind of confusion would be eliminated or at least reduced if a user is informed of the number of characters in a challenge.

It appears that the following design can simultaneously achieve good security and usability in a CAPTCHA: using a variable length of strings in the scheme, and at the same time, for each challenge, the length information is distorted together with the string, and then embedded as part of the challenge. A detailed study of this design is our ongoing work.

Random string vs. dictionary words. Lexical information was exploited to attack CAPTCHAs (see, e.g. [12, 15]). However, we are not convinced that it is absolutely a bad idea to make use of lexical information in CAPTCHA schemes. Typically, schemes using dictionary words are more usable than those using random strings. For example, people typically type words faster than random strings. Moreover, it might be difficult for people to recognise individual characters that were distorted too much. But when these characters occurred as part of a word in a challenge, people who understand the language used could easily solve the challenge using the lexical context.

Rather, what really matters is how a CAPTCHA is designed. For example, if a scheme is so designed that its robustness is entirely dependent on the property of segmentation resistance, that is, its segmentation resistant mechanism would provide all the security it requires, we do not see any problem in using lexical information in CAPTCHAs. Nevertheless, we agree that the use of lexical information should be carefully examined.

For people who would like to be cautious, an alternative is to use a phonetic generator to create non-English but pronounceable character strings. This can make dictionary attacks more difficult, and provides better usability than purely random strings. But one potential weakness of this method is that people might tend to identify those strings as real English words [6].

Offensive words. Whether the content of the string used in each challenge is appropriate can affect user satisfaction, and thus is another usability issue. For example, it would be offensive to present a challenge showing words such as "negro". Offensive content can occur in either random or dictionary words based schemes. For example, offensive words occurred in both the Google CAPTCHA and reCAPTCHA [10]. A typical solution is to maintain a blacklist of *taboo words* to filter out inappropriate strings generated by a CAPTCHA. However, this is not a perfect solution for systems like reCAPTCHA, since some words used by such schemes are document chunks that cannot be recognised by OCR (Optical Character Recognition) software – that is, nobody knows what is in them in the first place.

3.3 Presentation related issues

The way that a CAPTCHA presents its challenges (or tests) has usability concerns. For example, font type and size used for characters matter [6, 7, 3], so does the size of challenge images. In this section, we discuss two other main issues in the text-based CAPTCHAs: 1) the use of colour in challenge images, and 2) the integration of these challenges with web pages.

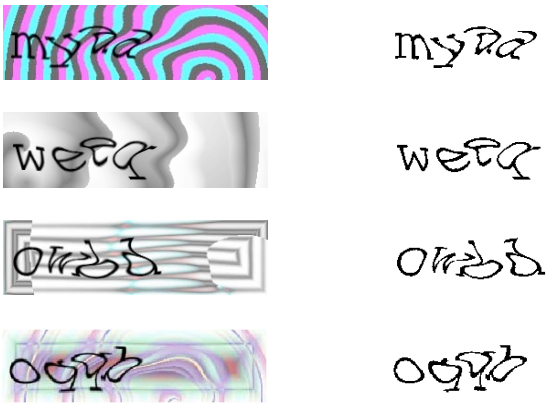
3.3.1 The use of colour

Colour is extensively used in user interfaces. When used properly, colour can much enhance user interface design [8]. Using colour has also been common in text-based CAPTCHAs, mainly for the following reasons.

- Colour is a strong attention-getting mechanism.
- Colour can provide variation to fit different user preferences [9].
- Colour is appealing and can make CAPTCHA challenges interesting.
- Colour can facilitate recognition, comprehension and positive affect.
- Colour can make CAPTCHA images compatible with the colour of web pages and make them look less intrusive [5].

In addition, colour schemes might also be expected to work as an additional defence against OCR software attacks in some schemes, since typically OCR software performs poorly in recognising texts in colour images – in particular, they do not do well in segmenting colour images.

However, we have seen many CAPTCHAs, in which **the use of colour is unhelpful for usability, has caused negative impact on security, or is problematic in terms of both usability and security**.



(a) (b)

Figure 2. Gimpy-r. (a) original challenges (b) text extracted by our automatic program (Note: images in (a) and (b) provide just the same level of security)

For example, Gimpy-r, a well-known early scheme designed at Carnegie Mellon University, used colourful images (see Figure 2 (a) for example challenges). However, the dominant colour of distorted texts in each challenge always had the lowest intensity amongst all colours used in the challenge, and this colour (often

black) never appeared in the background. This made it easy to extract the challenge text by a computer program - Figure 2 (b) shows the texts extracted by our automatic program.

The images in Figure 2 (a) and (b) show what the challenges look like for humans and computers respectively, and they provide just the same level of security. The colourful background was useless in terms of security – rather, its negative side effect is obvious: it confuses people and decreases the usability of the scheme.

The same problem also occurred in EZ-gimpy, another well-known early CAPTCHA designed at CMU (see Figure 3).

(a) (b)

Figure 3. EZ-gimpy. (a) original challenges (b) text extracted by our automatic program (Note: images in (a) and (b) provide just the same level of security)

To make challenge images look interesting, some CAPTCHAs generate images in which adjacent characters have distinct colours. The Cryptographp Captcha [19] is such a scheme, as shown in Figure 4 (a). However, this design feature turns out to be a misuse of colour, which leads to a fatal design mistake in terms of security as explained as follows.

(a) **(b)** **(c)**

Figure 4. Cryptographp CAPTCHA: (a) original images (b) after background noise removal (c) final segmented results

It is trivial to remove random shapes used as noise in the original challenges. For example, we used the following ad-hoc method. We scan an image pixel by pixel. If a pixel satisfies the following conditions, then it is removed:

1) The pixel colour is not the same as the dominant background colour (white);
2) At least 6 of its neighbour pixels have background colour.
3) Repeat the above 2 steps until no pixels satisfy condition 2.

Figure 4 (b) shows the results of applying such approach.

Typically, it is difficult to segment characters that overlap with each other. The state of the art of CAPTCHA design suggests that text-based schemes should rely on such segmentation resistance to provide security for CAPTCHA schemes [5]. However, since each character has a different (dominant) colour in this scheme, by picking up all pixels with the same colour, we effectively segment overlapped characters, as shown in Figure 4 (c)! We tested this method on 50 random challenges generated by the Cryptograph scheme, and achieved a success rate of 100% for segmentation (the average segmentation speed was about 60ms per challenge).

Breaking a CAPTCHA (in the sense of writing computer programs that automatically solve its challenges) typically involves a segmentation task and a recognition task, and it is trivial to apply standard techniques to recognise individual segmented characters with a high success. Therefore, this scheme is effectively broken. That is, the misuse of colour turned out to be a security disaster.

The similar mistake was also observed in FreeCap [20], another popular CAPTCHA (which has 22,800 hits in Google search, and been widely used in website including popular gaming sites such as [21]. In this scheme (see Figure 5), the feature that adjacent letters have different colours aids to segment touching and overlapping characters, which would be otherwise much harder to segment.

Figure 5. FreeCap CAPTCHA samples

BotBlock [17] is a good example showing that the misuse of colour in a CAPTCHA can cause both usability and security problems. As showed in Figure 6 (a), random letters are used in this scheme, and they appear in different places in a challenge. A sophisticated colour management method is introduced: backgrounds were of multiple colour blocks of random shapes, and foreground colours also occurred in the background.

However, this fancy colour scheme often made it hard for people with normal vision to recognise challenge texts.

On the other hand, this scheme relied too much on the colour scheme to provide security – we tested 100 samples of this scheme, and they were indeed all resistant to the best OCR program on the market. Unfortunately, a fatal design error made it trivial to get rid of all the fancy background: there is an exploitable colour pattern for foreground texts – the same colour occurs repetitively. By looking for that pattern, we successfully extracted the challenge text in all samples we tested. That is to say, the robustness of this scheme is just equivalent to that for the challenges showed in Figure 6 (b) – it's trivial to decode the latter.

(a) (b)

Figure 6. BotBlock CAPTCHA (a) sample challenges (b) challenge text extracted by our automatic program. Note: images in (a) and (b) provide just the same level of security.

As a matter of fact, this scheme is vulnerable to the "pixel count" attack we discovered in [15]. That is, by counting the number of foreground pixels, we could recognise most of the characters. For a few characters with identical pixel counts, a simple analysis of their geometrical shapes worked well to tell them apart. Such

simple attacks have successfully broken all of the 100 random samples we tested.

The lesson we have learned is that the use of colour in CAPTCHA can be tricky – it is more than a mere usability issue, because of its potential impact on security. In the mean time, we observed that the schemes deployed by major websites now do not use fancy colour schemes. For example:

- The MSN scheme: a simple colour scheme is used, where foreground (i.e. challenge text) is dark blue and background light gray.
- Google: all texts use a single colour (green, red or blue), and a white background.
- Yahoo and reCAPTCHA: simply uses black and white only (it might be worthwhile to mention that the main designer of reCAPTCHA designed Gimpy-r and EZ-gimpy, which Yahoo adopted for their websites a few years ago).

It seems that the "Las Vegas effect" on the use of colour in interface design - no colour might be better than too much colour – holds for text-based CAPTCHAs, and in this context, it is not only a usability principle but also a security lesson. Therefore, we have the following recommendations:

- When you are not sure (for example, if you are not an expert in both human vision and image processing), use two colours in your scheme with one for background and the other for foreground, for the sake of both security and usability. The simplest safe choice can be just to use black/white.
- Using fancy colour schemes might not only introduce usability problems - for example, it is far more difficult than it appears to tell what kind of colour images would cause problems for colour-blind people, given the number of different types of colour-blindness - but also fail to provide any resistance to attacks that aim to break your scheme.
- It is not necessarily impossible to use colour to enhance both security and usability of a CAPTCHA. Rather, this is an interesting and worthwhile open problem.
- For now, we would rather rely on segmentation resistance to provide security in a CAPTCHA scheme.

3.3.2 Integration with a web page/form

The integration of CAPTCHA challenges with web pages can also have usability concerns. For example, until very recently, the "type the two words" box in the popular reCAPTCHA scheme was not automatically enabled (see Figure 7). So users had often input their answers to nowhere, unless they manually activated the box in advance. But it certainly increased the users' burden by forcing them to enable the text box before they could enter an answer. To avoid annoying end users, a CAPTCHA should be integrated into a web page with care to minimize their burden.

Figure 7. reCAPTCHA user interface

4. APPLICABILITY TO AUDIO CAPTCHAS

In this section, we briefly show that the usability framework we have discussed earlier is also applicable to other type of CAPTCHAs. We use audio schemes, the second most widely used CAPTCHAs, as an example.

In audio CAPTCHAs, letters are read aloud instead of being displayed in an image. Typically, noises are deliberately added to prevent such audio schemes from being broken by current speech recognition technologies. For example, the audio version of reCAPTCHA uses as noise sound clips in the native Navajo language, which only a very limited number of people in the world understand – this natural language was used in the Second World War as an unbreakable radio cipher for the same reason.

Distortion. Background noises effectively distort sounds in audio CAPTCHAs. There is no rigorous study of what kind of background noises will introduce acceptable sound distortion. However, it is clear that distortion methods and levels, just as in text based CAPTCHAs, can have a significant impact on the usability of audio CAPTCHAs. For example, an early test in 2003 showed that the distorted sound in an audio CAPTCHA that was deployed at Microsoft's Hotmail service was unintelligible to all (four) journalists, with good hearing, that were tested [11].

Due to sound distortion, confusing characters can also occur in audio CAPTCHAs. For example, we observed that it is hard to tell apart 'p' and 'b'; 'g' and 'j', and 'a' and '8'.

Whether a scheme is friendly to non-native speakers is another usability concern for audio CAPTCHAs. For example, both authors of this paper are non-native English speakers with good hearing, but we found that the audio version of Google CAPTCHA, Microsoft's MSN CAPTCHA and reCAPTCHA are entirely unusable for us.

Content. Content materials used in audio CAPTCHAs are typically language specific. Digits and letters read in a language are often not understandable to people who do not speak the language. Therefore, unlike text-based schemes, localisation is a major issue that audio CAPTCHAs face.

Furthermore, issues in the content category discussed in Section 3 are all in principle applicable to audio CAPTCHAs.

Presentation. The use of colour is not an issue for audio CAPTCHAs, but the integration with web pages is still a concern.

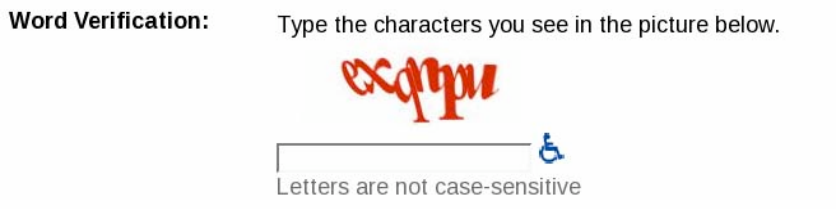


Figure 8. Interface of Google CAPTCHA

For example, there is no standard graphical symbol for representing an audio CAPTCHA on a web page. Although many schemes such as Microsoft and reCAPTCHA use a speaker symbol (see Figures 2 and 7), but Google uses a "disability" symbol (see Figure 8). It seems to us that a speaker symbol is more straightforward metaphor.

More importantly, what really matters for visually impaired users is that the html image alternative text attached to any of the above symbol should clearly indicate the need to solve an audio CAPTCHA.

When embedded in web pages, audio CAPTCHAs can also cause compatibility issues. For example, many such schemes require JavaScript to be enabled. However, some users might prefer to disable JavaScript in their browsers. Some other schemes can be even worse. For example, we found that one audio scheme requires Adobe Flash support [22]. With this scheme, vision-impaired users will not even notice that such a CAPTCHA challenge exist in the page, unless Flash is installed in their computers - apparently, no text alternative is attached to the speaker-like Flash object, either.

To summarize this section, we have the following. The three-dimensional usability framework, along with many issues identified for text-based schemes, are applicable to audio CAPTCHAs. In the mean time, audio schemes also face some new usability issues that do not exist in text-based schemes.

5. CONCLUSION

We have discussed usability aspects of CAPTCHA design, and the main contributions of this paper include the following.

First, we have systematically examined usability issues that should be addressed in the design of text-based CAPTCHAs, the most popular type of such schemes. In particular, for the first time, we have observed the following issues:

- Contrary to the common belief, text-based CAPTCHAs can be difficult for foreigners.
- Whether the length of strings used in a scheme is predictable or not can have interesting implications for both its security and usability.
- The use of colour in a CAPTCHA can have an impact on its usability, security or both.

All this contributes to further our understanding of the design of usable and secure CAPTCHAs, for which current collective knowledge is limited.

Second, we have proposed a simple but novel framework for examining usability issues of CAPTCHAs, and showed that this framework is applicable not only to text-based schemes, but also to other types of CAPTCHAs. We do not claim the list of usability issues we have discussed is complete, and encourage researchers to identify more of them using our framework. In particular, a lot more can be explored for sound-based and image-based CAPTCHAs, which is our future work.

Overall, the design of CAPTCHA is still an art, rather than a science. It requires considerable study to evolve the design of secure and usable CAPTCHAs into a science.

6. ACKNOWLEDGMENTS

We are grateful to Lindsay Marshall and Chris Kray for proofreading this paper and many valuable comments, and to anonymous reviewers for helpful comments and bringing [23] to our attention.

7. REFERENCES

[1] L von Ahn, M Blum and J Langford. "Telling Humans and Computer Apart Automatically", CACM, V47, No2, 2004.

[2] Luis von Ahn, Personal Communications, Oct 2007.

[3] HS Baird, MA Moll and SY Wang. "A highly legible captcha that resists segmentation attacks". Proc. of Second Int'l Workshop on Human Interactive Proofs (HIP'05), ed. by HS Baird and DP Lopresti, Springer-Verlag. LNCS 3517, Bethlehem, PA, USA, 2005.

[4] K Chellapilla, K Larson, P Simard and M Czerwinski, "Designing human friendly human interaction proofs", ACM CHI'05, 2005.

[5] K Chellapilla, K Larson, P Simard and M Czerwinski, "Building Segmentation Based Human-friendly Human Interaction Proofs", 2nd Int'l Workshop on Human Interaction Proofs, Springer-Verlag, LNCS 3517, 2005.

[6] M Chew and HS Baird. "BaffleText: a human interactive proof". Proc. of 10th IS&T/SPIE Document Recognition & Retrieval Conference, 2003.

[7] AL Coates, H S Baird and RJ Fateman. "PessimalPrint: A Reverse Turing Test", Int'l. J. on Document Analysis & Recognition, Vol. 5, pp. 158-163, 2003.

[8] Lindsay W. MacDonald. "Using Colour Effectively in Computer Graphics". IEEE Computer Graphics and Applications, July/August 1999.

[9] T Converse, "CAPTCHA generation as a web service", Proc. of Second Int'l Workshop on Human Interactive Proofs (HIP'05), ed. by HS Baird and DP Lopresti, Springer-Verlag. LNCS 3517, Bethlehem, PA, USA, 2005. pp. 82-96

[10] Dan Goodin, "Facebook takes the Captcha rap". The Register, Dec 13, 2007. Available at

http://www.theregister.co.uk/2007/12/13/facebook_captcha_goes_wrong/

[11] Paul Festa. “Spam-bot tests flunk the blind”, CNET News.com, July 2, 2003. Available at http://www.news.com/2100-1032-1022814.html.

[12] Greg Mori and Jitendra Malik. “Recognising Objects in Adversarial Clutter: Breaking a Visual CAPTCHA”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03), Vol 1, June 2003, pp.134-141.

[13] Jakob Nielsen. Usability 101: Introduction to Usability, 2003. Available at http://www.useit.com/alertbox/20030825.html.

[14] W3C Working Group, “Inaccessibility of CAPTCHA - Alternatives to Visual Turing Tests on the Web”, Nov, 2005. Available at http://www.w3.org/TR/turingtest/.

[15] J Yan and A S El Ahmad. “Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms”, in *Proc. of the 23rd Annual Computer Security Applications Conference (ACSAC’07)*. FL, USA, Dec 2007. IEEE computer society. pp 279-291.

[16] J Yan and A S El Ahmad. “A Low-cost Attack on a Microsoft CAPTCHA”, School of Computing Science Technical Report, Newcastle University, England. Feb, 2008.

[17] BotBlock. http://www.chimetv.com/tv/products/botblock.shtml. Accessed in Feb, 2008.

[18] http://recaptcha.net/

[19] http://www.cryptographp.com

[20] FreeCap. http://www.puremango.co.uk/cm_php_captcha_script_113.php

[21] Diablo 2 event, http://newd2event.net/index.php?id=hacks/redvex/HotPlug_Plugin

[22] The “Shout it out” audio CAPTCHA, http://www.nswardh.com/shout/. Accessed in Feb, 2008.

[23] R Chow, P Gollé, M Jakobsson, X Wang, L Wang. “Making CAPTCHAs clickable”. Ninth Workshop on Mobile Computing Systems and Applications (HotMobile 2008). 2008 February 25-26; Napa, CA.

[24] Ritendra Datta, Jia Li and James Z. Wang, “IMAGINATION: A Robust Image-based CAPTCHA Generation System”, Proceedings of the ACM Multimedia Conference, pp. 331-334, Singapore, ACM, November 2005.

[25] J Elson, JR Douceur, J Howell and J Saul. “Asirra: a CAPTCHA that exploits interest-aligned manual image categorization”. Proceedings of the 14th ACM conference on Computer and communications security (CCS), 2007.

[26] M Hoque, D Russomanno, M Yeasin. “2D Captchas from 3D Models”, IEEE SoutheastCon 2006 Memphis, TN, April 2006.

[27] J Yan and A S El Ahmad. “Is cheap labour behind the scene? - Low-cost automated attacks on Yahoo CAPTCHAs”, School of Computing Science Technical Report, Newcastle University, England. Apr, 2008.