

# State of the Art in Anti-Automation Technologies in Today's Web

Stefan Panhuber, Markus Haudum

**Abstract.** In this Paper the Concept of CAPTCHA's (Completely Automatic Public Turing test to tell Computers and Humans Apart) and why they are needed in today's Web will be introduced first. The start will be made with the first CAPTCHA's that were used by AltaVista and Yahoo. Then all of the different CAPTCHA's that are State of the Art right now will follow. The next part is about explaining the various CAPTCHA's, including the details of creating them and pointing out the most important attributes to be aware when creating CAPTCHA's.

## 1 Introduction

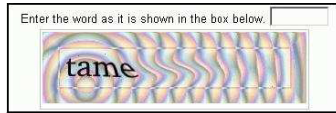
It was first in 1997, when the Problem of Bots considering Web-Services came up. The proliferation of publicly available services on the Internet has invited abuses by programs ('bots', 'spiders') designed to steal services and conduct fraudulent transactions [1].

AltaVista had the problem that people wrote Bots to resubmit their URL's over and over again to achieve a better ranking, which lead to the implementation of a new technology by them. Yahoo and PayPal came up with a similar technology in 2000 which lead to a cooperation between them. By January 2002 the subject was called 'human interactive proofs' (HIPs), defined broadly as challenge/response protocols which allow a human to authenticate herself as a member of a given group: e.g. human (vs. machine), herself (vs. anyone else), etc [2].

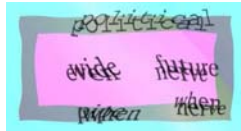
The HIPs worked by using the disability of current OCS systems to read text that could be read by humans but not by Bots. This was the time when the first CAPTCHAs were born. A Captcha is a **C**ompletely **A**utomatic **P**ublic **T**uring **T**est to tell **C**omputers and **H**umans **A**part.

The first used Concept of a Captcha was the Gimpy, the basic idea was to take a word, insert it into an image and manipulate the image so bots can't read it but humans still can. The original Gimpy showed seven words where three had to be recognized by the user. Later, the EZ-Gimpy was introduced which consisted only of one word to be recognized. The EZ-Gimpy is mostly used in today's Web-Applications.

Another important point is that the Source Code of the algorithms to create a Captcha is available to the public and still Captcha's should resist machine attacks for years. Today there exist a lot of different Captcha's who use different methods to achieve their goals.



**Fig. 1.** An EZ-Gimpy CAPTCHA in use at Yahoo [3]



**Fig. 2.** A Gimpy CAPTCHA. The task is to list 3 different words from the image [3].

## 2 Today's Variations of CAPTCHA's

Today, there exist a large amount of different Captcha approaches. This section will explain the Concept behind these Captcha's while in later Chapters it will be shown how efficient this Captcha's are against machine attacks.

### 2.1 Gimpy, EZ-Gimpy

As already explained in the Introduction chapter, the idea of this Captcha's is to take a word, lay it over an image and manipulate the image so only humans can read it. A good Gimpy Captcha needs to take care of the background. The background needs to be changed frequently, the line thickness of the words and the background should match, the background should break the outlines of to be recognized word and at last, the background must not consist of an easy to recognize pattern.

When it comes to image quality reduction, there are two very common strategies which are "BaffleText" and "Scatter Type".



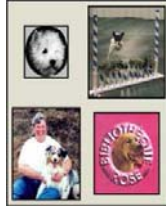
**Fig. 3.** BaffleText is a reading-based CAPTCHA ... that uses random masking to degrade images of non-English pronounceable character strings. [1]



**Fig. 4.** Scatter Type: Its challenges are pseudo randomly synthesized images of text strings rendered in machine-print typefaces: within each image, characters are fragmented using horizontal and vertical cuts, and the fragments are scattered by vertical and horizontal displacement. [4]

## 2.2 Pix

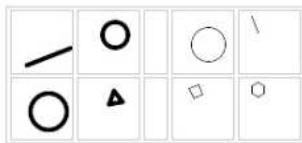
PIX is a Program with a big database containing labeled images. Every image represents a specific object, like a table, a house or a tree. The Program chooses an Object and four corresponding images automatically and shows them to the user. The users have to answer which Object is represented.



**Fig. 5.** Example of a PIX. [1] The answer is “dog”.

## 2.3 Bongo

The user gets two sets of images. The sets differ in one specific attribute which must be recognized by the user to pass the Test. Attributes are the thickness of the lines, the forms of the shapes, the amount of images or something different. Once the user has detected the specific attribute he has to add some images to one of the two sets so that the pattern is still consistent.



**Fig. 6.** Example of a Bongo [1]

## 2.4 Sounds

The Sound Captcha works similar to the Gimpy method. When taking the test, the user is presented a modified sound which makes it hard for machines to know what the word was but still easy enough for humans to get it right. This Captcha is often used with Gimpy's to make it possible for people to take the test, even if they can't see.

## 2.5 Formulas

Formula Captcha's are often needed when it is not only important to tell humans and computers apart but also to tell children and grownups apart. The idea is very simple; a formula is presented to the user. The Test is passed when the user solved the formula right.

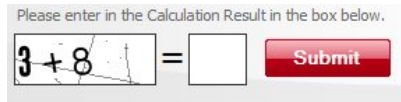


Fig. 7. A Formula Captcha taken from <http://www.linkvendor.com/seo-tools/url-rewrite.html>

## 2.6 Passfaces

Passfaces are based on the human ability to recognize human faces very quickly and efficient. The user first has to remember three or more Pictures which he has to recognize later. When the test starts, the user gets shown a 3x3 Matrix containing one of the Passfaces and 8 other random images for every PassFace.

Assuming there are 3 to 7 Passfaces using a 3x3 Matrix per PassFace, there are  $9^3=729$  and  $9^7=478269$  possible combinations. [5]



Fig. 8. An example for a 3x3 Matrix for Passfaces [5]

## 3 Creating CAPTCHA's

### 3.1 Requirements for creating CAPTCHA's

"Machines won't stay stupid forever. Range of problems they can solve is growing reasonably rapidly – it certainly isn't shrinking." [6]

This statement contains the two top essential criteria's for creating an effective CAPTCHA. The first aspect is the effectiveness of keeping out machines. The second one can be concluded in a single sentence which is: Is the test tolerable for humans or is it too complex. "While today there are a large number of generative CAPTCHA's to choose from, someday we may run out of tests that meet both criteria (hard for machines, tolerable to humans). Should we be concerned?" [6]

The simple answer is no, because there exists a nearly endless variety of tasks in the real World. [6]

3.2 Creating visual text based CAPTCHA's


3.2.1 Error sources for visual text Recognizers

There are four main error sources for visual text Recognizers. The first one is image quality, containing Background noise, printing surface and writing styles. The second one is image features, containing variable stroke width, slope, rotations, stretching, compressing. The third one is segmentation errors, containing Over-segmentation, merging, fragmentation, ligatures, and scrawls. The fourth and last one is the recognition of errors, containing "Confusion with similar or too large lexicon entries." [7]

3.2.2 Gestalt Laws

Another very important aspect of creating a CAPTCHA, are the three Gestalt Laws which are "Gestalt psychology is based on the observation that we often experience things that are not a part of our simple sensations "[7], followed by "What we are seeing is an effect of the whole event, not contained in the sum of the parts (holistic approach)" [7] ending with "By no means restricted to perception only (e.g. memory) "[7]

3.2.2.1 Law of closure:



3.2.2.2 Law of similarity:

OXX  
XOX  
XXO

Fig. 9. Law of closure [7]

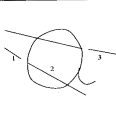
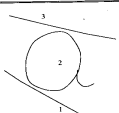
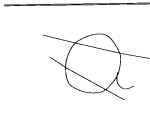
3.2.2.3 Law of proximity:

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

3.2.2.4 Law of symmetry:

[ ][ ][ ]

3.2.2.5 Law of continuity:



(a)

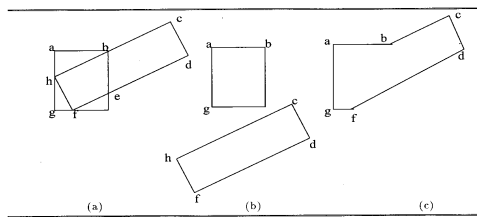
(b)

(c)

**Fig. 10.** Example picture for “Law of continuity” [7]

A uses the concept of Ambiguous segmentation while B is based on good continuity, following the path of minimal curvature change. Finally, C is using a concept called “Perceptually implausible segmentation” [7]

### 3.2.2.6 Law of familiarity:



**Fig. 11.** Example picture for “Law of familiarity”[7]

In An ambiguous segmentation is shown, while in B perceptual segmentation is used and in C the concept is based on the following idea: “Segmentation based on good continuity proves to be erroneous” [7]

### 3.2.2.7 Figure and ground:



**Fig. 12.** Example Picture for “Figure and ground”. [7]  
This Picture shows two faces or a vase.

### 3.2.2.8 Memory:



**Fig. 13.** Example Picture for “Memory”. [7]

### 3.2.3 Examples:

Here are some Example CAPTCHA’s by using “Gestalt Laws”:



Fig. 14. Example Picture for using “Gestalt Law” closure and proximity [7]

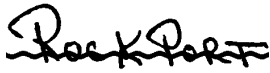


Fig. 15. Ex. Picture for using “Gestalt Law” continuity and figure and ground [7]



Fig. 16. Example Picture for using “Gestalt Law” memory and internal metrics [7]

### 3.3 Creating acoustic CAPTCHA's:

#### 3.3.1 Important Rules for creating acoustic CAPTCHA's

For creating an effective acoustic CAPTCHA there are three rules. The first rule is, “Making the dialogue difficult for machines (high level: syntactic or semantic)”. [8] The second one is “Making the speech signal difficult for machines (low level: recognizing phenomenon)”. [8] The last one is “Processing the speech to create aura illusions that affect humans but not machines.” [8]

#### 3.3.2 Problems for using acoustic CAPTCHA's

There exist three different Problems, that result in the hardly use of acoustic CAPTCHA's in today's web which are as followed.

*“No keyboard is available for inputting the result.”[8]*

This problem most occurs on public internet access stations, but some of these tests have some prepared buttons with different answers, so there is sometimes no need for a keyboard. Another solution for this problem is to ask the User to repeat the answer by speaking into a microphone.

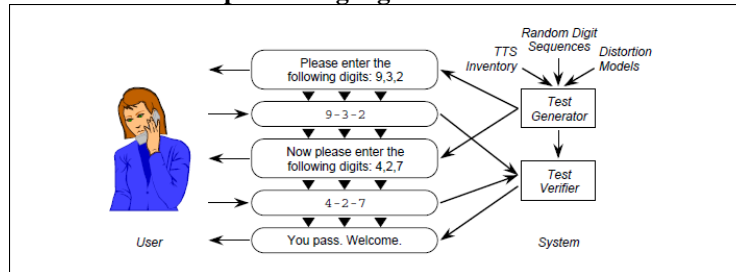
*“Similar sounding words (homonyms) can have different spellings.”[8]*

There is no other solution than to keep such words out of CAPTCHA's.

*“Most people are poor spellers.”[8]*

This is the main reason why acoustic CAPTCHA's are hardly used in today's web.

### 3.3.3 Protocol for spoken language Human Interactive Proof



**Fig. 17.** Example for a Protocol [8]. The interaction of an acoustic CAPTCHA with a human user is shown here.

### 3.3.4 Cleverer solutions for acoustic CAPTCHA's

There exist some good solutions that differ from the usual repeating word or number systems like "What number comes after (or before)  $m$ ? Where  $m$  is a modest-size integer, e.g.  $0 < m < 1000$ ." [8] Or "Which day of the week comes after (or before)  $X$ ?" where  $X$  signifies one of the seven days of the week. [8]

## References

- [1] H.S. Baird and M. Luk. Protecting Websites with Reading-based CAPTCHAs. In Proc. 2nd Int. Web Document Analysis Workshop, pages 53–56, Edinburgh, Scotland, 2003.
- [2] Complex Image Recognition and Web Security, in M. Basu & T. K. Ho (Eds), Data Complexity in Pattern Recognition, Springer-Verlag London Ltd, 2006.
- [3] G. Mori and J. Malik, "Breaking a visual CAPTCHA," 2002. In submission to Computer Vision and Pattern Recognition 2003.
- [4] H. S. Baird and T. Riopka, "ScatterType: a Reading CAPTCHA Resistant to Segmentation Attack," Proc., IS&T/SPIE Document Recognition and Retrieval Conf, San Jose, CA, January 16{20, 2005 [in the present proceedings].
- [5] Christine Pape: Alternative Authentifizierungsverfahren: Passfaces und CAPTCHAs. Konferenzseminar "Verlässliche Verteilte Systeme", Wintersemester 2005/2006 vom Lehr- und Forschungsgebiet Informatik 4 der RWTH Aachen.
- [6] Daniel Lopresti: Leveraging the CAPTCHA Problem. Proceedings of the Second HIP Conference, 2005.
- [7] A. Rusu and V. Govindaraju. Visual captcha with handwritten image analysis. Proc. of 2nd International Workshop on Human Interactive Proofs, LNCS 3517:42–52, 2005.
- [8] Lopresti, D., Shih, C., & Kochanski, G. (2001). Human Interactive Proofs for spoken language interfaces. Proceedings of the First HIP Conference, 2002.