

CAPTCHA Security Engine

Abstract

A CAPTCHA is a type of challenge-response test used in computing to determine whether the user is human or not. "CAPTCHA" is a contrived acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart".

A CAPTCHA involves one computer (a server) which asks a user to complete a test. While the computer is able to generate and grade the test, it is not able to solve the test on its own. Because computers are unable to solve the CAPTCHA, any user entering a correct solution is presumed to be human.

A CAPTCHA is sometimes described as a reverse Turing test, because it is administered by a machine and targeted to a human, in contrast to the standard Turing test that is typically administered by a human and targeted to a machine. For example, human can generally read degraded images, but OCR machines cannot. CAPTCHAs are designed to prevent bots – programs that pose as humans on the Internet – from abusing internet services. Bots, driven not to dominate but to sell, sign up for thousands of free email accounts every minute, sending millions of spam messages from them

Existing System

The existing system consists of the typing test given to help determine if the person typing is really a human or some bot. It can be an anti-spam device. You must prove yourself a true human posting some message or recommending URL rather than a computer program spamming the universe. The test typically requires you to type in some warped series of letters.

Proposed System

The proposed system consists of a CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot.

Advantages over Existing System

- *background colors*
- *background gradient fill colors*
- *fonts (2 default and limited to the amount of font's on your system)*
- *font color*
- *random character generator (characters can be configured)*
- *optional border around the CAPTCHA*
- *border color (defaults to black)*
- *border thinness (defaults to one)*

Scope of the System

The proposed system scope is limited, in such away Captchas displayed only with text. May be in future we will enhance our project captcha can permit users to opt for an audio or sound Captcha.

And also CAPTCHAs can be displayed video.

Module Description

Number of Modules:

- 1) Generation of Image on the fly (Dynamic)*
- 2) Base64 Encoder/Decoder*
- 3) Image Comparison*

1) GENERATION OF IMAGE ON THE FLY(Dynamic) :

1. Property Configuration:

The properties of an image are number of charaters,background color, foreground color, width of an image, height of an image, number of pixels, number of lines, number of boxes.

2. Image Probability:

Images displayed with the different probabilities of an image are shown below.



FIG 1



FIG 2

3. Image Creation:

Generally to create an image we use bmp formats, gif formats, wmf formats but here we used jpeg format which have enough advantage than these formats such as loss less compression. here there are steps first one is the jpeg encoder second one is creation of file to store an image. They are shown below.

- *JPEG Encoder*
- *File Creation*

2) BASE64 ENCODER/DECODER:

Encodes binary data into Base64 representation. Decodes Base64 form into Binary data. In computing, base64 is a data encoding scheme whereby binary-encoded data is converted to printable ASCII characters. It is defined as a MIME content transfer encoding for use in internet e-mail. The only characters

used are the upper- and lower-case Roman alphabet characters (A-Z, a-z), the numerals (0-9), and the "+" and "/" symbols, with the "=" symbol as a special suffix code. Full specifications for base64 are contained in RFC 1421 and RFC 2045. The scheme is defined only for data whose original length is a multiple of 8 bits, a requirement met by most computer file formats.

3) IMAGE COMPARISON:

Captcha compares the typed text with existing text, which is hided in the header of the image. When the user types text in text field, then the Base64 will decode text internally.

Then Comparison is done between the typed text and the actual text, which is present in header.

Features to be implemented

- ***Maintainability***
- ***Easy deployment with Ant script.***
- ***Exception handling***
- ***Client-side validations***

Technologies to be used

- ***Web Presentation: HTML, CSS***

- **Client – side Scripting: JavaScript**
- **Programming Language: Java**
- **Web based Technologies: Servlets, JSP**
- **Build Tool: ANT**
- **Debug Tool: Log 4J**
- **CASE tool: Rational Rose, Visual Paradigm, Enterprise Architect**
- **Operating System: Windows XP/2000/2003, LINUX, Solaris**
- **J2EE Web/Application Server: Tomcat/Weblogic/WebSphere/JBoss/Glass Fish**
- **IDEs: Eclipse with My Eclipse plug-ins/Net Beans/RAD**
- **Browser: IE/Mozilla**

Hardware requirements

- | | | |
|---------------------|-------|------------------|
| • Pentium processor | ----- | 233 MHZ or above |
| • RAM Capacity | ----- | 256MB |
| • Hard Disk | ----- | 40GB |
| • Floppy disk | ----- | 1.44 MB |
| • CD-ROM Drive | ----- | 32 HZ |
| • KEYBOARD | ----- | 108 Standard |

