

# Developing a CAPTCHA Utilizing Cognitive Ability of Human through PHP

**Dr. Vijay Dhaka**

Professor (Dept. of Computer Science)  
Jaipur National University  
vijaypal.dhaka@gmail.com

**Mrs. Geeta Gandhi**

Research Scholar  
Jaipur National University  
gandhigeeta3@gmail.com

## -----ABSTRACT-----

*The internet has evolved from being an academic pursuit to a huge commercial commodity. The IP thin waist associated with the simplicity of the present design has been a remarkable architectural choice, motivated by the need to converge multiple link layer technologies and end-to-end transport mechanisms. Undoubtedly, this increased usage of internet and network technology has given a drastic change in online communication and web usage leading to the requirement to focus on web security.*

*One of the most common security tools is CAPTCHA which is utilized to block bots or automated script execution. The present paper focuses on the need of CAPTCHA which can maintain a balance between usability and security. For this, the paper introduces the generation of CAPTCHA codes utilizing the cognitive ability of human and then developing these codes randomly on the screen using the features of server side scripting language, PHP.*

## I. Introduction:-

The increased usage of Internet and Network technology has changed the focus in accessing computer environments [1]. Only ten years ago, it was hard to imagine that computers today would be engaged in automated web navigation and requests for contents and resources [2].

Now web applications like email, social networking sites, blogs, e-governance sites etc. has become every body's need [3].

Thus, this rapid development of information technology has increased the urgency to identify security approaches [4]. The pervasive availability and wide usage of wireless networks with different kind of topologies, techniques and protocol suites have brought with them the need to improve security mechanisms. So, we require a methodology to provide security to web applications. This design development and evaluation of security techniques must begin with a thorough analysis of the requirements and a deeper understanding of the approaches that are practical within the system constraints.

One of the most common reminders of this type of abuse is the so called CAPTCHA, short for Completely Automated Public Turing Test to Tell Computers and Humans Apart.

Many websites uses CAPTCHA to block bots or automated script execution [5]. Thus, CAPTCHA is a class of HIP who's most common usage includes:

- i. Signing up for a free email account (Google, Hotmail, Yahoo and others).
- ii. Accessing some online resources such as buying tickets at ticketmasters.com or executing a whole query at register.com
- iii. For protection against denial of service attacks.

More generally, it is a HIP consists of a broad class of challenge- response protocol which allows an unaided human to authenticate herself as a member of a given group, such as humans or adults. Such proofs must resist passive attacks: any party that sees the proof should be unable to falsely generate a proof of membership [6].

Therefore, CAPTCHA plays a very critical role for providing security to these web services. But, the CAPTCHA available nowadays is becoming prone to machine learning attacks with the increase in Artificial Intelligence which further leads to the requirement of the development of new CAPTCHA in such a way that it could maintain a balance between usability and security.

The objective of the present paper here is to focus on the need of the development of an Innovative CAPTCHA to move towards an optimum solution to web security. Also the paper utilizes the cognitive ability of human being for generating a new dictionary of CAPTCHA code. Further, it utilizes a server side scripting language, PHP for generating these codes randomly on the screen.

## II. Literature Review:

CAPTCHA, i.e. Completely Automated Public Turing Test to Tell Computers and Humans Apart is a strange kind of a lock one that is designed to be easily broken by human visual pattern matching, but not by means of automated software. The most widely used CAPTCHA challenge users to type in a string of distorted letters found in an image. As long as extracting the characters via computer vision techniques is sufficiently difficult, one can reasonably be sure that any correct answers were provided by human users.

Alternatives to the textual test include image-based CAPTCHAs which involve identification of image contents, selection of anomalous images from a set or similar tasks. These techniques have not been used as widely as the ubiquitous text in image CAPTCHAs.

The accessibility concerns with text-based CAPTCHAs have led to the introduction of audio alternatives. The Audio CAPTCHAs typically ask users to type in a series of digits as spoken in audio stream. As spoken digits can be recognized by speech recognition software, these systems typically use added noise and a variety of voices to defeat potential attacks.

The resulting audio streams may be hard to interpret, making the system accessible but perhaps less than usable. An alternative design based on transcription of spoken words eliminates the difficulties associated with remembering strings of random numbers [7].

Conclusively, there are different types of CAPTCHAs available nowadays which can be categorized as text-based, image-based or audio-based and video-based CAPTCHAs as a whole.

The work on building HIPs dates back to 1997 with the first HIP being invented at the DEC system Research center for blocking abusive automatic submission of URLs to the Alta Vista website (www.altavista.com). Since then numerous HIPs have been proposed and several have been adopted by companies to protect various services on the web. However, the basic challenge still remains the same: design a computer program that can automatically generate and grade tests that most humans can pass but current computer programs cannot pass. For a HIP to be successful in practice, it should also be fast and be capable of generating millions of unique samples a day.

Conclusively, we can say that our objective is to find a feasible solution to CAPTCHA which is readable for the user as well as hard to predict for the machine. In other words a good CAPTCHA is one that does not put too much burden on human in order to avoid discouraging the use of services. This can be summarized in figure (1.1).which shows as sweet spot to be followed for an ideal distribution of HIPs. It contains a sweet spot region where the HIPs are easy for humans to recognize but difficult for hackers to crack, not guaranteed to actually exist.

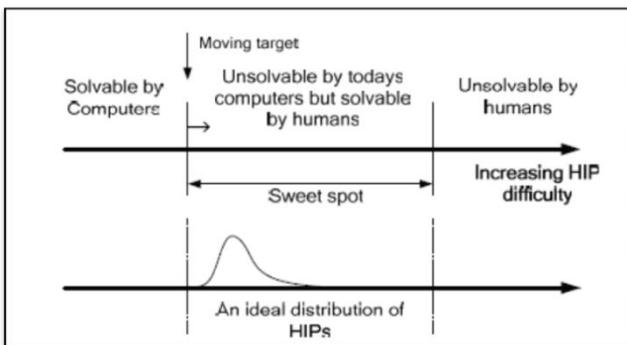


Fig. (1.1):- Regions of feasibility as a function of HIP difficulty for humans and computers algorithms.

Furthermore, automatically generated HIPs, being random in nature will have a distribution of difficulty with some particular instances extending beyond the hypothesized sweet spot.

Thus the aim is to produce such a CAPTCHA which can be in the region of sweet spot leading to development of an ideal CAPTCHA.

**III. CAPTCHA: BASED ON COGNITION**

By definition cognition means the mental action or process of acquiring knowledge and understanding through experience and senses. It is a set of abilities, skills or processes that are part of nearly every human action which can be utilized to make computers and humans apart which further leads to the objective of web security. Instead, cognitive science literature abounds with studies on visual perception showing that, for the most part, people do not require noticeably more processing time for object categorization comparative to an automated machine [8].

Based on this cognitive ability which can be utilized as a tool to differentiate a human from a machine we have generated a new Innovative CAPTCHA code. This CAPTCHA code is created by using alphabets and numbers to be hidden within any of the designs. In other words, we have taken a single number or alphabet created a design such that the specific alphanumeric letter becomes hidden between those designs, some of these developed design are shown in table (1) below.

S. No.	Image 1	Image 1	Image 1
1			
2			
3			
4			
5			

6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			

**Table 1: Designs based on cognition**

Undoubtedly the user can easily predict the hidden character or numbers within these designs through visual perception where as it would be difficult for the machine to recognize the code hidden within the specific design.

Now, these patterns would be displayed randomly of the screen with the help of secured server side scripting language PHP.

**IV. PHP: A Server side approach to produce CAPTCHA**  
In this paper we have generated the cognition based CAPTCHA code in the above section.

Now, the objective is to generate these codes randomly on the screen such that they would not be broken by any of the available machine learning algorithm.

For this the paper utilizes PHP; a server side security approach towards the generation of randomized CAPTCHA code on the screen.

PHP is a server side scripting language where the actual data is hidden from the user or automated attacker if any, it would be quite complex for a machine to pass the test which finally serves our purpose of web security.

**V. Methodology :**

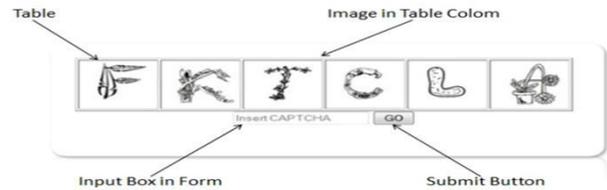
In this method first of all we generate the images based on cognitive ability of human as it is the major strength towards the web security through CAPTCHA. For this, first of all we create 24 images of each of the alphabet from A to Z, thus totally having 24 \*26= 624 images of alphabets as a whole .Now ,we create 26 folders each containing designs from alphabetic “A” to “Z “. The major feature of these images is that all images are displaying a specific Alphabet which is hidden in the drawn pattern and can be recognized only through the utilization of cognition in human.

Now, we have totally 26 folders as shown in fig (2) containing image of each alphabet .Each folder here contains the alphabet as a image having name as 1.jpg, 2.jpg etc. till the number of images available for a specific alphabet. The reason for giving numeric names to these images is that the PHP code will call the name of the image randomly through the rand function , which is applied to a numeric digit after providing the range.



**Fig (2): Images of Alphabets “A to Z” into specific Folders**

After the design of these folders we will create a table in HTML with one row of six columns in which the generated CAPTCHA will be displayed. Then a Form in HTML as shown in fig (3) is designed containing an input box where the user is supposed to enter the CAPTCHA, and a SUBMIT Button which is to be pressed so as to make the computer accept the CAPTCHA.



**Fig (3): The CAPTCHA test presented to the user**

Now comes the turn of using PHP code where we declare 6 variables which will store any 6 images to be randomly generated using rand function available in PHP. Then we save these 6 images by concatenating in a string variable say \$code.

After this, the requirement is to show the corresponding images of the randomly generated CAPTCHA code on the screen. For this we require My SQL database connectivity with My PHP Admin page. Here, we make a database containing a table with the fields namely, “id”, “Img\_name”, and “Fol\_name” .The structure of this database to generate CAPTCHA code includes table” tbl” is shown in table (2) below:

Field	Type	Null	Default
Id	int(10)	No	
Img_name	varchar(255)	No	
folder_name	varchar(255)	No	

**Table (2) the structure for table tbl\_CAPTCHA**

Here, Table (3) shows the Dumping data for table tbl\_CAPTCHA

Id	Img_name	Folder_name
1	1.jpg	A
2	2.jpg	B
3	3.jpg	C
4	4.jpg	D
5	5.jpg	E
6	6.jpg	F
7	7.jpg	G
8	8.jpg	H
9	9.jpg	I
10	10.jpg	J
11	11.jpg	K
12	12.jpg	L
13	13.jpg	M
14	14.jpg	N
15	15.jpg	O
16	16.jpg	P
17	17.jpg	Q
18	18.jpg	R
19	19.jpg	S
20	20.jpg	T
21	21.jpg	U
22	22.jpg	V
23	23.jpg	W
24	24.jpg	X
25	25.jpg	Y
26	26.jpg	Z

**Table (3) the Dumping data for table tbl\_CAPTCHA**

The major feature associated here is that whenever we refresh a new pattern will be displayed on the screen as a Challenge – Response test for the user .It will be helpful in those cases when the user will not be able to recognize the pattern displayed and wants another test to be performed as HIP.

Thus, the complete work can be summarized in the algorithm shown below:-

#### ALGORITHM OF THE MENTIONED PROGRAM:

- Step. 1 Start
- Step. 2 Create images of alphabet A to Z.
- Step. 3 Create folder naming from 1 to 26 and store images from “A” to “Z” in these folders.
- Step. 4 Create a 6 column table in html.
- Step. 5 Create a form use “POST” Method and use input box and submit button.
- Step. 6 Display images in table’s column with <IMG> tag.
- Step. 7 Use PHP and declare 6 variables (\$a, \$b, \$c, \$d, \$e, and \$f).
- Step. 8 Store an integer value in each variable use rand () function.
- Step. 9 Create a new variable CODE and join 6 variables like a string in this variable (\$Code=\$a. \$b. \$c. \$d. \$e. \$f).
- Step. 10 Make a database in MySQL id, img\_name, fol\_name.
- Step. 11 Connect database in PHP.
- Step. 12 Retrieve img\_name and fol\_name with id use with sql\_query.
- Step. 13 Show images in the table with <? PHP?>
- Step. 14 Declare a variable USER and read when form submits (POST Method apply).
- Step. 15 If \$user==\$code
- Step. 16 {
- Step. 17 print “True”
- Step. 18 }
- Step. 19 else
- Step. 20 {
- Step. 21 print “False”
- Step. 22 }
- Step. 23 End.

#### VI. CONCLUSION:

The present paper has utilized the cognition ability of human as a base to develop CAPTCHA along with the utilization of PHP for randomized code. The PHP is chosen here as its code is executed on the server and the result is returned to the browser as a plain HTML which makes it more secure to use. Also, many generic web servers support server side scripting using Active Server Pages (ASP), PHP, OR other scripting languages. This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged.

#### VII. REFERENCES:

- [1] "AshwaniGarg, Shekhar Singh, "A Review of web application security vulnerabilities", *International Journal of Advanced Research in computer Science and Software Engineering*", Vol 3, Issue 1, Jan 2013 Issu : 2277 128X available at [www.ijforcsse.com](http://www.ijforcsse.com).
- [2] "Rizwan UR Rahman, "Survey on CAPTCHA systems", *Journal of Global Research in Computer Science*", Vol 3, No. 6, June 2012 Available Online at [www.jgres.info](http://www.jgres.info).
- [3] "Richard Chow, Philippe Golle, Markus JakobssonLusha Wang, Xiaofeng Wang", "Making CAPTCHA Clickable" Available at [luswang@indiana.edu](mailto:luswang@indiana.edu).
- [4] "Joseph B. Evans, Weichad Wang and Benjamin J. Ewy", "Wireless Networking Security; Open issues in trust, management, interoperation and measurement", *Int. J. Security and Networks*, vol 1, No 1/2 , 2006.
- [5] "Kumar Chellapilla, Kevin Larson Patrice Simard, Mary Czerwinski", "Computers beat humans at single character recognition in erading based human interaction proofs (HIPs)", *Microsoft Research Redmond, WA 98053*.
- [6] Monica Chew and Henry S. Baird "BaffleText: a Human Interactive Proof" Published in *Proceedings of the SPIE/IS&T Document Recognition & Retrieval Conf. X, Santa Clara, CA, January 22-23, 2003*.
- [7] "Siddharth Ghansela", "Network Security: Attack, tools and technique", *International Journal of Advance Research in Computer Science and Software Engineering*", Vol.3, Issue 6, June 2013, ISSN: 2277128X.