

# Implementation of Advanced Captcha based Security System

Sandeep Mahato<sup>1</sup>, Priyaswarupa Rani Pati<sup>2</sup>, Prashant Tiwari<sup>3</sup>,  
Raj Gaurav Mishra<sup>4</sup>

sandeep.mahato94@gmail.com

---

## Abstract

Web sites providing different web services and applications with resources which millions of users need to access. This might lead to unethical use and wastage of resources. To avoid this Captcha security test which is an image verification test, is being used. Image Verification security method ensures the HIP where human user can easily recognize the same from distorted text. Captcha should not be only difficult to solve by computer programs, it should also be user friendly.

## Keywords

Web Security, Captcha Security, BOT.

## Introduction

In this work we analysed previous works on Captcha and found that so far many types of text based Captchas have been proposed and developed. All of them tried to make the test easier for human user and difficult enough for computer programs and bots. In their efforts to make a hard Captcha many schemes used background confusion, blurring and tilting of texts which may lead to hard enough for human user to pass the test. More background confusion and tilting and twisting of test may cause recognition problem for human user also.

In this article a robust Captcha was proposed. The proposed scheme is designed in asp.net under visual studio platform with C# which is easy to implement. The solving process was kept simple so that human user can solve the test easily; they need to provide input according to associated query not entire Captcha character set. Users require little time to provide input with accuracy. Principle design features were considered to provide better security and construct a generic text based Captcha. To avoid several attacks like dictionary attack, segmentation attack and brute force attack (random guess attack) we have varied some considerable features and used different queries with randomness [3].

---

<sup>1</sup>M.Tech (computer Science), Central University of Jharkhand, India. sandeep.mahato@gmail.com

<sup>2</sup>M.Tech (computer Science) Central University of Jharkhand, India. prashantcuj@gmail.com

<sup>3</sup>Central University of Jharkhand, India.

<sup>4</sup>Faculty Member, The ICFAI University, Dehradun (Uttarakhand), India. rajgmishra@hctl.org

---

## Existing Text based Captcha scheme

In this existing robust Captcha scheme algorithm random alphanumeric code of fixed length (size 6) is generated. This alphanumeric code is then converted into image with some noise. After this a random query is related to this code image in same session and user is asked to answer this query. Validation is done with user input and session value to allow user to proceed.

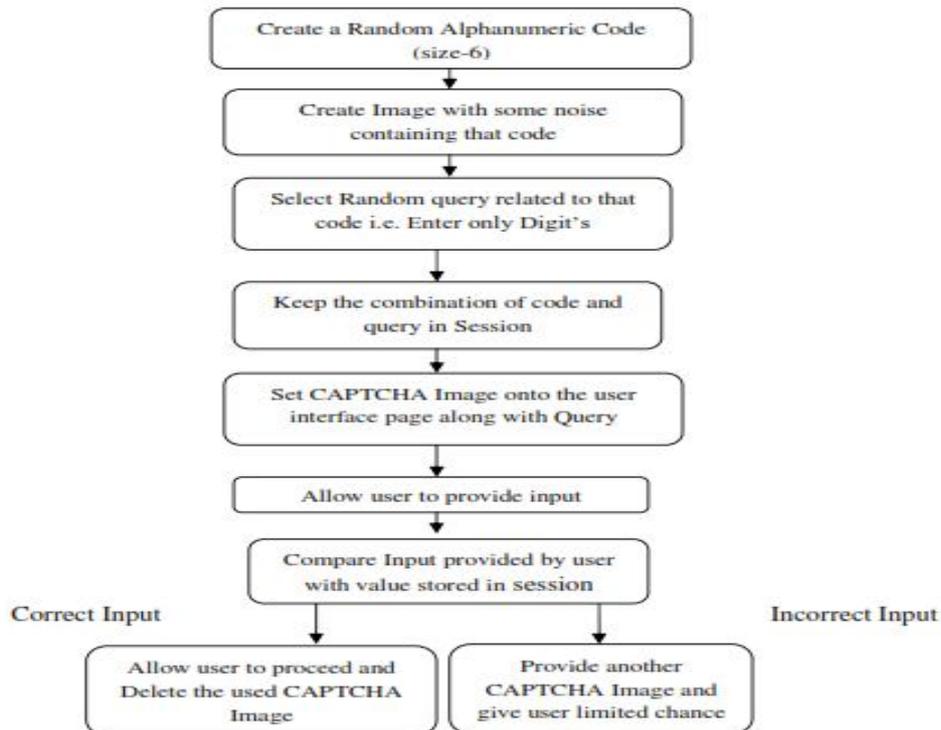


Figure1: Algorithm of an existing robust CAPTCHA [1]

## Proposed Scheme and Methodology

In this chapter, we propose a robust CAPTCHA technique which includes two stages: First is Captcha generation in which a random series of characters was generated. Generated series of characters are combination of alphabets (upper or lower case) and numbers which may avoid the dictionary attack. In second stage a random query is associated with generated query. After that user is asked to pass the test, in which according to associated query user requires submitting few characters rather than typing entire Captcha. It saves time for user and associated query makes it hard to solve for program bots.

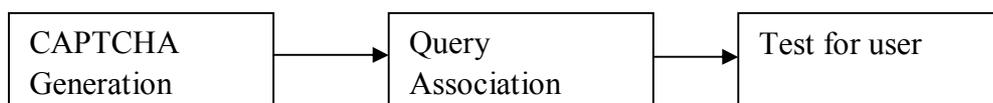


Figure 2: Block diagram of proposed scheme

## Problem Definition

In this work we analysed previous works on Captcha and found that so far many types of text based Captchas have been proposed and developed .All of them tried to make the test easier for human user and difficult enough for computer programs and bots. In their efforts to make a hard Captcha many schemes used background confusion, blurring and tilting of texts which may lead to hard enough for human user to pass the test. More background confusion and tilting and twisting of test may cause recognition problem for human user also. Image based and group based Captcha tests need large data base and may face usability problem. So the scheme should be simple and secured enough to avoid abuses from bots.

## Our Objective for Proposed Scheme

1. To develop a robust text based Captcha following the secure Captcha design principles with random character set made of alphabets (upper and lower case) and numbers.
2. To improve security and hardness length and font size of generated text is randomized every time.
3. Associate a random question to new generated Captcha to improve usability and user response time

## Two Stages for proposed Captcha technique

**Captcha Generation:** In this stage an alphanumeric code is generated using rand () function to randomize the combination of alphabets (upper or lower case) and numbers. The font size and length of generated code is again randomized to improve the robustness of the Captcha code. After this code is converted into image with little background confusion and set on the user interface for the test.

**Query Association:** After Captcha generation, a random query from a set of queries is associated with the Captcha code to improve the response time of user as user need not type the entire code . It also improves the usability and can avoid many attacks like Brute force attack which tries to randomly guess the correct answer. In this proposed work adding query to Captcha code is an important task. Type of query to be associated with code is also very important. Query must be something which is easy to understand and easy to response for user to improve the usability of this scheme.

## Algorithm of proposed Captcha Scheme

1. Create random alphanumeric code
  - 1.1. Vary the generated alphanumeric code by random length and font size.
2. Create Image with little background confusion
3. Randomly select a query related to that code.
4. Keeping the combination of generated code and query in session.

5. Set the CAPTCHA combination of generated code and query on user interface (login page).

6. Ask user to provide proper input.

7. Compare the input provided by user and session value.

7.1. If inputs provided by user and session value are same with correctness then user is allowed to access service page.

7.2. If inputs provided by user and session value are not same then user is asked to provide proper inputs again.

## Implementation and Experiments

### About Visual Studio 2010 and functions used

The above proposed scheme was implemented with the help of visual studio 2010. Visual Studio 2010 allows to design and develop windows and web based applications with language support like Visual Basic, Visual C# and visual C++ etc. In this work C# language was used to make active server pages for a web site to implement proposed scheme.

**Random. Next int(int n,int m)** - In this thesis work this function performs randomization of font size and queries. Parameters specified n for minimum value and m for maximum value.

**Random. Next (int m)** - In this thesis work this function performs randomization of characters. Parameters specified m for maximum value.

**Graphics. Clear (color color)** - This function clears entire drawing surface and fills with specified background colour.

**Graphics. DrawString (string s, font font, brush brush, Rectangle layout)** – This function draws specified text string in specified rectangle with specified font and brush objects.

The above functions and some more functions were used to implement the proposed Captcha scheme.

## Analysis of effect of age groups and response time of users

This thesis work conducted an experiment to study the effect of age groups and response time of users on Captcha schemes.

User	Age	Response time (in sec.)	Input/solving time(in sec)	Experience of use
User-1	27	2	4.99	Good
User-2	24	3	6.10	Good
User-3	28	3	7.10	Good
User-4	25	4	6.80	Good
User-5	30	4	7.90	Good
User-6	23	3	5.50	Excellent
User-7	25	3	5.40	Good
User-8				
User-9				
User-10				
User-11				

Table 1

Based on our experiment we found that a good experience of using the proposed Captcha for every age group which ensures better usability. We also found that the average response time (3.14) and input/solving time(6.25) have been improved. It may be acceptable according to Bursztein's recommendations [2] that the optimal recognition time for text-based Captchas is under 9 seconds.

## Result and Discussion

The image shows a web application interface for user login. At the top, there is a navigation bar with links for Home, Login, About Us, and Contact Us. Below the navigation bar, there are two input fields: 'User ID:' and 'password:'. Below these fields is a Captcha image displaying the text 'DFu7Qc' in green on a dark background. To the right of the Captcha is a 'Refresh' button. Below the Captcha is a small input field with the instruction 'Type 1st,4th and 5th character' and a 'Submit' button.

Figure 3: Snapshot before user logged in



The screenshot shows a web application interface for user login. At the top, there is a navigation bar with links for 'Home', 'Login', 'About Us', and 'Contact Us'. Below the navigation bar, there is a login form. The form includes a 'User ID:' field with the value 'a' and a 'password:' field with a masked character. A captcha image is displayed, showing the text 'DFu7Qc' in a stylized font. To the right of the captcha image is a 'Refresh' button. Below the captcha image, there is a text input field containing 'D7Q' and a prompt that says 'Type 1st,4th and 5th character'. At the bottom of the form is a 'Submit' button.

*Figure 4: Snapshot when user logging in*

Snapshot of proposed scheme were shown above in which it can be observed that here user needs to provide input according to query associated with the test. It saves time for user as it only provides few from entire code. Here type of query associated to Captcha code is important and choosing query for query set should be done carefully. If query like "type how many numeric character? ", "Type how many alphabets in the code?" are to be associated test may suffer with brute force or random guess attack. So we need to avoid query like this in which answer is single character or number. We must consider queries whose answers will be easier for human user but difficult for bot programs.

## Conclusion

In this article we studied different recently and frequently used Captcha. According problem statement a robust Captcha should not be only difficult to solve by computer programs, it should also be user friendly. Currently proposed scheme is designed in asp.net under visual studio platform with C# which is easy to implement. We kept the solving process simple so that human user can solve the test easily; they need to provide input according to associated query not entire Captcha character set. Users require little time to provide input with accuracy. We considered principle design features to provide better security and construct a generic text based Captcha. To avoid several attacks like dictionary attack, segmentation attack and brute force attack (random guess attack) we have varied some considerable features and used different queries with randomness. It results improved usability, solving time and robustness of text based Captcha. Using larger query set and choosing different kind of queries may improve the robustness of this type of Captcha scheme in future.

## References

- [1] Poonam Yadav; Sujata, Security Issues in Cloud Computing Solution of DDOS and Introducing Two-Tier CAPTCHA, IJCCSA, Volume 3, No. 3, June 2013.
- [2] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, How good are humans at solving captchas? A large scale evaluation, in Proceedings of the 2010 IEEE Symposium on Security and Privacy, ser. SP'10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 399-413.
- [3] Jeff Yen, Ahmad Salah, "CAPTCHA security-Case Study", IEEE CS, 2009.
- [4] Sandeep Mahato; Varun Prakash Saxena; Raj Gaurav Mishra, Securing Web Services and Application using Captcha Security, HCTL Open International Journal of Technology Innovations and Research (IJTIR), Volume 14, April 2015, eISSN: 2321-1814, ISBN (Print): 978-1-62951-946-3.
- [5] Sandeep Mahato; Varun Prakash Saxena; Devendra Bhavsar, A Survey of Captcha based Web and Application Security Methods and Techniques, HCTL Open International Journal of Technology Innovations and Research (IJTIR), Volume 14, April 2015, eISSN: 2321-1814, ISBN (Print): 978-1-62951-946-3.

---

*This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>).*

© 2015 by the Authors. Licensed by HCTL Open, India.