

Generation of Random Order using Mouse Clicked in Text Based Color CAPTCHA

¹S. Pradeep Kumar, ²Dhinakaran K, ³G. Kalpana, ⁴Maria Daniel Raj. I,
⁵Karthik. R, ⁶Karthy.V

¹Associate Professor, ²Assistant Professor, ^{4,5,6}UG Scholars,

Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, Chennai, India

³Assistant Professor, Computer Science and Engineering, Sri Venkateswara College of Engineering,
Chennai, India

Article Info

Volume 82

Page Number: 10557 - 10571

Publication Issue:

January-February 2020

Abstract

CAPTCHA is used as a standard security mechanism to prevent the bots to enter into the commercial websites like e-governance, inventory, educational sector and so on. Now a days hackers wrote malicious program and enter into the website to destroy their resources. To prevent such kind of activities, in this paper introduces a Random Ordered Mouse Clicked Curve based CAPTCHA (ROMCCC) is completely differing from the existing web CAPTCHA. It combines the features of text based and image based CAPTCHA. In order to improve the representation of a CAPTCHA character by involving two major fields such as user friendly and secure, the methodology proposed in this paper introduces a Curve (upper/lower) Based Text (CBT) CAPTCHA. A group of CAPTCHA characters (i.e 6 to 8) is to be arranged and look like a parabola shape. In the CAPTCHA screening test, a set of CAPTCHA character along with an available usage character represented. In this proposed methodology, a random number is to be displayed under each CAPTCHA character. Based on the sorted order of the random number, userclick the usage character one by one and these characters are displayed in the text box. After clicking all the usage characters, then only the user can sign and enter into the corresponding websites. This type of CAPTCHA can be applied in the major available web area. The number of usable CAPTCHA characters is to differ for each attempt and also each CAPTCHA character is represented in different colors. This makes the same character can have the chance to represent in different colors which in turn the pixel intensity can vary. Meanwhile of displaying usage characters are represented in different order for each attempt along with the CAPTCHA character set. Consequently this type of process can be broken by any robot and web security can be strengthened in the areas such as inventory, defense and banking sectors.

Keywords: ROMCC–; Random Ordered Mouse Clicked Curve based CAPTCHA, CAPTCHA-Completely Automated Public Turing test to tell Computer and Human Apart., Web security

Article History

Article Received: 18 May 2019

Revised: 14 July 2019

Accepted: 22 December 2019

Publication: 19 February 2020

1. Introduction

In the modern era, the requirement of online services in the internet for providing email, purchasing, search engine with a available resources. In such a case, denial of services made by a malicious program has become a serious problem and hackers create a false account to destroy the resources.

In order to avoid such a problem from the malicious problem, the CAPTCHA has been introduced to distinguish between a Robot and a user[1]. Most of the web application requires the user to do registration to enter into the website. The server of a particular website can give a large number of users to access this information for the daily needs at any point of time. In generally humans are superior than machine to recognize the Image or text based CAPTCHA. But now a days to make the CAPTCHA face two defects such that i) not able to recognize the character by the user and ii) hackers wrote a malicious program to extract the character from the CAPTCHA set. To clear the above two aspects, in this paper introduces a Random order Mouse Clicked Curve based CAPTCHA provides convenience to the user and not able to detect the CAPTCHA character at any time for the hackers. It combine the features of text based and image based CAPTCHA[2]. The representation of CAPTCHA text consists of two parts such as a character image with noise and an input text box. The appearance of CAPTCHA characters is tilted to some specific angle and few lines or curves are added to make confusion for hackers to grasp easily. This mechanism helps to prevent the CAPTCHA generation system from the hackers. Today, many of the websites registration process can be accomplished by following some set of procedure. After successful completion of the procedure, the

particular web application can be utilized For a sake of discussion, the design principles of well known websites such as google, yahoo and MSN use different styles of own CAPTCHA representation. In this connection, the Yahoo and MSN do not use any color in representing the CAPTCHA character (as noise) whereas Badongo uses colored lines as noise and YouTube uses colored blocks.

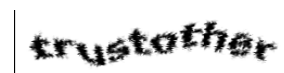
There are some rules to be followed for design the CAPTCHA

(i) Computer program can able to generate the CAPTCHA character then and there.

(ii) User can able to recognize these CAPTCHA characters with a reasonable amount of time.

(iii) Hackers wrote malicious program using sophisticated software to detect CAPTCHA character at any time.

The fig 1 shows, some of the draw backs faced in commonly used websites[3].



WikipediaFace book

(Hackers can do easy Segmentation)



(Same black and white color can be used in background and CAPTCHA character representation)

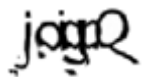


Google Microsoft

(Human user got confuse to recognize CAPTCHA)



(Same color can be used for cracking and CAPTCHA character)



(Human user got confuse to Recognize facebook CAPTCHA)



(Hackers can do easy segmentation) ReCAPTCHA

Figure 1: CAPTCHA problems faced in commonly used websites

The number of characters used in the CAPTCHA set is small. Since characters can have their own fixed pixel count and it is possible to identify them through Optical Character Recognition (OCR) and segmentation process. Due to the similar color usage in the CAPTCHA representation, some noise and distortion are to be embedded in the CAPTCHA representation and it creates a big problem to the human user for recognizing the same (survey on CAPTCHA system).

Hence, this proposed work introduces a Curve Based Text (CBT) CAPTCHA for displaying a CAPTCHA over the webpage. The web users have to fill all the necessary information after that only the CAPTCHA page will be appeared. The representation of CAPTCHA must be human friendly and it can restrict the bots to do the segmentation process. Meanwhile user enter all the CAPTCHA character with the help of mouse clicked and

all the set of alphanumeric character are displayed in the CAPTCHA screening test.

In this paper, the section II discusses the related works with their drawbacks. The section III describes the new proposed method for enabling security on web access by utilizing the Random order mouse based CAPTCHAs. The IV section explores the creation of dynamic random order CAPTCHA's character scheme. The section V shows the implementation of Random ordered Mouse clicked in text based CAPTCHA character scheme. The section VI demonstrates the completion of CAPTCHA text by the user. Finally last section VII describes the conclusion and Future enhancement on CAPTCHA scheme.

2. Related Works

Gimphy method uses the CAPTCHA word for the screening test. Most of the word are available from the dictionary and it is easily broken by the hackers. It is found in the yahoo website and to identify the CAPTCHA word easily. In a cluttered image, the task is to identify three of the approximately seven words[6]. Using machine learning algorithm, Chellapilla and Simard had successfully attempt to break a number of visual CAPTCHA taken from the web- Google and Yahoo and lead a success rate of around 66%[7]. ElieBursztein have suggested that using DeCaptcha tools to break most of the text-based or Visual based CAPTCHA and found in a well know websites such as eBay, Digg and Wikipedia[5]. Using modern technology grows and it is possible to enhance the security of existing CAPTCHA by adding noise and distortion for arranging the CAPTCHA character more tightly. But it is harder for the user to recognize the CAPTCHA character in higher rate and network load.[4]. In this connection, the other researcher Yan has broken a most of the visual CAPTCHA as

collected from Captcha service.org with a high success rate of 100%. And also, the character segmentation can be easily done with the help of current available OCR (Optical Character Recognition) software and it simply counts the number of pixels in each segment. Hence, it is necessary to propose a new methodology that can consider all the above discussed issues while displaying the CAPTCHA's in securable manner.

3. Proposed Work

Evolution trends in the CAPTCHA entry test has introduced a Dynamic random order Mouse clicked Curve based CAPTCHA. This proposed work shows the journey from text based to the mouse interaction based challenging CAPTCHA test. User fills all the sufficient details in the web applications form then only the CAPTCHA screening test will appear. The separate window for the CAPTCHA screening test is to be displayed under the same web application form. Each and every CAPTCHA character are to be represented in different colors. Collectivity of all CAPTCHA characters shows the upper/lower curve formats. At present, most of the web applications are not representing the curve based CAPTCHAs for enabling registration process. At the same time, the user cannot get confused or irritate to recognize a CAPTCHA's character. In our proposed work, bots cannot easily get the chance to break a CAPTCHA test. In our work, the CAPTCHAs characters are collided with other and each character is titled in different angle. In addition to that, every character below corresponding random number is to be displayed. Inside the right side of the CAPTCHA screening test, a set of alphabets and numerical are displayed under a separate grid layout. The total number of characters set are displayed in the screening text is to be

differed in various ways by using the same number of CAPTCHA character or same CAPTCHA representation. Based on the sorted random number, user has to click all the character one by one and it is displayed in the given empty text box. The random number displayed is differed for the same number of CAPTCHA's representations. This proposed CAPTCHA is a convenient graphical user interface to the human user and it is free from OCR attacks. So the hackers cannot have the chance break a CAPTCHA design test. Each and every attempt, the number of CAPTCHAs characters is to be varied (i.e. 6 to 8). Due to the various colors used, the same pixel count of one character may have the chance to represent some other character. So, the hackers cannot have the chance to predict a CAPTCHA character representation.

4. Creation of Dynamic Random Ordered Mouse Clicked Captcha Characters

CAPTCHA are used to give the security for the web based applications. The strength of the CAPTCHA are extensively depend upon the degree of distortion and it also have the chance lead to the failure recognition by the user[8]. In this CAPTCHA character representation, all the characters are displayed in different colors. It leads to the pixel intensity values can vary for the same character with different colors. In addition to that some graphical operation and cracks are added in the CAPTCHA screening text. Meanwhile it won't disturb the human user to recognize the CAPTCHA character[9] All the CAPTCHA characters that are collided in the middle of neighboring character and forms a parabolic shape.it does not lead to the OCR attack, online attack, dictionary attack[10]. OCR is a program that doesn't have the chance to recognize the collided CAPTCHA characters that have some cracks. Meanwhile all the

clicked characters that are entered thru dynamic random order generation. This mechanism does not have chance for the hackers lead to the dictionary and online attacks. In this paper, the strength of the CAPTCHA havethree aspects, one is the representation of CAPTCHA(A), preprocessing(B) and another is the generation of dynamic random order CAPTCHA(C).

A. Creation of Curve Based Captcha

The sample curve based text CAPTCHA as shown in the Figure 2, which has been distorted and tilted in different positions. All the text based CAPTCHA's character that may be generated either in upper or lower curve. The white color used as a background text and a black color is used to crack the CAPTCHA's character.

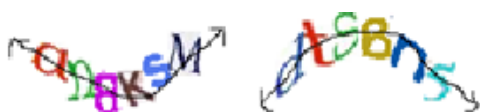


Figure 2: Lower Curve CAPTCHA Upper Curve CAPTCHA

In general, the alphabetic characters (a to z : upper and lower case) and numeric symbols (0,1,..9) are used in the text CAPTCHA representation. In this scheme, each and every CAPTCHA characters are to be represented in different colors. Due to various color representation, user may not confused to recognize the CAPTCHA's set. Meanwhile, the same character can have the chance to represent in different colors. This is helpful for displaying the same character with different colors and different no. of pixels.

(i) Working Procedures

The random sorted characters are generated and represent a word, which is very difficult for the

bots to predict due to the hard segmentation and different colors CAPTCHAs character representation. User filled all the sufficient details in the web application form, the system is allowed for the CAPTCHA screening test.

(ii) Create CAPTCHA word

Using the random number generator procedures, the collection of CAPTCHA characters from the primary database is used to represent a word in a text. The collection of all CAPTCHAs characters appears like parabolic curve formats. Thus, the representation of this format makes the segmentation much harder. The user has to fill up all the CAPTCHA characters as given in the text box.

(iii)Primary database

This database uses a CAPTCHA character set of 62 characters ($\{A,B..Z\}, \{a,b...z\}, \{0,1,2..9\}$). With the same color usage, the different pixel count is varied for all the 62 characters. But the same time, the same pixel count may have the chance to get in some other color of different characters. In the primary database, a CAPTCHA character is maintained along with a number of use pixels, color and a recognized character (as shown in the table 1).

Table 1: Sample CAPTCHA character set along with recognized letter

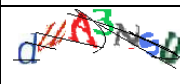


Color used	CAPTCHA character	Pixel count	Recognized letter
Red	K	176	K
Violet	8	185	8
Green	S	193	S
Blue	g	169	g

Secondary Database

Using the random number generator, the collection of CAPTCHA character from the primary database is stored in the secondary data

base in a different order. Using the graphical operation, the CAPTCHA characters are to be represented in different styles (as discussed in C-ii). In this database maintains a 600 CAPTCHA character set. Each set contains a different number of CAPTCHA characters(6 to 8) and a corresponding random number displayed(as shown in Table 2). Based on the sorted random numbers, a set of equivalent characters frame a word that is maintained in a character set. These words are not available in the current using dictionary. Using a random number generator collectsthe a particular set and put up a corresponding equivalent CAPTCHA characters in the screening test process. In the set the number of characters used may vary from 6 to 8.

Table 2: Sample CAPTCHA word with Random number Allocation

CAPTCHA Character	No of Characters	Random Number	Recognized Word
	7	2 4 5 1 6 7 3	U 3 N d s U A
	6	6 2 1 4 3 5	5 t d 8 S n
	6	3 4 1 5 2 6	d C a R 2 e

All the CAPTCHA's characters are used in the set are to be represented in different colors (Table II is shown). Hence the same character lead to a different pixel count and the intensity value for the same CAPTCHA. It gives the prediction and security to the owner of the website for maintaining the CAPTCHA character set allotted for the screening test. The number of random numbers is used in the

database set is equivalent to the number of CAPTCHA character used (a shown in the Table 2).

(iv)Color Representation

In general, white color is used as a background text and black color is used for cracking the CAPTCHA character set that has to be appeared in a text box. The cracks may be dot or curve or line. Other than white and black color may used in the CAPTCHA screening test.

B. Preprocessing

The text based CAPTCHA can be analyzed for ensuring the security during the registration process pertaining to the web access. In such a case, there is a need for understanding the displayed characters by the system, which can be usually done through the segmentation process. Before going to the segmentation, CAPTCHA should follow the following constraints.

(i) Readable

Under any kind of circumstances, the user who is trying to access the web does not get confused to recognize the CAPTCHA character. All the CAPTCHA characters are in readable form and it should be apparent to the human user only. Cracks are represented by line, dots and curves in the CAPTCHA scheme. User should not get irritated at any point of time. The process of smoothing has to be done to remove the black pixel.

(ii)Stability

The each and every characters of the given CAPTCHA that are collided with each other especially in the middle portion of every character as shown in the figure (2). In this way, we cannot see a single segmented character separately. That means, each segmented

character requires further more preprocessing for removing the neighboring color pixel(as shown in the figure 3).

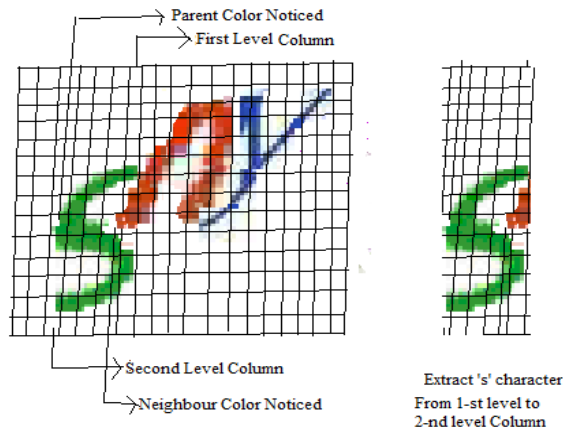


Figure 3: Shows Extraction of 's' CAPTCHA character

The working procedure for detect a CAPTCHA:

(a) Start with the Bottom-left pixel color value

Track the same column in upward and downward to detect the color (red, green, blue...) pixel value in a CAPTCHA set.

(b) Identification of primary and neighbor color value

If any color pixel value obtained, note the color pixel value and then consider as parent color. Initially mark the entire column for the first identification of parent color. This column consider as first segmentation line. Trace the same parent color in every column using LSA. If any other color obtained consider as neighbor color (as shown in Fig 5)

(c) Segmentation of vertical columns

While tracing the parent color in the CAPTCHA set, if the parent color as not obtained in any column, stop the process. This column consider as second segmented line.

Then segment the group of column from the first segment line to the second segment line.

In the segmented 's' character is in green color have the some portion of neighbor 'n' character brown color pixel position. Meanwhile black color is also added to crack the entire set of CAPTCHAs character set. It clearly shows the hacker faced a great difficulty to detect a single segmented character alone.

(iii) Clarity

The above process i.e the above paragraph (i) to (iii) is used to remove the noise and to assure the security challenges among the hackers to identify the segmented single character. Due to the various colors representation of CAPTCHAs, the hackers can face a difficult task to understand the original CAPTCHAs from the given different 256 pixel intensity values. In general, the color of the image consists of white color (which refers to the foreground text) and the combination of rest of the colors (which refers to the CAPTCHA characters). Hence, it is very difficult to work on 256 pixel intensity values for identifying CAPTCHA characters by the hackers.

C. Generation of Dynamic random order CAPTCHA

In the CAPTCHA screening test, a separate window is allotted for displaying the CAPTCHA characters along with dynamic random numbers as shown in the fig 4. Inside the right side of the screening text, all the alphanumerical characters are available in different grid layout of fixed size. The arrangement of these characters are different from the standard keyboard format. In addition to that, four labels are used below the user clicked characters set.

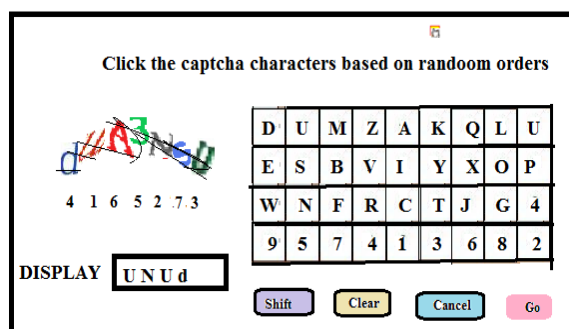


Figure 4: Mouse clicked based CAPTCHA

Shift: It is helpful to display Upper character

Clear: It delete the entire user entry characters

Cancel: It delete a latest typed character

Go: Go and proceed for the CAPTCHA screening test

1. User Entry

User filled all the information in the available web form and then allowed to attempt for the CAPTCHA screening test. Each and every CAPTCHA character corresponding random number is displayed. Based on the sorted random order, the CAPTCHA test asks the user to click the characters (with the help of a mouse) available in the challenging area of a given screen [11]. The CAPTCHA screening test will appear only after the successful completion of all entries in the given web application form. If the user not able to click the sorted CAPTCHA characters correctly, once gain the new CAPTCHA screening test will appear in the same given web form. The test is allotted for the maximum of four times in order to restrict the bots to access the information from the web.

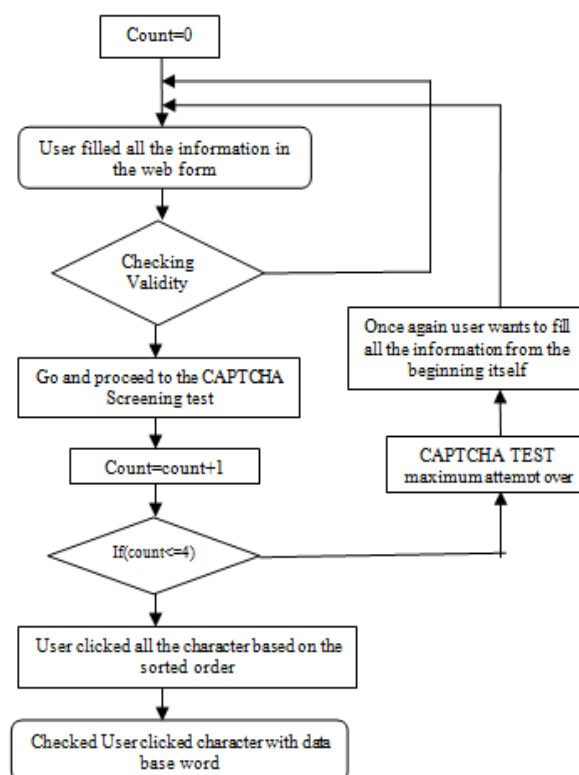





Figure 5: Working flow of a Mouse_Clicked_CAPTCHA character screening test

CAPTCHA screening test will appear only after the successful completion of all entries in the given web application form. If the user not able to click the sorted CAPTCHA characters correctly, once gain the a new CAPTCHA screening test will appear in the same given web form. The test is allotted for the maximum of four times in order to restrict the bots to access the information from the web[12].

2. Screening Text display

The total number of CAPTCHA characters are allotted in the screening test may vary from 6 to 8. The white colors is used to represent a background text and black color is used to crack the screening text box. The characters ((a,b,..y,z),(A,B,..Y,Z),(0,1..9)) are used in the CAPTCHA screening test[13].

Table 3: Lookuptableentries

Letter	Color Used	Pixel Count	CAPTCHA Character
S	Violet	169	
S	Blue	175	
S	Green	184	

In the table 3, the CAPTCHA character 'S' are represented in different colors and also tilted in different position. It lead to the difference in the pixel intensity value and size. Each and every character can have a different pixel count with respect to a particular color. Thru the pixel count and color used to recognize a character and maintain in the database character set. Hence the hackers cannot predict the CAPTCHA character at anywhere and anytime[14].

Inside the right side of the screening test, all the usage characters $\{(a,b..z),(0,1..9)\}$ are displayed in a separate grid layout. The representation order of all usage characters are differed by each attempt. It should be completely differ from the keyboard character order.

3. Representation of CAPTCHA characters and Usagecharacters

All the CAPTCHA characters are readable only by the user. Some noise such as line, dots and curves are added in the screening test. User never get confused at any time. Each CAPTCHA characters are colliding in the middle of neighboring characters. Hence the hackers have a negative chance to break a single CAPTCHA character alone[15].

User entered all the characters thru means of mouse in a available grid layout. Each

characters hold a separate grid key in the screening test. The usage characters are $\{(a,b,..z),(0,1,..9)\}$. There are various combination to display all the 36 usage characters in the separate grid layout structure (Fig 2).

Graphical Operation

The displaying the CAPTCHA character in the screening text leads to the graphical operations are,

- (1) Scaling: Some of the CAPTCHA character may shrink or enlarge in either in 'X' or 'Y' axis. It can be held in difference in size of 20%.
- (2) Rotation: Some of the CAPTCHAs character that may rotate in between 20 to -20 degree.
- (3) Translation: Using the matrix transformation to translatea particular CAPTCHA character in a available text box.
- (4) Sliding: Some CAPTCHA character may optimally slide either in left or right relative to row above/below.

All the usage characters are displayed inside the CAPTCHA screening test of a user typed web application form. Each character holds a separate grid layout. So there are 36 separate $\{(a,b..z),(0,1..9)\}$ grid layout for the characters used. Each and every CAPTCHA test, all the arrangement of usage characters is to differ. There are maintaining a ten different order to confuse the bots for predicting the correct characters.

Advantages

- Bots using sophisticated software cannot have the chance to predict the CAPTCHA character at any time.

ii. Restrict to many attacks such as On line guessing attacks, OCR attacks, Dictionary attacks...

iii. Even using graphical operation, user never gets confused to recognize a CAPTCHA character in the screening test.

5. Implementation

The representation of CAPTCHA must be a user friendly to the user and strongly restrict the bots to enter into the commercial websites. In the database maintains nearly a 600 CAPTCHA character set. Using the random number generator collect the CAPTCHA character from the database and allotted in the screening test. The working algorithm (mouse clicked CAPTCHA) holds the working process of a CAPTCHA screening test. The proposed methodology is undergoing a different objective based on the user entry and representation of CAPTCHAs.

i. User Entry

Since our technique is based on the combination of text and imaged. The user does not entry the characters as it is in the screening test.

ii. Image Recognition

A separate window is allotted in the CAPTCHA screening test. The User clicked all the recognized character based on dynamic order CAPTCHAs.

iii. Less Storage

Small sized window is maintained during the screening test. Typically of user use mouse cursor and provide easily visible to the user.

iv. OCR and Dictionary attack

CAPTCHA character is represented on the curve based format. All the CAPTCHAs character is represented in different colors. It does not lead to the OCR and dictionary attack.

The time required to solve the clickable CAPTCHAs cursor is to identify the English letter and numerical in the screening test is less. The frequency of repetitive challenge lead to the variability in the number of CAPTCHs character used (6 to 8) in the screen text, representation (upper/lower curve) and useable character. Curve based CAPTCHA were solved with the help of a mouse and traditionally CAPTCHA were solved with the help of the keyboard.

Table 4: Average Time to solve Clickable vs Traditional CAPTCHA

Without Use Dynamic Order			
Traditional CAPTCHA character screening text.(Keyboard Entry)		Clickable CAPTCHA character screening text (Mouse Entry)	
No of CAPTCHA character used	Average Time Period	No of CAPTCHA character used	Average Time Period
6	10-11	6	6-7
7	12-14	7	7-9
8	13-15	8	10-12
Use Dynamic Order			
Traditional CAPTCHA character screening text.(Keyboard Entry)		Clickable CAPTCHA character screening text (Mouse Entry)	
No of CAPTCHA character used	Average Time Period	No of CAPTCHA character used	Average Time Period
6	16-18	6	9-10

7	19-21	7	10-12
8	21-24	8	12-14

The Table 4 shows the average time to solve the traditional and clickable CAPTCHAs test using dynamic and without dynamic order representation. In the above observation made clickable CAPTCHA can be solved 40% less time period than the traditional CAPTCHA character screening text in without using the dynamic order and 33% less time period using dynamic order CAPTCHA. Timing measurements were taken first in the personal computer system and solved with the help of computer keyboard and clickable mouse entry. The observation can also observe that clickable the CAPTCHA text provides much more faster than the cellular phone.

1. Algorithm: mouse clicked CAPTCHA

Var count: Number of times a user appears the CAPTCHA

screening test

maximum_attempt: limit for the maximum number of

times to the user appear in the screening test

Begin

Step 1: User entered all the entries in the web application

Form

Step 2: Count=0

Step 3: If user entry information are “wrong” or “empty”

Display a Message ,”Invalid data”

Else

While(count<=maximum_attempt)

Begin

Step 4: Display a message ,”Go and proceed for the

CAPTCHA screening test”

Step 5: count=count+1

Step 6: User clicked the CAPTCHA character based

on sorted order

Step 7: If the user entry CAPTCHA character matched

with the data base set

Display a message, ”CAPTCHA DONE SUCESSFULLY”

Else

Count=count+1

Goto Step 1.

End

End

2.Generation of random ordered Mouse clicked

CAPTCHA

The various statistical analysis can be carried out to generate a CAPTCHA character set. It uses Cryptographic hash function can be used to generate a random ordered mouse clicked CAPTCHA(ROMCC) in a high securable manner[16]. Random number generator uses a various combination of all available characters{(a,b..z),(A,B..Z),(0.1,..9)}. Thru the sorted random number, pixel count and color used to create a recognize word and maintain in

the database itself (as discussed in section IV.3).

Algorithm: Random Number Generator

Input: $x, a[x], \text{key}$

Output $\text{random_data}(\text{key}, a[x])$

//X: no of characters used

//Key: hash key function

//A[x]: an array containing data set

Set contains
 $\text{data}\{A, B, \dots, Y, Z\}(a, b, \dots, y, z)(0, 1, \dots, 8, 9)\}$

$\text{Key}' = F(\text{key}, x[n])$

//key': Modified Random function

//F(): Cryptographic function

Return random data

RNG produces much more number of combinations of all CAPTCHA characters with different colors. In such a case, a set of characters form a word that is not available in the current using dictionary.

Advantages

- i. User can easily attempt the CAPTCHA screening test.
- ii. The number of used CAPTCHA is not fixed.
- iii. The same character can be represented in different colors for various attempt.

3. Validation

User filled all the necessary details in the currently used web application form and then

allowed to appear for the CAPTCHA screening test. The number of CAPTCHA characters allotted in the screening test can vary from 6 to 8. Below each CAPTCHA character a corresponding number is displayed (as shown in the fig 4). Based on the sorted random values user clicked all the characters and the system checked the clicked character along with the equivalent word in the database. The database maintains an equivalent recognized word along with an equivalent CAPTCHA characters as allotted in the screening test. If the user not able to clicked all the characters based on the sorted order, a new set of CAPTCHA characters will generate after some time period. The fig 4 shows the validation of a user entry character along with the recognized word in the Data base set.

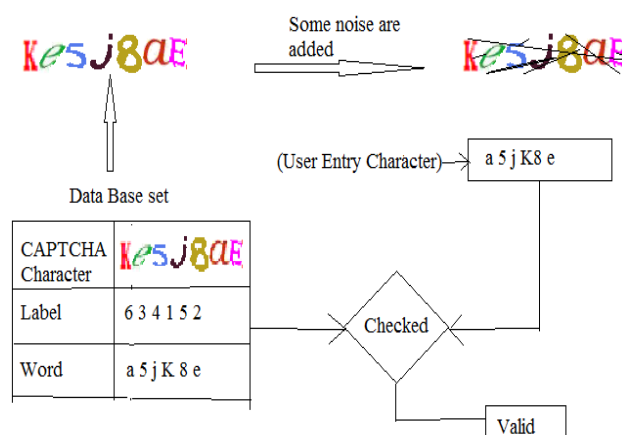


Figure 6: validation of a user Entry test

6. Result and Discussion

We explore the CAPTCHA screening test in the same user entry web form. Fig 5 shows the snapshot for the successful completion of a user entry validation process. The effectiveness of a ROMCC is measured by evaluating its accuracy[17]. An accuracy is defined as the fraction of CAPTCHA that were correctly answered by the human user.

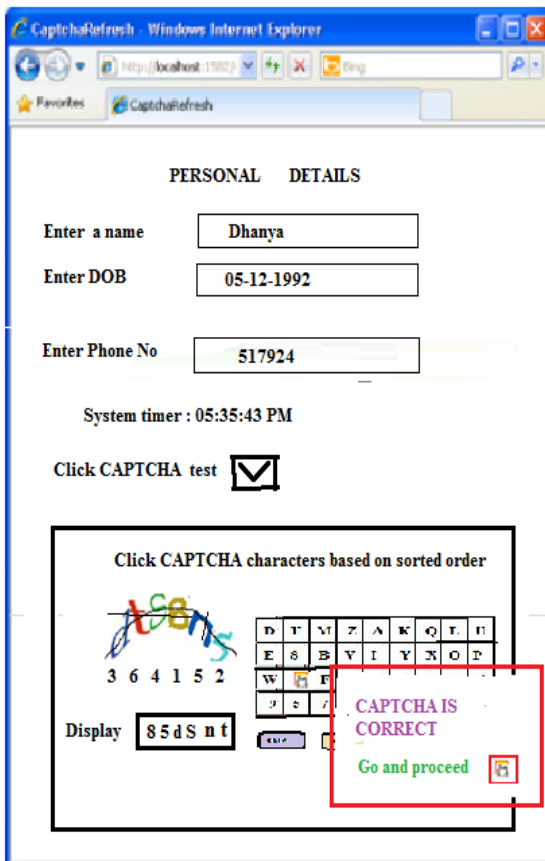


Figure 7: CAPTCHA screening test by the user

Accuracy = Sum/N

Sum: Number of CAPTCHA successfully performed

by the user

n: Number of CAPTCHA test conducted

The accuracy is calculated as follows:

$$\text{Accuracy} \rightarrow g(x) = \sum_{x=1}^{x=m} * \left[\sum_{y=1}^{y=n} a[y] + \sum_{z=1}^{z=n} b[z] \right] s[x], y, z$$

Where y iteration → human user recognize 'n' number of

Table 5: ROMCC performed by the user

CAPTCHA's character

z iteration → human user predicts the random number generator(RNG) i.e based on sorted RNG, user enters the CAPTCHA's character xiteration → m times ,number of sample test conducted

$a[y] = 1$ {if the user predicts all CAPTCHA's character correctly, otherwise $a[j]=0$ }

$b[z] = 1$ {user entered CAPTCHA's character correctly

otherwise $a[j]=0$ }

$$g(x) = \sum_{x=1}^{x=m} * \left[\sum_{y=1}^{y=n} a[y] + \sum_{z=1}^{z=n} b[z] \right] s[x]$$

$$g(x) = \sum_{x=1}^{x=m} * \left[\sum_{y=1}^{y=n} (1) + \sum_{z=1}^{z=n} (1) \right] s[x]$$

$$g(x) = \sum_{x=1}^{x=m} * [(n - 1 + 1) + (n - 1 + 1)](1)$$

Finally, the $g(x) = [(m - 1 + 1)(2n)] = \theta(2nm) = \theta(nm)$.

Thus, the time complexity of Accuracy_CAPTCHA_Entry algorithm is $\theta(nm)$.

An accuracy has reached for a 96% in all the possible combination of CAPTCHAs. For 100 number of test conducted (m=100) using various number of CAPTCHAs and the Table 5 shows that the proposed method has achieved good results rather than other approaches.

CAPTCHA Test			Predict all the ROMCC CAPTCHA character /Human Recognition		
No of character used	Attacks	Total samples	Scenario-1	Scenario-2	Scenario-3
6 to 8	Online, Dictionary, OCR	100	95%	96%	96%

In my proposed methodology, user filled all the sufficient and correct details in the web application form and then allotted for the CAPTCHA screening test. The different colors can use for the same CAPTCHA character in different screening test, this makes the size and pixel count vary for the same CAPTCHA character. In addition to that, some graphical operation has made in the CAPTCHA character, along with some noise (cracks are there in the text box). This leads to the stronger security level for their CAPTCHA refreshment.

7. Conclusion and Future Enhancement

After the successful completion of the user registration form, the system can allocate a CAPTCHA screening test in the web application form. All the CAPTCHA character appears in the screening test in different colors. Thus make the representation of CAPTCHA provide easier for the user to understand, even though the characters collide with others and have some cracks. In order to create the mouse clicked Color CAPTCHA in a more securable way, the CAPTCHA screening test is to be displayed within a limited time period. In future, the user wants to write a Mouse clicked CAPTCHA test within a given time period, and then only sign and enter into the required web area. If the user unable (takes longer time) to finish the CAPTCHA test during the

registration process, a new set of Mouse clicked CAPTCHA will be generated.

References

- [1] Mumtaz M. Ali AL-Mukhtar and Rana Riad K. AL-Taie, "A More Robust Text Based CAPTCHA For Security in Web Applications" *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* Volume 3, Issue 2, March – April 2014 ISSN 22786856
- [2] Varun Ambrose Thomas, karavir Kavr, "Cursor CAPTCHA : captcha mechanism using Mouse cursor", *International journal of computer application*, Vol 67, no 22, Apr 2013
- [3] Ur Rahman Maulana Azad, "Survey on CAPTCHA systems", *Journal of Global Research in Computer Science*, VOL 3, No 6, PP, June, 2012
- [4] Silky Azad & Kiran Jain, "CAPTCHA : Attacks and Weaknesses against OCR Technology", *Global journal of science and technology Neural & Artificial Intelligence*, Vol 13, issue 3, version 1.0, May 2013
- [5] Elie Bursztein, Matthieu Martin, John C. Mitchell, "Text based CAPTCHA Strengths and Weaknesses," *ACM Computer and Communication security (CCS)*, pp. 1-14, 2011
- [6] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, USA, June 16-22, 2003, vol. 1,

- pp.134-141.DOI:
10.1109/CVPR.2003.1211347.
- [7] K Chellapilla, and P Simard, "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)," Advances in Neural Information Processing Systems 17, Neural Information Processing Systems (NIPS), MIT Press, 2004.
- [8] Divya Shanker, Prashant Gupta, Aditya Jaiswal, "Hybrid collage CAPTCHA", International Journal of Scientific & Engineering Research, Volume 4, Issue 1, January-2013 Issn 2229-5518
- [9] Bin B. Zhu, Jeff Yan, Guanbo Bao, Maowei Yang, and Ning Xu "Captcha as Graphical Passwords —A New Security Primitive Based on Hard AI Problems", IEEE Transactions on Information Forensics and Security, VOL. 9, NO. 6, JUNE 2014.
- [10] Sushama Kulkarni, H.S Fadewar, "Mouse Dynamic based CAPTCHA: Brief Review", International journal of advanced research in computer science and software engineering, Vol 5, issue 12, Dec 2015.
- [11] Sushama Kulkarni*, Dr. H. S. Fadewar, "Mouse Dynamics based CAPTCHA: A Brief Review" International Journal of Advanced Research in Computer Science and Software Engineering" Volume 5, Issue 12, December 2015
- [12] Varun Ambrose Thomas, Karanvir Kaur, "Cursor CAPTCHA – Captcha Mechanism using Mouse Cursor", *International Journal of Computer Applications* (0975 – 8887) Volume 67– No.22, April 2013
- [13] Oleg Starostenko, Fernando Uceda Ponga and Vicente Alarcon Aquino, "Breaking text-based Captchas with variable word and character orientation", Elsevier, Pattern Recognition, VOL. 18, No 4, APRIL 2014.
- [14] Chandavale, A.M. Sapkal and R.M. Jalnekar, "A Framework to analyze the security of Text based CAPTCHA", International Journal of Computer Applications, VOL. 1, 2010.
- [15] L. Von Ahn, M. Blum, and J. Langford, "Telling Human and Computers Apart Automatically," in *Communications of the ACM*, vol. 47, February 2004, NO. 2, PP. 57-60.
- [16] Er. Vivek Kumar, Er. Prem Shanker Yadava, "Position based Captcha: Changing place restriction minimize the automatic access" International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, VOL 3, NO 10, OCT 2013
- [17] Maddela Shireesha and V.B. Gaikwad Terna, "Performance Evaluation of CAPTCHA WORD RANKING Algorithm to Break Video CAPTCHA", International Journal of Computer Application, VOL. 75, No. 10, 2013.