International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 2, February 2015, PP 18-24
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Novel Security Method Using Captcha as Graphical Password

Shruti Doiphode, Jagruti Jadhav, Priyanka Shelke, Prof.M.S.Pokale

Department of Computer Engineering, Pune Vidyarthi Griha's College of Engineering and Technology, Pune, India

ABSTRACT

A captcha is acronym for completely automated public turing tests to tell computers and humans apart Captcha is used for security purpose but there are many attacks which can break captchas. This paper proposes carp (captcha as graphical password) technique which is nothing but combination of captcha and a graphical password and it is click-based password. Carp saves from attacks like online guessing attacks relay attacks, shoulder surfing attacks, online dictionary attacks, human guessing attacks etc. This carp technique is based on hard ai problems. This click-based technique requires sha1 and discretize centralization algorithm for better performance

Keywords: Captcha, carp, Discretized centralization algorithm, Graphical Password, SHA1 algorithm.

INTRODUCTION

Captcha is now a standard internet security technique to protect many online services and it also protects from bots at some extent. But now there are many techniques available to break captcha. In this paper, carp technique is introduced which is based on hard ai problems where hard ai means hard to break by intelligent algorithms.

Carp is click-based graphical password. In this technique, a particular sequence of clicks on an image is used to generate password. It is not like any other click-based graphical passwords. Images which are used in carp are nothing but captcha challenges. For every login attempt a new carp image is generated dynamically. At run time it creates an image which is combination of many images [1].

Carps can be form on both text as well as image recognition captcha [2][3]. In text carp, the generated password is a particular sequence of characters like a normal text password but it is not by typing that password by keyboard it is by clicking the right character sequence on carp images.

In order to secure online services carp is an efficient technique than text passwords or graphical password. Because text password is very insecure for user authentication and many attacks are possible on graphical passwords. So it is highly vulnerable because of many attacks like shoulder surfing attacks [4] so it is very difficult for hackers.

RELATED WORK

There are many different techniques a user can be authenticated by a system [4].

TEXT PASSWORD

Text password is nothing but alphanumeric password in which there are upper case letters, lower case letters, numbers and few special symbols can be used. Combination of all these are used to form a string which is a password. This password is very easy to remember .But it is easy for hackers too.

Textual password is normally of 10 characters that means 26 uppercase characters, 26 lowercase characters, 10 digits (0 to 9) and 10 special symbols. So by adding all these total 72 characters are there. By taking this into consideration 72^{10} permutations are possible. Text passwords are very vulnerable to shoulder surfing attacks, online dictionary attacks, human guessing attacks, relay attacks etc.

**Address for correspondence*

jagrutijadhav32@gmail.com

GRAPHICAL PASSWORD

Graphical password is nothing but images. Human brain is better at recalling images than text. So graphical password is better than textual password. Normal resolution of graphical password is 800*600. And for window it is 10 * 10 .Therefore permutations will be 4800^{10} which is hard to break.

Figure1. *Graphical Password*

CAPTCHA

A Captcha is a program that can generate and grade tests that most humans can pass but computer programs or bots cannot pass [6]. Such programs or technique differentiates humans and bots effectively. It is Completely Automated Public Turing test to tell Computers and Humans Apart. It is type of challenge-response test, so bots, computer programs cannot complete such challenges.It distinguishes human users from computers by presenting a challenge .It is standard Internet security technique.

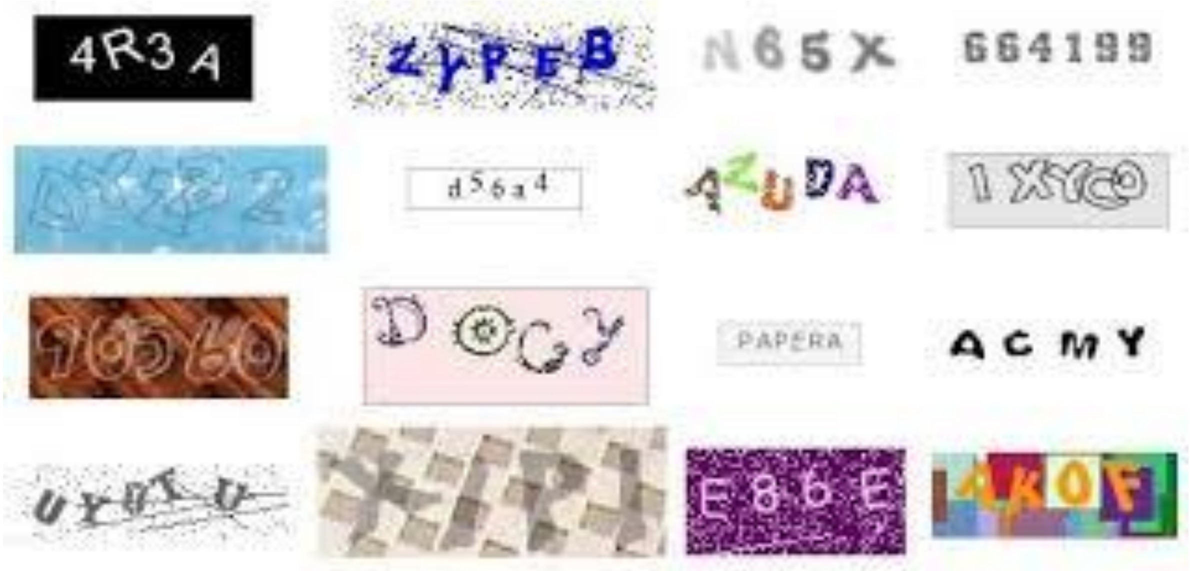


Figure2. *Captcha*

ARCHITECTURE

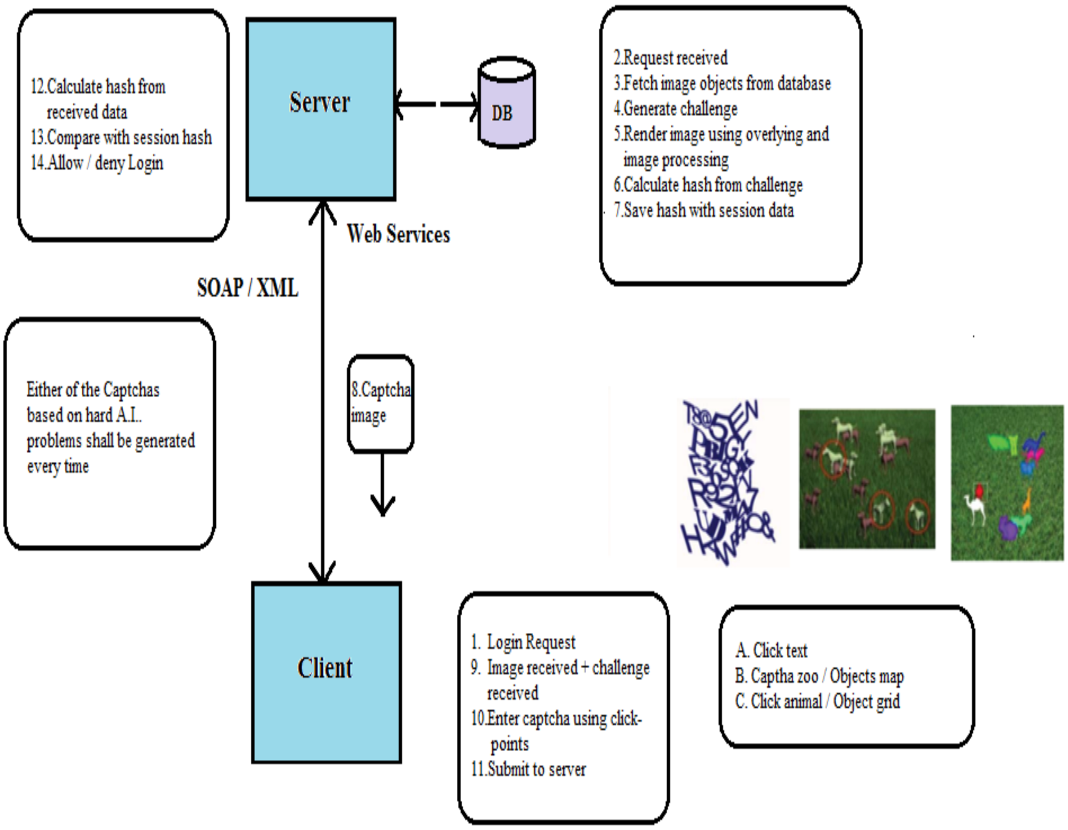

12.Calculate hash from received data
13.Compare with session hash
14.Allow / deny Login
Server
DB
2.Request received
3.Fetch image objects from database
4.Generate challenge
5.Render image using overlying and image processing
6.Calculate hash from challenge
7.Save hash with session data
Web Services
SOAP / XML
Either of the Captchas based on hard A.I.. problems shall be generated every time
8.Captcha image
Client
1. Login Request
9. Image received + challenge received
10.Enter captcha using click-points
11.Submit to server
A. Click text
B. Captha zoo / Objects map
C. Click animal / Object grid

Figure3. *Architecture*

From above architecture there are two possibilities that either user is registered or not that means sign in or sign up. If the user is not registered then user has to create an account by giving username and password. And according to that password, user will get a new Captcha challenge every time. By clicking on correct points user can login. Then Authenticated server receives password of particular account and calculate its hash value using algorithm like SHA-1. Authentication is successful if and only if the two hash values are matched [5].

ALGORITHMS USED

Sha1 Algorithm

SHA-1 is a cryptographic hash function designed by the National Security Agency and published by the NIST as a U.S. Federal Information Processing Standard.SHA1 stands for Secure Hash Algorithm1, which is necessary to ensure the security of the Digital Signature Algorithm (DSA). When, a message of any length < 2^{64} bits is input, the SHA1 produces a 160-bit output called a message digest. The message digest is then input to the DSA, which computes the signature for the message. The SHA1 is called secure because it is designed to be computationally infeasible to recover a message corresponding to the message digest. Any change to the message in transit will, with a very high probability, result in a different message digest, and the signature will fail to verify. The SHA1 is based on principles similar to those used by Professor Ronald L. Rivest of MIT when designing the MD4 message digest algorithm, and is closely modeled after that algorithm.

ALGORITHM FRAMEWORK

Step 1: Append Padding Bits

Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits fewer than an even multiple of 512.

Step 2: Append Length

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

Step 3: Prepare Processing Functions

SHA1 requires 80 processing functions defined as:

f(t;B,C,D) = (B AND C) OR ((NOT B) AND D)   (0 <= t <= 19)

f(t;B,C,D) = B XOR C XOR D   (20 <= t <= 39)

f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)

f(t;B,C,D) = B XOR C XOR D   (60 <= t <= 79)

Step 4: Prepare Processing Constants

SHA1 requires 80 processing constant words defined as:

K(t) = 0x5A827999	(0 <= t <= 19)
K(t) = 0x6ED9EBA1	(20 <= t <= 39)
K(t) = 0x8F1BBCDC	(40 <= t <= 59)
K(t) = 0xCA62C1D6	(60 <= t <= 79)

Step 5: Initialize Buffers

SHA1 requires 160 bits or 5 buffers of words (32 bits):

H0 = 0x67452301

H1 = 0xEFCDAB89

H2 = 0x98BADCFE

H3 = 0x10325476

H4 = 0xC3D2E1F0

Step 6: Processing Message in 512-bit blocks (L blocks in total message)

This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.

Input and predefined functions:

M[1, 2, ..., L]: Blocks of the padded and appended message

f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): 80 Processing Functions

K(0), K(1), ..., K(79): 80 Processing Constant Words

H0, H1, H2, H3, H4, H5: 5 Word buffers with initial values

Example: 'test' => SHA-1 => 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3'

In SHA1 algorithm, the converted part is not reversible. that means 'test' is converted to above string but the conversion of that string to 'test' is not possible that is why SHA1 is very secure algorithm.

Discretized Centralization Algorithm

Discretized centralization algorithm is also known as centered discretization. Discretization is a method which is used in click-based passwords. So that approximately correct entries can be accepted by the system [7]. As user is not so accurate, the user can not click on the same pixel during every login and for that reason this algorithm is very important. It has the great property known as centered tolerance, by which it can accept the approximate values.

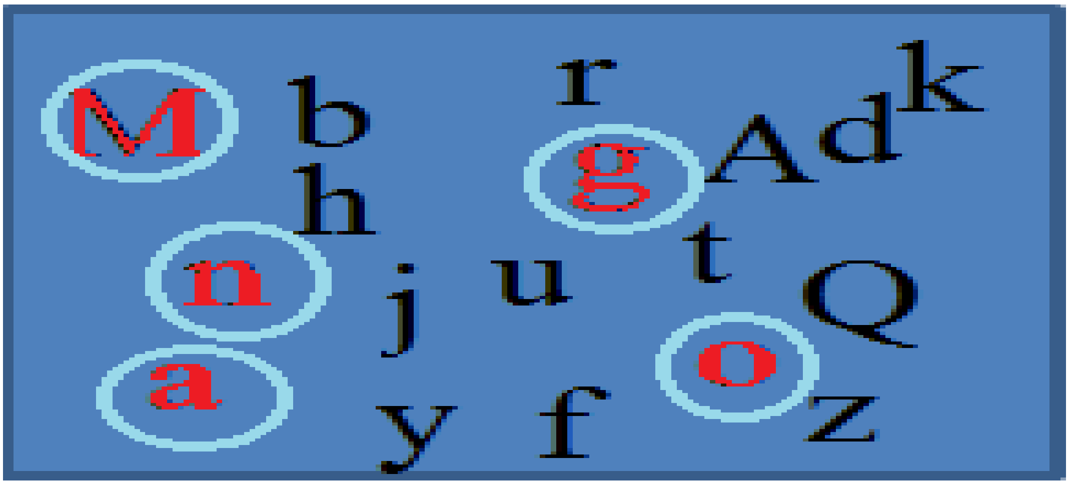
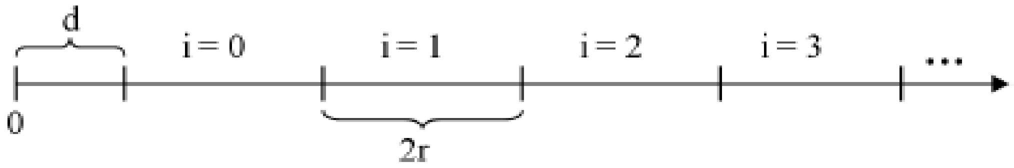


Figure4. *Example that uses discretized centralization*

From above example, suppose the password of user is "Mango" then during sign in user will see the Captcha challenge which is like above figure. 4 where the M, a, n, g, o are present at different locations and there are different alphabets too, and user has to click on the locations of those characters in correct sequence. If user will enter this password in correct sequence then login is successful otherwise login is fail.

For this there are different formulae in discretized centralization algorithm,



$i = [(x-r)/2r]$ (1)

$d=(x-r)\mathrm{mod}2r$ (2)

$i'=[(x'-d)/2r]$ (3)

Where,

i= index

d=displacement

i'=index inverse

Example:

In Figure 4, If the coordinates of M are (50, 50) then

$i = [(x-r)/2r]$

$i = [(50-5)/10] = 45/10 = 4$

$d = (x-r) \bmod 2r$

$d = (50-5) \bmod 10 =5$

From this if user clicks on point (53, 47) then

$i' =[(x'-d)/2r] = (53-5)/10 = 48/10 = 4$ so it is accepted.

And if user clicks (43, 47) then

$i' = [(x' - d)/2r] = [(43 - 5)/10] = 3$ so this point is rejected as the value of i' is not equal to i that is not equal to 4.

MATHEMATICAL MODEL

S = {I, O, F, FF}

Where,

I is input.

Input is nothing but the click-based password by using which a user can login to account.

O is output.

Output is nothing but the authentication result whether yes or no.

F is failure case.

If user enters wrong password then there will be failure case.

Ff is friend function.

Friend functions are read ( ) and write ( )

Input={i1, t, b, s}

Where,

I1 is image set

T is text character set

B is background.

S is schema.

Intermediate = {prng (pseudo random number generator)}

Output = {authentication result (yes or no)}

Failure = {user does not remember password}

Friend function = {read ( ), write ( )}

CONCLUSION

The overall goal of this project is to provide security at the best level. This will improve the performance of online services and prevent from many attacks. Our system proposes carp technique which is nothing but the combination of Captcha and graphical password. Because of this combination it becomes very difficult for hackers to hack the account. And it prevents from the attacks of bots. As, this system generates every time a new Captcha challenge at run time it becomes really difficult to guess the password.

REFERENCES

[1] Bin B. 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

[2] Ragavi. V, Dr. G. Geetha , " CAPTCHA Celebrating its Quattuordecennial – A Complete Reference " IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2, November 2011

[3] Ved Prakash Singh, Preet Pal "Survey of Different Types of CAPTCHA" Ved Prakash Singh et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2242-2245.

[4] T. S. Ravi Kiran, Y. Rama Krishna, "COMBINING CAPTCHA AND GRAPHICAL PASSWORDS FOR USER AUTHENTICATION" IJRIM Volume 2, Issue 4 (April 2012) (ISSN 2231- 4334).

[5] Jayshree Ghorpade, Shamika Mukane, Devika Patil, Dhanashree Poal, Ritesh Prasad, "Novel Method for Graphical Passwords using CAPTCHA," International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4 Issue-5, November 2014

[6] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, a John Langford, " CAPTCHA: Using Hard AI Problems For Security".

[7] Sonia Chiasson, Jayakumar Srinivasan, Robert Biddle, P. C. Van Oorschot," Centered Discretization with Application to Graphical Passwords (full paper)"

AUTHOR'S BIOGRAPHY

Shruti Doiphode is a student pursuing her B.E Degree under Department of Computer Engineering from University of Pune. She is presently working on GUI of project.

Jagruti Jadhav is a student pursuing her B.E. Degree under department of computer Engineering from University of Pune. She is presently working Mathematical part of project.

Priyanka Shelke is a student pursuing her B.E. Degree under Department of Computer Engineering from University of Pune. She is presently working on the algorithms of project.

Prof. M. S. Pokale is an assistant professor under department of Computer Engineering. She is having 9+ years' experience in the field of teaching.