International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Experimental study of CAPTCHA: A Security Primitive

Pawar Sonali[1], Kalyankar Pravin[2]

[1, 2]T.P.C.T.'s College of Engineering,Osmanabad, Solapur-Osmanabad Road, Osmanabad, India

Abstract: ***Captcha as graphical passwords (CaRP) is one of the new security primitive based on hard AI problems which is a novel family of graphical password systems built on Captcha technology. Now a days different applications are employed for users, where only the human interaction is needed. To ensure that the response is generated only by the human one test is conducted at the time of authentication. This type of test is nothing but "challenge-response" test generated by developer. This avoids automated softwares to perform their actions on behalf of the user. Captcha avoids multiple user interactions which are actually not the human. Registration forms of number of email services use this scheme of Captcha to ensure that only humans have an account in their organization.***

Keywords: CAPTCHA, PIX, BONGO, GIMPY, CaRP

1. Introduction

CAPTCHA stands for "Completely Automated Public Turing test to tell Computers and Humans Apart". The term Captcha was coined in 2000 by Luis Von Ahn,Manuel Blum,Nicholas J. Hopper. Textual passwords are used mostly to secure the application from abuse. As new technologies are invented, such passwords can be easily guessed by some automated softwares. Thus, textual passwords are not working or secure for next generation.

Captcha is now a standard Internet security technique to protect online email and other services from being abused by bots. The main goal of Captcha is to put forth a test which is simple and straight forward for any human to answer but for a computer it is almost impossible to solve. Usually this test is conducted at the end of a sign up form while signing up for Gmail or Yahoo account. For example free web based e-mail services allow people to create an account free of charge. Computers are not as intelligent as humans. Machines have lack the ability to process on visual data. This is because Computers lack the "Real intelligence".Captcha makes the use of this thing and provides a visual test to the user or human. It is more easily possible for human to look for an image and find out appropriate pattern from it.In this password scheme at the time of implementation the user is presented with an image which contains some pattern. This pattern contains distorted or randomly stretched characters which only humans should be able to identify.

1.1 Breaking Captcha

The challenge in breaking the Captcha is really a hard task. It is hard because it is impossible to teach a computer to be think like humans, how to process information in a way similar to how humans think. To make the Computers think like humans ,Algorithms with Artificial intelligence will have to be designed when it comes to recognize the patterns in Captcha images. However, there is no universal algorithm that could pass through and break any Captcha system. Thus, each Captcha algorithm must have to be Tackled individually.

2. Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. CAPTCHAs are used to prevent robots from submitting forms and creating accounts, spamming and various other things. In some cases robots can cause some problems. Take for example a robot signing up for thousands of Gmail accounts. While it might not cause much stress on Gmail's servers it would create lots of email accounts that could be used for spamming people. Another case is spammers creating accounts on forums and then spam the forum. CAPTCHAs help prevent robots from using websites and webapps.

CAPTCHA technology has its foundation in an experiment called the **Turing Test**. Alan Turing, sometimes called the father of modern computing, proposed the test as a way to examine whether or not machines can think -- or appear to think -- like humans. The classic test is a game of imitation. In this game, an interrogator asks two participants a series of questions. One of the participants is a machine and the other is a human. The interrogator can't see or hear the participants and has no way of knowing which is which. If the interrogator is unable to figure out which participant is a machine based on the responses, the machine passes the Turing Test.

3. Overview of the Technique

There are different types of Captcha depending on the form in which they are presented to the user. The main point to be considered is the pattern in Captcha, it can be Textual Captcha and Graphical Captcha.

3.1 Text Based Captcha

These types of Captcha are simple to implement. This Captcha presents some queries to the user whose answers are only given by users and not by computers or any machines. Examples of such questions are:

1. What are twenty minus five?
2. What is the third letter in INDIA?
3. Which of Yellow, Thursday and Richard is a color?
4. If yesterday was a Monday, what is today?

Only Humans can answer above types of questions , which ensures that no computer program or any boat access them.
Other type of CAPTCHAs involves text distortions and the user is asked to identify the text hidden, such type of Captcha are called as Text-based Captcha. The various implementations of Text- based captcha are:

3.1.1 *Gimpy and Ez-Gimpy*

Gimpy presents a set of words which belongs to dictionary, and displaying them in a distorted and overlapped manner. Gimpy then asks the users to enter a subset of the words in the image. Only human user is capable of identifying the words correctly, whereas a computer program cannot.

Ez-Gimpy is same as Gimpy Captcha, Whereas Ez – Gimpy randomly picks a single word from a dictionary and applies distortion to the text. The user is then asked to identify the text correctly. These two types are adopted by Yahoo in their signup page.

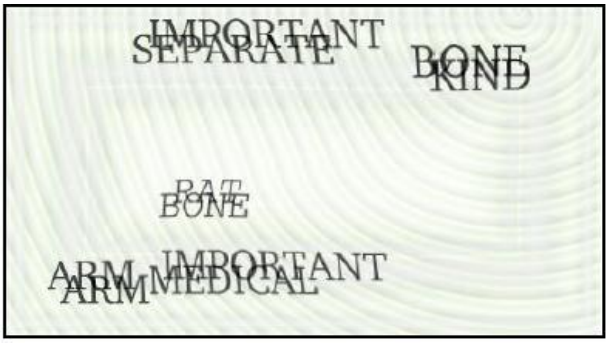

Figure 1: Gimpy Captcha

Figure 2: Ez-Gimpy

3.1.2 *Baffle Text*

This technique overcomes the drawback of Gimpy CAPTCHA because, Gimpy uses dictionary words and hence, clever bots could be designed to check the dictionary for the matching word by brute-force. This is a variation of the Gimpy. This doesn‘t contain dictionary words, but it picks up random alphabets to create a nonsense but pronounceable text. Distortions are then added to this text and the user is challenged to guess the right word.

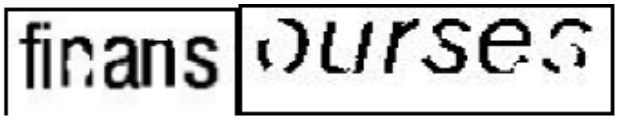

Figure 3: Baffle Text

3.2 *Graphical Captcha*

As its name implies, these type of Captcha includes Graphical data.These Captchas includes some sort of pictures or objects with some properties or characteristics that the user have to guess.

3.2.1 PIX

PIX is a program that has a large database of labelled images. All of these images are pictures of concrete objects (a horse, a table, a house, a flower). The program picks an object at random, finds six images of that object from its database, presents them to the user and then asks the question “what are these pictures of?” Current computer programs should not be able to answer this question, so PIX should be a CAPTCHA.

3.2.2 BONGO

BONGO asks the user to solve a visual pattern recognition problem. It displays two series of blocks, the left and the right. The blocks in the left series differ from those in the right, and the user must find the characteristic that sets them apart.

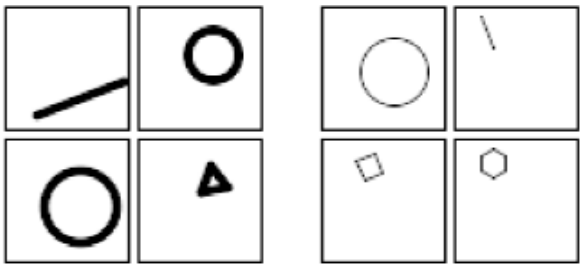

Figure 4: Bongo Captcha

3.3 Audio CAPTCHAs

The Audio Captcha is based on sound. The program picks a word or a sequence of numbers at random, renders the word or the numbers into a sound clip and distorts the sound clip; it then presents the distorted sound clip to the user and asks users to enter its contents. This CAPTCHA is based on the difference in ability between humans and computers in recognizing spoken language. The idea is that a human is able to efficiently disregard the distortion and interpret the characters being read out while software would struggle with the distortion being applied, and need to be effective at speech to text translation in order to be successful.

4. Methodology

Basically CAPTCHA works in the following manner:

1) Create Random Value: Some random string is generated, random values are often hard to guess and predict.
2) Generate an Image: Images are used as these are generally a lot harder to read for computers while being nice and readable to humans. This is also the most important step as simple text in images can be read (and CAPTCHA cracked) quite easily. To make it difficult for them, developers employ different techniques so that the text in the image becomes hard to read for computers. Some create zig-zag lines for background while others twist-and-

turn individual characters in the image. Possibilities are many and new techniques are being developed all the time as crackers are always into finding ways to break them.

3) Store it: The random string generated (which is also in the image) is stored for matching the user input. The easiest way to do so is to use the *Session variables*.
4) Matching: After the above step, the CAPTCHA image is generated and shown on some form which we want to protect from being abused. The users fills in the form along with the CAPTCHA text and submits it. Now we have the following:
   a) All submitted form data.
   b) CAPTCHA string (from form), input by user.
   c) CAPTCHA string (real one, generated by us), from session variable. Session variable is generally used as it can keep stored values across page requests. Here, we needed to preserve stored values from one page (form page) to another (action page-that receives form data).
5) If both match, it's okay otherwise not, in that case we can give the user a message that the CAPTCHA they had entered was wrong and their form could not be submitted. You could also ask them to verify it again.

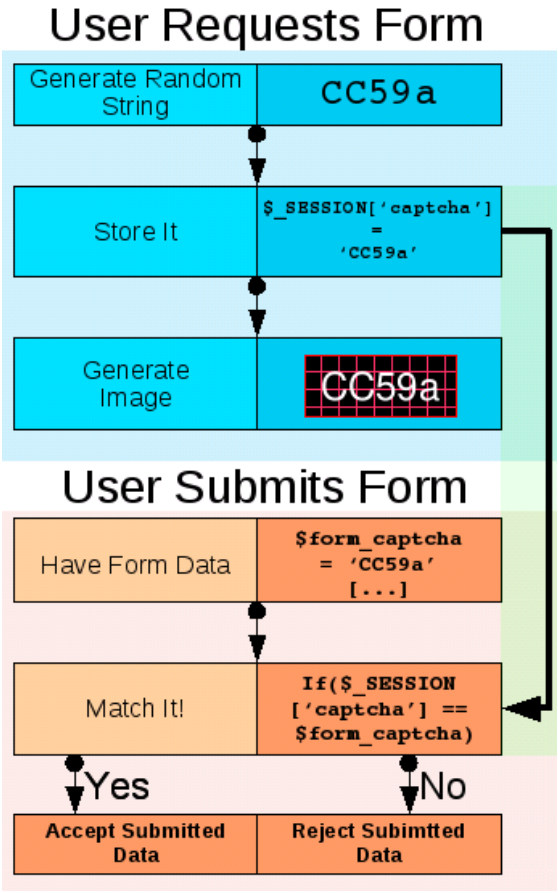

Figure 5: Working of Captcha

4.1 Materials and Methods

A Way to Avoid Guessing Attacks

In a guessing attack, number of trials are attempted to enter an password ,a password guess tested in an unsuccessful trial is determined wrong and excluded from subsequent trials. The number of undetermined password guesses decreases with more trials, leading to a better chance of finding the password. Mathematically, let G be the set of password guesses before any trial, p be the password to find, T denote the probability that p is tested in trial T. Let In be the set of password guesses tested in trials up to (including) Tn. The password guess to be tested in n-th trial Tn is from set $G \backslash In-1$, i.e., the relative complement of $In-1$ in G. If $p \in G$, then we have,

$$p\,(T = p \mid T1 \neq p, \ldots, Tn-1 \neq p) > p(T = p), \quad (1)$$

and

$$In \rightarrow G$$
$$p(T = pT1 \neq p, \ldots, Tn-1 \neq p) \rightarrow 1 \quad \text{with} \quad n \rightarrow |G|, \quad (2)$$

where $|G|$ denotes the cardinality of G. From Eq. (2), the password is always found within $|G|$ trials if it is in G; otherwise G is exhausted after $|G|$ trials. Each trial determines if the tested password guess is the actual password or not, and the trial's result is deterministic.

4.1.1 Usability vs. Security vs. Practicality

Developing a CAPTCHA is always a trade-off between different goals. We identify three orthogonal requirements

Usability refers to the difficulty of solving the CAPTCHA for a human, as well as the time that a human actually needs to find the solution**.**

Security on the other hand, gives a rough guide how difficult it is to find a solution for the computer. Practicality refers to the effort to realize the CAPTCHAs in practice, e.g., if it requires only a standard web browser or can be used easily on a smart phone.

Practicality could also refer to the acceptance on the users' side.

4.2 Constructing CAPTCHAs

4.2.1 Things to know

The first step to create a CAPTCHA is to look at different ways humans and machines process information. Machines follow sets of instructions. If something falls outside the realm of those instructions, the machines aren't able to compensate. A CAPTCHA designer has to take this into account when creating a test. For example, it's easy to build a program that looks at metadata – the information on the Web that's invisible to humans but machines can read. If you create a visual CAPTCHA and the images' metadata includes the solution, your CAPTCHA will be broken in no time. Similarly, it's unwise to build a CAPTCHA that doesn't distort letters and numbers in some way. An undistorted series of characters isn't very secure. Many computer programs can scan an image and recognize simple shapes like letters and numbers. One way to create a CAPTCHA is to pre-determine the images and solutions it will use. This approach requires a database that includes all the CAPTCHA solutions, which can compromise the reliability of the test. If a spammer managed to find a list of all CAPTCHA solutions, he or she could create an application that bombards the CAPTCHA with every possible answer in a brute-force attack. The database would need more than 10,000 possible CAPTCHAs to meet the qualifications of a good CAPTCHA.

The longer he string of characters, the less likely a bot will get lucky. CAPTCHAs take different approaches to distorting words. Some stretch and bend letters in weird ways, as if you're looking at the word through melted glass. Others put the word behind a crosshatched pattern of bars to break up the shape of letters. A few use different colours or a field of dots to achieve the same effect. In the end, the goal is the same: to make it really hard for a computer to figure out what's in the CAPTCHA. Designers can also create puzzles or problems that are easy for humans to solve. Some CAPTCHAs rely on pattern recognition and extrapolation. For example, a CAPTCHA might include series of shapes and ask the user which shape among several choices would logically come next. The problem with this approach is that not all humans are good with these kinds of problems and the success rate for a human user can go below 80 percent.

4.2.2 Implementation

Embeddable CAPTCHAs: The easiest implementation of a CAPTCHA to a Website would be to insert a few lines of CAPTCHA code into the Website's HTML code, from an open source CAPTCHA builder, which will provide the authentication services remotely. Most such services are free. Popular among them is the service provided by reCAPTCHA project.

Custom CAPTCHAs

These are less popular because of the extra work needed to create a secure implementation. Anyway, these are popular among researchers who verify existing CAPTCHAs and suggest alternative implementations. There are advantages in building custom CAPTCHAs:

1) A custom CAPTCHA can fit exactly into the design and theme of your site. It will not look like some alien element that does not belong there.
2) We want to take away the perception of a CAPTCHA as an annoyance, and make it convenient for the user.
3) Because a custom CAPTCHA, unlike the major CAPTCHA mechanisms, obscure you as a target for spammers. Spammers have little interest in cracking a niche implementation.
4) Because we want to learn how they work, so it is best to build one ourselves.

5. Applications

- **Secure Website Registration**.

Many companies like Yahoo!, Microsoft, etc. offer free email services. Most of these services are not secured as they are suffered from attacks called as bots which leads to sign up for thousands of email accounts every minute. It is not sure that account is created by human; Captchas provides the solution to this problem to ensure that only humans can create their accounts and obtain free accounts.

- **Protecting Email Addresses From Scrapers.**

Captchas provides the facility in which you can hide your email address from web scrapers. The idea is to require users to solve a CAPTCHA before showing your email address. It is an effective mechanism to protect your email address and its abuse.

- **Online Polls**

Now a day, many reality programs are taking their decisions depending on the audience choice. For this reason their votes are collected online. As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, some of people found a way to stuff the ballots using programs that voted for one thousands of times. One of them score started growing rapidly. The next day, another person wrote their own program and the poll became a contest between voting one person and another one. Can the result of any online poll be trusted? Not unless the poll ensures that only humans can vote.

- **Dictionary Attacks**

In general password system like text based passwords Dictionary attacks are made to guess the password. CAPTCHAs are used to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.

- **Preventing Search Engine from Bots**

Indexed webpages can found easily, however It is sometimes desirable to keep webpages unindexed to prevent others from finding them easily. Search engines bots can be prevented from reading web pages by an html tag. It may work sometimes but not sure that bots won't read a web page. Search engine bots, usually belong to large companies, respect web pages that don't want to allow them in. In this case CAPTCHAs are needed to guarantee that bots won't enter a web site.

- **Worms and Spam**

CAPTCHAs also offer a plausible solution against email worms and spam. It means that they will accept the emails only if the email is sent bu human and not by any automated software. This idea is now used by many companies.

6. Conclusion

Designing good CAPTCHAs is a tedious business. Text CAPTCHAs achieve a high level of practicality, but very often fall short of providing a good balance between usability and security. Currently, the best choice seems to be reCAPTCHA: it is easy to incorperate, achieves appropriate security and usability levels, and because of the centralized structure behind reCAPTCHA it allows fast migration in response to emerging threats.

Sites with attractive resources and millions of users will always have a need for access control systems that limit widespread abuse. At that level, it is reasonable to employ many concurrent approaches, including audio and visual CAPTCHA, to do so.

6.1 Present Scenario

Conclusion of CAPTCHAs are associated with its usefulness and deciphering ability of HUMAN and BOTS. In present scenario of web world millions of website is using this protocol to minimizes the exploitation of resources. CAPTCHA-based Code Voting is easy to use

- has a good scalability
- Protection against Malware voting fraud
- Security depends on the premise that computers cannot read the CAPTCHAs
- CAPTCHAs may be solved by humans
- Malware is able to cast random votes

6.2 Future of CAPTCHAs

The future of Captcha is also interesting. There's no doubt that image processing software and computers themselves will become more powerful and eventually will be able to automatically decipher today's Captcha images. captcha will be obsoleted by verified id's but may have a future in answering student's homework questions.

References

[1] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surveys*, vol. 44, no. 4, 2012.
[2] (2012, Feb.). *The Science Behind Passfaces* [Online]. Available: http://www.realuser.com/published/ScienceBehindPassfaces.pdf
[3] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The design and analysis of graphical passwords," in *Proc. 8th USENIX Security Symp.*, 1999, pp. 1–15.
[4] H. Tao and C. Adams, "Pass-Go: A proposal to improve the usability of graphical passwords," *Int. J. Netw. Security*, vol. 7, no. 2, pp. 273–292, 2008.
[5] S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical password system," *Int. J. HCI*, vol. 63, pp. 102–127, Jul. 2005. ZHU *et al.*: NEW SECURITY PRIMITIVE BASED ON HARD AI PROBLEMS 903
[6] P. C. van Oorschot and J. Thorpe, "On predictive models and userdrawn graphical passwords," *ACM Trans. Inf. Syst. Security*, vol. 10, no. 4, pp. 1–33, 2008.
[7] K. Golofit, "Click passwords under investigation," in *Proc. ESORICS*, 2007, pp. 343–358.
[8] A. E. Dirik, N. Memon, and J.-C. Birget, "Modeling user choice in the passpoints graphical password scheme," in *Proc. Symp. Usable Privacy Security*, 2007, pp. 20–28.

Author Profile

Sonali Pawar received the B.E. degree in Computer science and Engineering from Terana Public charitable Trust's College of Engineering, Osmanabad in 2011 and pursuing M.E. in Computer science and Engineering from the same institution.