International Journal of Hybrid Information Technology
Vol.9, No.1 (2016), pp. 163-182
http://dx.doi.org/10.14257/ijhit.2016.9.1.15

Simple, Robust & User Friendly CAPTCHA 'InstaCap' for Web Security

Tanvir Ahmed, Kaiser Ahmed Tushar, Sunjida Islam Nova and Md. Mahbubur Rahman

Department of Computer Science & Engineering
Bangladesh University of Business & Technology (BUBT)
Dhaka, Bangladesh
tanvircse10@gmail.com, kaiser.tushar@yahoo.com, sunjida500@gmail.com, mahabub.cse.buet@gmail.com

Abstract

Nowadays security matter is a great challenge in web-based systems and we are using various types of HIP to defend malwares in web. But today's conventional HIP models are able to defend bots but make a user ireful. This paper will focus on a new and simple model of CAPTCHA to defend bot-attack. This model makes a user easily to pass and also amuse them with its simplicity. Considering computational complexity, we are using a thin algorithm with some other technical tricks to make it both robust & simple.

Keywords*: HIP, CAPTCHA, AI, Web-bot, Turing-test, Internet Security, OCR, Web Services, Typoglycemia*

1. Introduction

Due to the wideness of bots, web applications must try to identify when interacting with an actual human user from an automated tool. Automated tools can be used for lots of malicious purposes, like scraping, spamming, and application-level DoS attacks. More exclusively, attackers may use automated tools to post comments in blogs and forums, create fake accounts, salvage mailing lists and advertise products. For this purpose, there used a technique named CAPTCHA as a common security measure using at present against automated attacks [1].

1.1. The CAPTCHA: is an abbreviation for Completely Automated Public Turing Test to Tell Computers and Humans Apart. It is a test, which ensures that it is interacting with a human or a computer. In this test, users have to perform some task like reading digits, words or listening to speech and then tell the users to type on the screen what they saw or heard [15]. The picture or sound is usually distorted in various ways. For humans, it is very easy to pass the task but difficult and time-consuming for bots to complete. In other words, we can say that CAPTCHA is a class of HIPs which has been able to efficiently prevent Web bots to getting unauthorized access to Web services [5].

CAPTCHA is similar to Turing test. But the difference is, here the judge is a computer and the participants are web bots and humans. Computer has to distinguish between them in this test [9].

CAPTCHAs are based on AI problems that cannot be solved by current computer programs and Bots but are solvable by humans. For building a model of CAPTCHA a standard must be followed. In every model, some principles are widely maintained [3].

1.2. CAPTCHA Principles: There are three basic properties that a CAPTCHA must satisfy [9]:

ISSN: 1738-9968 IJHIT

1. It should be easy enough for a user to participate and pass the test.
2. It should be easy for tester machine to generate and grade.
3. It should virtually accept all human users and reject software robots.

1.3. Area of Applications: As the World Wide Web has grown, many web services have been developed. With these developments, some problems have also occurred like web bots automatically register a large number of free accounts and send junk e-mail messages, mass spam blogs are being created and cause other denial of services [9]. To handle this situation CAPTCHAs are used. But it is also used in some other areas as follows:

1. Many programs submit bogus comment on blogs to raise search ranks of websites. CAPTCHA ensures that only human can comment on blogs.
2. Some web pages want to be unindexed by search engines. There is an html tag to stop search engine bots from reading web pages. But tags have not guaranteed that bots will not read the pages. However in order to make sure that bots won't enter web pages, CAPTCHAs are needed.
3. Several companies like Google, Microsoft, and Yahoo! provide free email services. Most of them are suffering from a specific type of attack. Since bots are the program, they can easily sign up for thousands of email accounts in every minute and waste web space. To avoid the misuse of such services, users have to prove that they are human by solving the CAPTCHA.
4. CAPTCHA can be used to prevent dictionary attack in password systems. The idea is simple: prevent a computer from being able to iterate through the complete space of passwords by requiring human to type the passwords.
5. CAPTCHA is now also used to digitalize books.
6. Everyone can write a program to vote for their favorite option thousands of time. To get an accurate and reliable result from online poll, CAPTCHAs are used. A user is allowed to vote only after solving CAPTCHA.
7. Websites often provide advertising for other sites and get paid when user visits the advertised website. To deceive advertisers, bots generate fake visits. As a result, advertisers have to pay for the ads which were not viewed by human. CAPTCHAs are used to solve such problem.

1.4. Types of CAPTCHA: Websites are using different types of CAPTCHAs as a security measurement to distinguish human users from Bots. Here we will discuss about some today's most used CAPTCHAs:

Text-Based CAPTCHA: In text-based CAPTCHA, characters are distorted and connected to prevent recognition by Bots. Security of a text-based CAPTCHA is increased by adding noise and distortion and arranging characters more tightly. Usability is always an important issue in designing a CAPTCHA. Successful text used by Microsoft, Yahoo, and Google use technique that are resistant to segmentation attacks by using random arcs, connected random lines and crowding characters [3]. Some text-based CAPTCHA's are given below:

Figure 1. Some Examples of Text-Based CAPTCHA

Image-based CAPTCHA: In image-based CAPTCHA, the user is provided with a small set of images to name or differentiate or identify anomalies. Here user does not need to read, just simple click is required. Bots find difficulty in identifying images [13]. They cannot recognize human facial expression, colors as efficiently as human does. Various implementations of image-based CAPTCHAs are as follows:

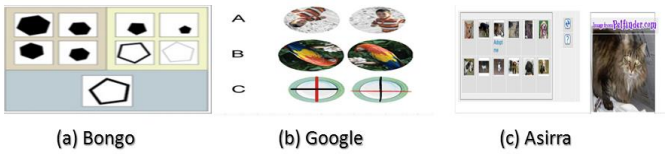


Figure 2. Some Examples of Image-Based CAPTCHA

Multi Model CAPTCHA: It uses both text and image based systems together. Here images with multiple text labels are given. User needs to select the right text label.

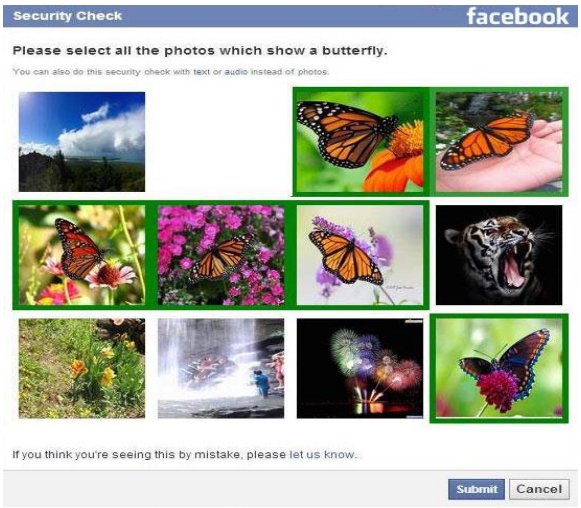


Figure 3. An Example of Multi-model CAPTCHA

Audio-Based CAPTCHA: It takes a random sequence drawn from recordings of simple words or numbers combine them and add some disturbance and racket to it. This soundtrack is played when the user clicks a button given on the web page. The CAPTCHA system then asks the user to enter the words and/or numbers in the recording. Audio CAPTCHAs are more complicated to solve, hard to internationalize and more tough in terms of time and efforts in comparison to text and image CAPTCHAs [8].

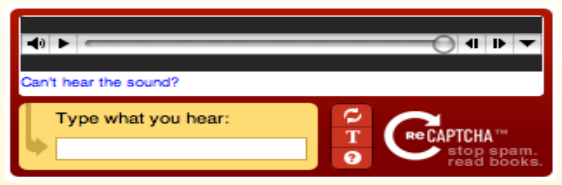


Figure 4. An Example of Audio-Based CAPTCHA

Video-Based CAPTCHA: Here three words (tags) are provided to user which describes a video. If a user's tag belongs to a set of automatically generated position fact tags, then a challenge is passed [11].

Figure 5. An Example of Video-Based CAPTCHA

2. Motivation

There are so many CAPTCHA models are used in real-world, but each of them enhances disturbance of our potential users and also contradict CAPTCHA principles. For optimizing this situation we are motivated to build a new CAPTCHA model which consists:

1. Robustness (difficult to break)
2. Usability (human-friendly)

2.1. Robustness: refers to the terms that a bot cannot break the CAPTCHA model easily. A bot can break a model by various types of bot-attacks as follows [3]:

Dictionary Attack: It is a method of breaking into a password protected computer or server by methodically entering every word in a dictionary as a password. There are two types of dictionary attack [2]:

1. Online Dictionary Attack: In online dictionary attack each password attempt is sent to verifier to check.
2. Offline Dictionary Attack: Here the attacker knows something that allows him to determine the password correctness by himself.

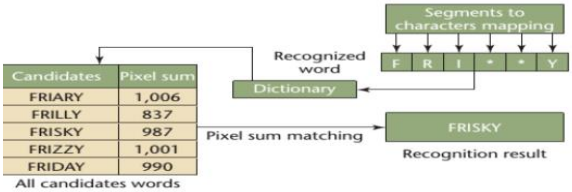


Figure 6. Mechanism of Dictionary Attack

Pixel Count Attack: In this attack, the number of foreground pixel is counted in each segmented character and used to look up in a pre-computed table to determine the character in the segment [4].

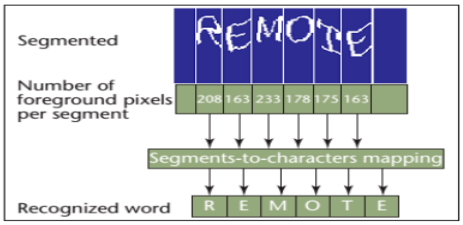


Figure 7. Mechanism of Pixel Count

Denial-of-service (DoS) Attack: It is an incident where a user or institute is deprived of the services of a resource they would normally expect to have [2].

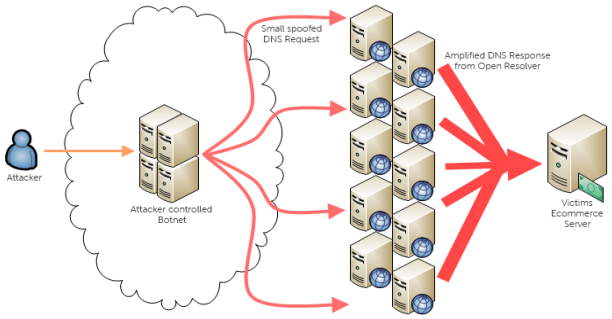
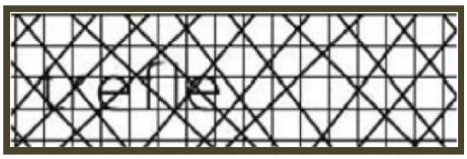


Figure 8. DoS Attack

Brute-force Attack: In brute force attack, it systematically checks all possible keys or password until the correct one is found [4].

2.2. Usability: It concerned with making CAPTCHA tests easy to learn, easy to use, easy to understand and interpret for users. That means a better understandability is also called usability. But if we make the model very much easy, then the robustness becomes less. So usability study can be very much important to design a good CAPTCHA in the field of HIP's [6].

2.3. Robustness VS. Usability: We need to trade-off between robustness & usability to make a complete model. For example we can consider the image:



(a) Better robustness but lack of usability

(b) Better usability but lack of robustness

Figure 9. Robustness vs. Usability

Figure A is a good design with high robustness but it is very difficult for a user to understand *i.e.,* lack of usability. On the other hand, the fig-b has better usability but it can easily break by automated system i.e. lack of robustness [3].

These are not only the reasons but we are also motivated to construct a new model because of many limitations of popularly used conventional CAPTCHA's.

At present those following models are very popular:

1. Gimpy
2. Ez-gimpy
3. Buffle Text
4. Asirra
5. Recaptcha
6. Emoticons
7. Facebook

In previous pages, we highlighted those entire models with a sample image. Now let's find the problems of those CAPTCHA models:

Gimpy: It is a traditional model used since 1997. Basically, it is a kind of text-based model. Here a pair of text is written with morph. The limitations are:

1. Very much easy to break
2. Simple plain-text are used
3. Less robustness

Ez-gimpy: It is an extended version of gimpy model. Here including a text some horizontal & vertical straight lines are also used. But it cannot remove the limitations, as:

1. Also breakable because of using straight lines
2. Enhance noise that annoys a user

Buffle Taxt: Here a buffle type font is used to make CAPTCHA, which have some following limitations:

1. Almost same to plain-text
2. Not effective
3. Now obsolete

Asirra: The abbreviation of it is *Animal Species Image Recognition for Restricting Access***.** It is researched by Microsoft, but still it has some pitfalls:

1. Only use animal image
2. Every species are not known by people, so sometime it may be hard for human too
3. Image processing can break it

Recaptcha: It is mostly used CAPTCHA model since 2009. It is proposed by Google. It started with the text-based model but now it uses images to build CAPTCHA. The pitfalls are:

1. Cracked by OCR
2. Low success rates near about 17.33%
3. Same problems as both ASIRRA & Buffle

EmotIcons: It is a newer version of CAPTCHA which comes from a research [3], but:

1. It takes large size of database, image and dictionary that create huge pressure to a server
2. All emotions are not recognized by human
3. It's engine that generate CAPTCHA and testing results are too much difficult to implement.

Facebook: Today's most popular social site is facebook. It uses multi-modeled CAPTCHA for its security. It cannot use only a specific model every time. But the problems are:

1. Sometimes it shows wrong result, although it was correct
2. Some people are using fake image & name, which can be useless
3. In social network every friend are may not be known, so this kind of question cannot identify the right user
4. This CAPTCHA cannot identify the bot, it just identify authorized user only
5. It cannot use everywhere, because every website doesn't store profile picture or friends information.

Considering those conditions we need to balance them equally. For this reason, we are motivated to design a new model which is both robust & better usability so that a user can easily pass the CAPTCHA test.

3. Proposed Model

3.1. Idea: There are so many techniques used in Turing test. Any kinds of puzzle problem can be considered as a CAPTCHA. Here we proposed a model which is nothing but a fill in the gaps problem. In this model a word is given that have a missing letter. User needs to guess the missing letter that can make a meaningful word. For example we can consider the following Figure:

Figure 10. Pseudo Model of InstaCap

Well, this is our proposed model. It is very much simple for human to pass but also a great challenge to defend automated programs.

3.2. Key-challenges: Although our proposed model is vulnerable for web-bots and that makes some key-challenges:

1. Protecting bots from extraction of features
2. Making a robust CAPTCHA
3. Generating an automated simple CAPTCHA Engine/Program
4. Code implementation

3.3. Improvisations: For making our model unbreakable, we just apply some tricks, as follows:

Increase the quantity of gap: Dictionary attack can simply break our current model but if we use more than one gap, then the situation may change. As followed by the Figure:

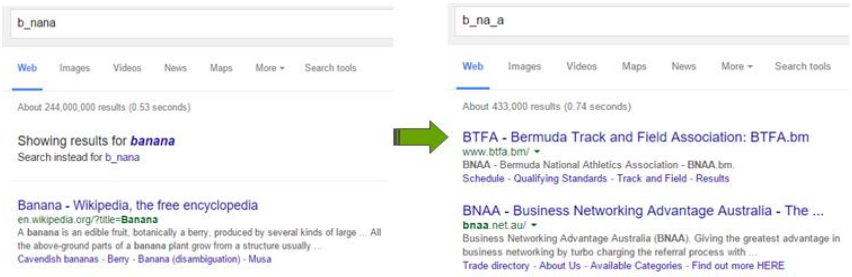


Figure 11. Google-Search cannot Find if Gap Increase

But this approach makes the word difficult to understand, that's why we use an image into our model that refers to suspect object, as shown:

Figure 12. Improvised by Increasing Gap/Blank

Use Gradient Fill Background: If we attached an image into the CAPTCHA, then the pixel-count attack can break our model. So we can fill the background using gradient color:

Figure 13. Improvised by Gradient Background

There are various image processing operations for example, Photoshop have magic tools for cutting/extracting background automatically. If we use the gradient background, then it cannot extract any feature from it, as shown by the Figure:

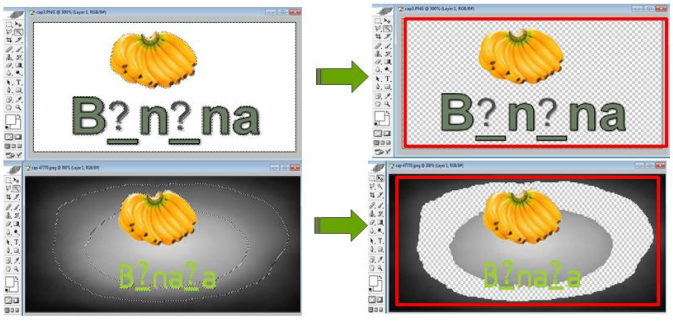


Figure 14. Magic Operation cannot Select Background

Use different types of text-fonts: For increasing the efficiency we can use different types of fonts, as shown by the Figure:

Figure 15. Improvised by using Various Fonts

Hide the Image using Cover: If we hide the CAPTCHA image using cover, where it uncover by hovering mouse into the right position, then it will be more unbreakable. Because an automated program cannot hover mouse like a human, as shown:

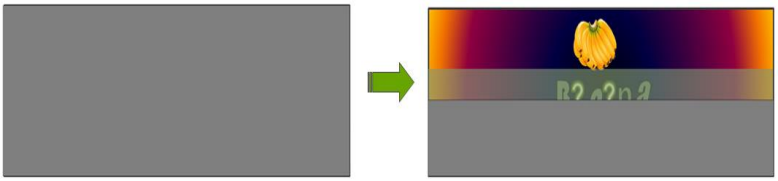


Figure 16. Improvised by Covering Main Image

Mask or Hide the Image Link or Path into the Source Code: If we able to hide image link or path anyhow then the robot cannot identify the image.

Disable Right-click Options: With improvisation of our model visually, we also need to improve inside the code. To protect our code, we can disable all right click options such that save image, view code, save page etc.

3.4. Algorithm: For implementing this model we need to construct an algorithm first. The steps of generating CAPTCHA are given below:

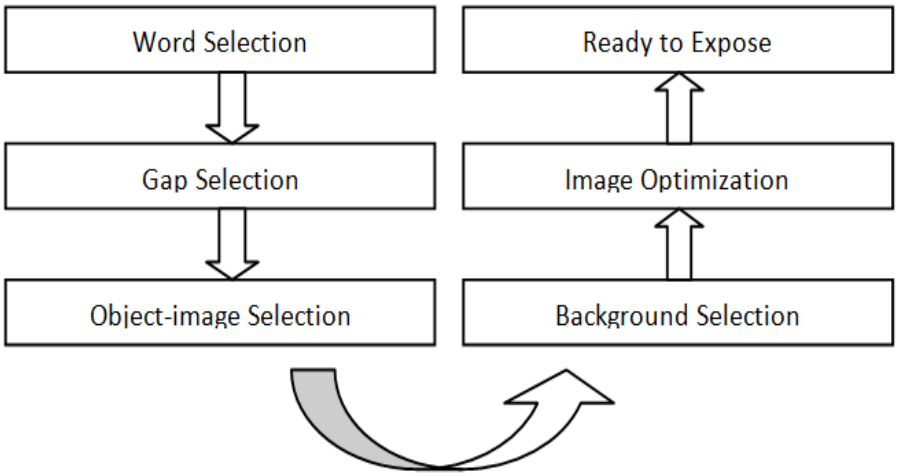


Figure 17. Mechanism of InstaCap

According to those steps, flow-chart would be:

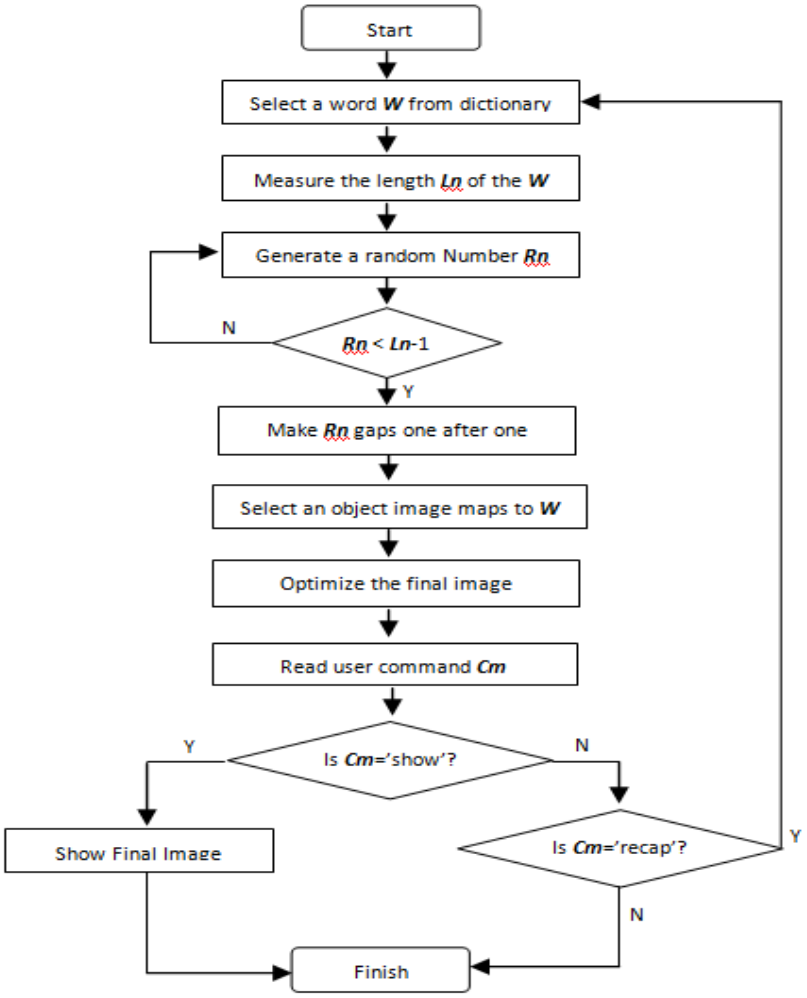


Figure 18. Flow-chart to Implement

3.5. Implementation: As our model will be used in web-based systems, we use php for implementation. The visual techniques are implemented with JavaScript & bootstrap. For word collection, we use database tools like mySql or SQLite to make a large dictionary where the object images are also stored. The main codes are not shown here, but we use it in a project and from the reference of that project we are showing it below:

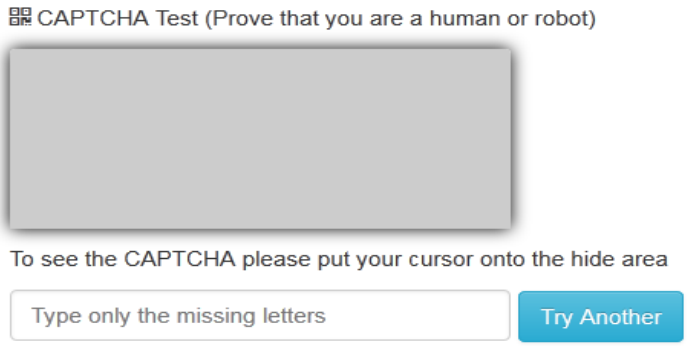


Figure 19. Output of Implementation (Before Hover)

After hovering the mouse, the actual CAPTCHA image uncovers like that:

Figure 20. Output of Implementation (After Hover)

Here you don't need to type full word, just need to type only the missing letters like 'ag'. Until you passed the test, the subscription cannot success. For increasing our security here we make CAPTCHA with all possible combinations for a specific word. We also disable the right-click options using JavaScript so that the code cannot be reveal.

4. Security Analysis

We use CAPTCHA to distinguish bots from human. If bots can access resources of a website, then the result can be devastating. CAPTCHA can detect bots. So hackers first and foremost try will be making the CAPTCHA system to believe that the bots are human, which will help bots to access unauthorized content. If bots cannot be identified properly, then CAPTCHA becomes worthless. So the measurement of the strength of CAPTCHA becomes important. For image-based CAPTCHA's the following steps are followed to break a model [12]:

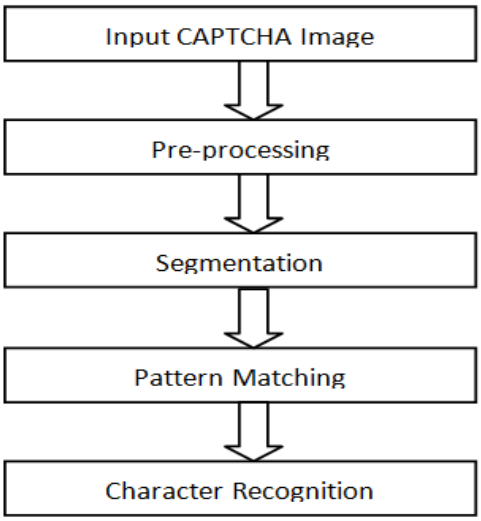


Figure 21. Steps for Breaking Image-based CAPTCHA

Now we will go through each step to analyze security in purpose of our proposed model as follows:

Input CAPTCHA Image: refers to the capture of a CAPTCHA image. At first, bots will detect in which portion/position CAPTCHA is located. It will capture that image to

performing next step called pre-processing. In our model, we hide the image under a cover. When user hover the mouse into correct position, then the cover unveils and user can show real CAPTCHA. When user moves pointer to any other place other than CAPTCHA portion, the CAPTCHA will again hide itself in the cover. As bots are not very efficient to hover on the exact location like a human, it will give us an advantage to secure our CAPTCHA to capture by bots. Instead, bots will capture the cover above it. We also disable our right click mouse option so that manual capturing will be tough. Moreover, we hide image link and musk it so that bots cannot download image by analyzing code [12].

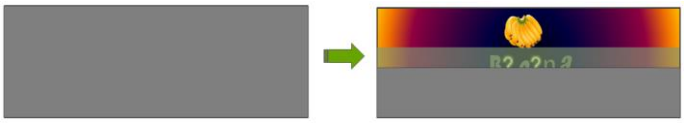


Figure 22. Using Cover Step-1 Cannot Perform

Pre-processing: It process captured image ready to solve. In this phase, it will use a collection of operations which will break the CAPTCHA to binary data. First CAPTCHA image will be converted to gray scale. Then it will convert to binary data where it will be removed from the noise and unwanted bit pattern. But in our implementation, we hide our CAPTCHA under a cover. However if bots can capture our CAPTCHA, then it will use the stated operation for preprocessing. As our CAPTCHA image is filled with gradient background where gradient background consist lots of colors, it will be hard to extract text from background. If several gradient background's color and CAPTCHA icons or text have similar color, then icons and text will be removed with the background removal process and it will fail the process of CAPTCHA solving.

Figure 23. Using Gradient Background Step-2 Cannot Perform

Segmentation: refers to extracting letters from given word or text. Here word will be divided into multiple segments. From these segments bots will get letters like patterns.

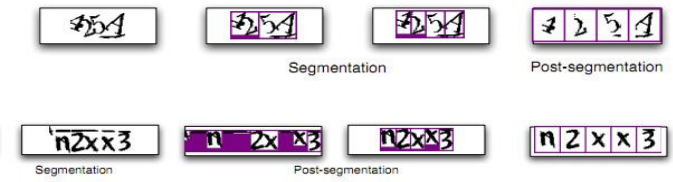


Figure 24. Segmentation Process

For protecting this, we use different types of fonts with unique size for each letter. Anyhow if bots can apply pre-process phase successfully and remove the background from CAPTCHA, then it will use segmentation to extract characters. As fonts and font sizes are different it will be tough to segment each letter as distance from one to another letter will be different. We have also blank fields with question-mark which size is not known for different font sizes and that can be also considered as a letter. So it will be difficult to guess each character at a time [7].

Pattern-Matching & Character-Recognition: compare input CAPTCHA image's letters to stored letters pattern for recognizing actual letter. By analyzing the probability of comparison bots will detect the real letter or alphabet which is given in CAPTCHA image.

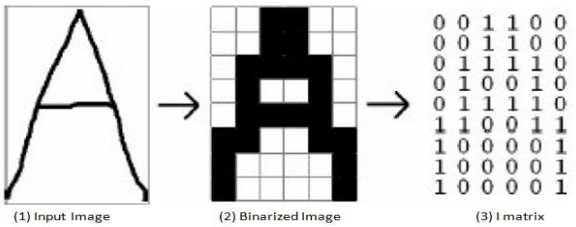


Figure 25. Process of Pattern Matching

If a bot can successfully able to reach before this step, it can easily perform this phase also. However by any way if bots can detect the characters in the CAPTCHA, then it needs to give the required answer. To solve what to put in the blanks bots have to use different types of algorithm as it does not recognize which object's icon we are using. We will use different icons for similar object so bots won't detect what to write by detecting an object's icon. Bots can try brute force or dictionary attack. Thus, it will be harder for bots to detect the correct word and required answer [1].

If bots are using brute force or random guess attack then the probability of a single random guess will be:

$$C = \frac{1}{2*26*n}$$

Where,

- 26 alphabets from a to z & A to Z,
- Probability of an automated BOT entering into a site is $\mathbf{1}/numberofchoices$ and
- $\boldsymbol{n}$ is the number of blanks

Number of blanks $\boldsymbol{n}$	$2 * 26 * n$	Probability **c**
2	104	0.009
3	156	0.006
4	208	0.004

Figure 26. Table of Difficulty Levels of InstaCap

The table shows the result for three different conditions depends on the number of blanks. The design of an **InstaCap** allows administrators to simple tuning of the security level depending on the nature and popularity of the website. However, increasing difficulty may also raise the response time and eventually decrease the success rate of novice potential users.

5. User Studies & Results

We arrange a user study event for our proposed model via online with the help of another project to observe real users feedback. An invitation was announced to university website in order to connect them to investigate the actual view of our model.

5.1. Demographics of Participants: About 100 peoples are participated in our event in range of age from 15 to 39 at 23rd to 26th November, 2015. The Demographics of those samples are depicted by the following table:

	Gender		Age Ranges			
	Male	Female	15-19	20-24	25-29	30-34
N	67	33	8	52	11	29
%	67	33	8	52	11	29

Figure 27. Table of Demographics of Sample

5.2. User Study Layout: In this segment, we are studying the user with following terms:

1. Age
2. Gender
3. Educational background
4. Language proficiency
5. Test InstaCap through a registration
6. Take rating of InstaCap from user

The study took an average of 7 minutes to complete for each participant.

5.3. Usability Study: It is measured by those five qualities:

i) Learnability: How easy is it for users to accomplish basic tasks for the first time they meet the design?

ii) Efficiency: Formerly users have learned the design, how fast can they complete tasks?

iii) Memorability: When users come back to the design after a period of not using it, how simply can they restore proficiency?

iv) ***Accuracy:*** how successfully can a user to pass a challenge? And how simply can they recover from the errors?

v) ***Satisfaction:*** How pleasurable is it to use the design?

We also consider the other components:

1. How was their feeling or response
2. How much time they need to solve
3. How many retry occurred
4. Success or failure

5.4. Accuracy: Accuracy or success rate refers to how successfully a participant can pass the CAPTCHA test. Total 100 participants are applying here. The outcome of success rate is given below by the table:

	Outcome					
	Success	Failure	Retry	Like	Unlike	Avg Success Time(Sec)
N	86	14	11	90	10	36.92
%	86	14	11	90	10	

Figure 28. Table of Overview of user Data

This Table also shows the user's choice about how they interact with this new model of CAPTCHA through like or unlike. The average success time the participant needs are also stated into the table.

5.5. Feedback: After taking an observation of a participant, we also take some feedbacks on different types of CAPTCHA models. We just show them a collection of samples of recently used CAPTCHA models including our model too. Then we ask them to rate the entire models from 1 to 5 on the basis of their choice. After calculating the results, we received a good feedback that they rated our model at high score as shown by the following table:

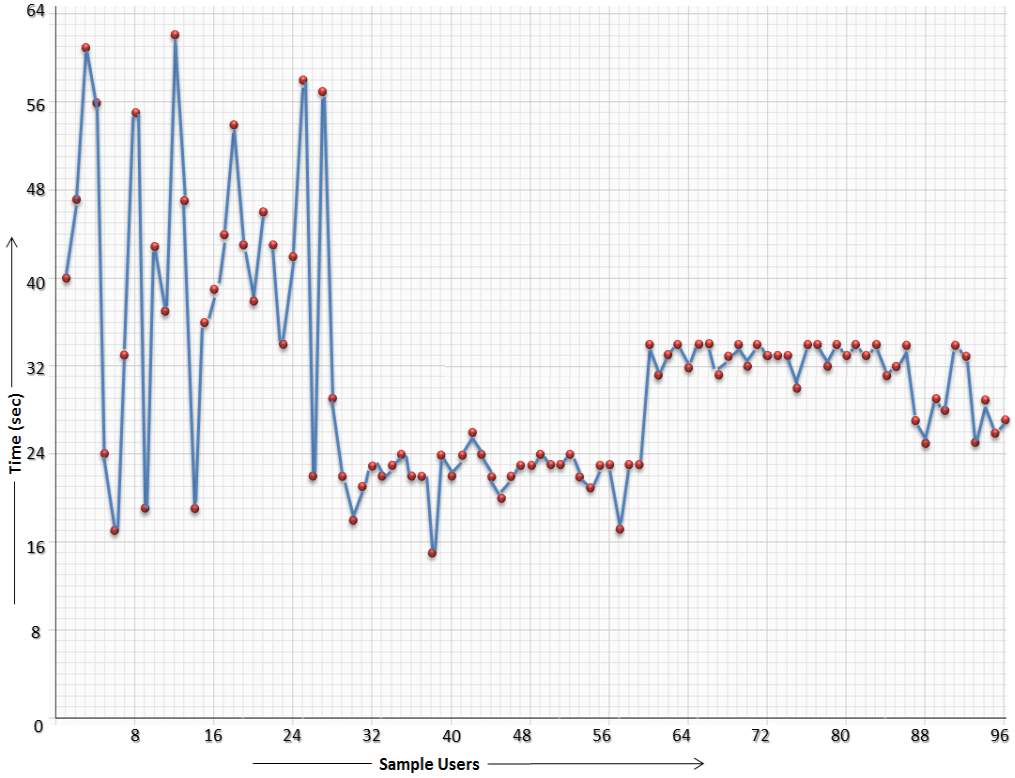


Figure 29. Timing Data Points Distribution of InstaCap

5.6. Security Measure: During the test, we observe most of the participant type full word instead of only the missing alphabets. Some of them are failed to understand the problem correctly. We proposed hints with sample images. But maximum users cooperate with it very positively. On the other hand, we call some hackers & IT Experts to break our model using an automated program. At first they took this very easily but finally they were unable to break our model and our model wins the security test too. This test strongly predicts that this model looks simple for human but unable to break by a bot. So this makes our model robust [8].

Through the hackers we can also measure robustness according to pattern recognition algorithm. For both image & text based CAPTCHA's can be broken by various pattern recognition schemes. We already discuss in Section 4 (Security Analysis) that how to break a CAPTCHA by pattern recognition using segmentation. For performing these phase hackers may need help from some OCR tools. One of is ***img2txt.com***

Figure 30. Submission of Sample Image on Img2txt

After uploading two different images it show following results:

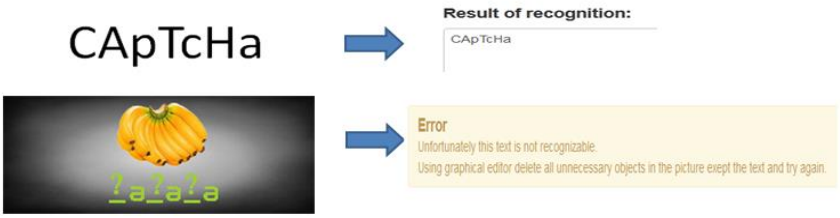


Figure 31. img2txt Cannot Beat Our Model

So, we went to ***img2txt.com*** to extract the written text from the image. But this OCR service cannot break our model. Here we also use ***ABBYY FineReader12*** which is a great tool for OCR support. Here if we apply OCR operation into ours & others model, then it also unable to find the actual word.

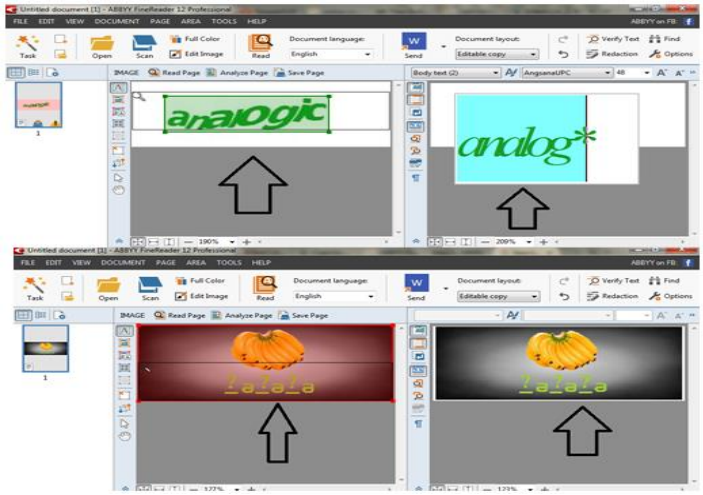


Figure 32. ABBYY Fine Reader-12 Cannot Beat Our Model

So, it cannot recognize any character from our model rather it is showing our input image. We also test our model by comparing with several samples & plain text, which shows the following results, as followed by the table

CAPTCHA Scheme	Number of Challenges					
	All recognized		Partially recognized		Zero recognized	
	Plain text	InstaCap	Plain text	InstaCap	Plain text	InstaCap
Img2txt	26	0	4	0	0	30

FineReader12	28	0	2	2	0	28
%(Img2txt)	86.66	0	13.33	0	0	100
%(ABBYY)	93.33	0	6.66	6.66	0	93.33

Figure 33. Results of Security Measure

Here this table shows that in total of 30 samples are tested to check robustness between a plaintext & InstaCap where both img2txt & ABYY Fine Reader break 0% of all recognized and almost 100% of zero recognized as the table depicted.

So, ultimately OCR tools cannot beat our model & which notify that this model is pretty enough secured.

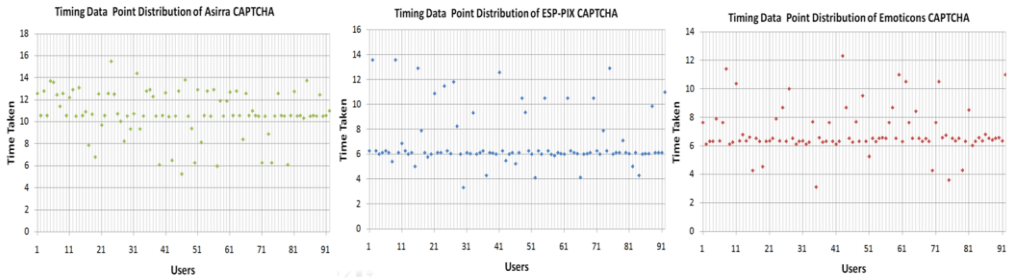


Figure 34. Comparative Timing Data Analysis of other CAPTCHA's [3]

5.7. Discussion: In this study, we use both physical survey and web survey to collect data and also represent that in which CAPTCHA scheme could be applied to provide good service and it could be practically useful.

The intention of our study was able to make claims about general human success rates and examining the feasibility of our proposed CAPTCHA scheme. Therefore, the calculations and predictions in the following sections are truly successful.

6. Advantages & Disadvantage

Every system or model has some good or bad sides. Here we consider both portions. After observing everything of our proposed model following advantages are found:

1. User interactive
2. Easy to solve
3. Less time required to complete
4. Entertain user
5. Create less pressure or load for server
6. Less complexity
7. Robust

On the other hand this model has some limitations. The flipping points are:

1. Use only simple & widely used words
2. Applicable for a particular language
3. Possible to repeat a word because of using random number

7. Future Work

Can you think it is sufficient in future?

Well, this model provides better support for now. But in future, it needs to be more effective to adopting in new technology. In future we can expand this model using various techniques as shown:

Use Latest Technology: To improve this model we can use latest technology like 3D based text, drag and drop option to fill the blank areas etc.

Use Strong Typoglycemia: There is a technique of puzzling human to change the position of the letter into a word that also called typoglycemia. For example consider the following sentence:

"I am a Stuednt"

Yes, you read it well but the student spelling has a mistake. So we can use this technique in future.

Morph or Overlap Letters: If we place the letter overlap with one another or morph it like that:

Figure 35. Further Improvisations

8. Conclusions

For web security CAPTCHA is a great tool, but to build a robust & secured CAPTCHA model we need to know how easily it can be break or how a robot can be detected. There are so many methods for detection of web-bots like ***BotGraph*** and also some other methods for breaking CAPTCHA like ***Projection Segmentation***. According to those matters, better design can be implemented with some extension in ***HTML*** or ***javaScript***.

In this paper, we described our implementations in details and presented performance development plans. These frameworks have become increasingly popular for processing CAPTCHA images and we believe our experience will be helpful to a wide category of applications for constructing and analyzing more secured HIP in the sector of web-security.

References

[1] L. V. Ahn, M. Blum, N. Hopper and J. Langford, "CAPTCHA: Using Hard AI Problems for Security", Advances in Cryptology, Eurocrypt, **(2003)**, pp. 294-311.
[2] K. Chellapilla and P. Simard, "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)", in L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems, MIT Press, vol. 17, pp. 265-272.
[3] M. T. Nayeem, S. H. Mukta, S. Ahmed and M. Rahman, "Use of human cognition in HIP Design via EmotIcons to defend BOT Attacks".
[4] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA", in Computer Vision and Pattern Recognition (CVPR), **(2003)**.
[5] K. Chellapilla, K. Larson, P. Simard and M. Czerwinski, "Designing Human Friendly Human Interaction Proofs (HIPs)", CHI, **(2005)**.
[6] J. Yan and A. S. E. Ahmad, "Usability of CAPTCHAs or Usability issue in CAPTCHA design", In Symposium on Usable Privacy and Security (SOUPS), **(2008)**.
[7] H. S. Baird, M. A. Moll and S. Y. Wang, "A highly legible captcha that resists segmentation attacks", Proc. of Second Int'l Workshop on Human Interactive Proofs (HIP'05), ed. by HS Baird and DP Lopresti, Springer-Verlag. LNCS, Bethlehem, PA, USA, vol. 3517, **(2005)**.
[8] "CAPTCHAs: The good, bad & ugly by Paul, Mani, Lion", Robert On the Security of reCAPTCHA by Benjamin Mild, **(2010)**.
[9] H. S. Baird and J. L. Bentley, "Implicit CAPTCHAs", Proc., SPIE/IS&T Conf. on Document Recognition and Retrieval XII (DR&R), San Jose, CA, **(2005)** January.
[10] "Enhanced CAPTCHAs: Using Animation to Tell Humans And Computers Apart by Elias Athanasopoulos and Spiros Antonatos", In Proceedings of the 10th IFIP Open Conference on Communications and Multimedia Security (CMS), **(2006)** October.

[11] Reduced process thinking algorithm for CAPTCHA strength measurement by Prof. A. A. Chandavale & Prof. A. M. Sapkal.
[12] "What's Up CAPTCHA? A CAPTCHA Based on Image Orientation Rich Gossweiler", Maryam Kamvar, Shumeet Baluja paper at WWW 2009, International World Wide Web Conference.
[13] J.-S. Cui, J.-T. Mei, X. Wang, D. Zhang and W.-Z. Zhang, "A CAPTCHA Implementation Based on 3D Animation", In: International Conference on Multimedia Information Networking and Security, MINES, vol. 2, **(2009)**, pp. 179–182.
[14] A. Turing, "Computing machinery and intelligence", Mind, vol. 49, **(1950)**, pp. 433–460.

Authors

Tanvir Ahmed, He is a final year student of B.Sc. in CSE from Bangladesh University of Business & Technology (BUBT), one of the topmost private university of Bangladesh. He is a talented web developer cum SQA of a well known software company named "Projukti Corporation". Now he is completing his final year project & thesis. His research interests are Internet Security, Data mining, Web database, Artificial Intelligence & Image processing.

Kaiser Ahmed Tushar, He is a final year student of B.Sc. in CSE from Bangladesh University of Business & Technology (BUBT), a reputed university of Bangladesh. He has recently involved in research. His research interests are Internet Security, Database, Data Mining, and Artificial Intelligence. He is now working with complexity analysis of algorithms.

Sunjida Islam Nova, She is a final year student of B.Sc. in CSE from Bangladesh University of Business & Technology (BUBT), a reputed university of Bangladesh. She has recently involved in research. Her research interests are Internet Security, Neural Network, Data Mining, Artificial Intelligence, Graphics and Robotics. She is now working with estimation and decision making model of Neural Network.

Md. Mahbubur Rahman, He has been lecturing in CSE since mid of 2011, he received his B.Sc.Engg. in CSE from Patuakhali Science and Technology University in 2011 and M.Sc. Engg. in CSE at Bangladesh University of Engineering and Technology(BUET), Bangladesh. He is now serving one of the top most private Universities in Bangladesh named Bangladesh University of Business and Technology (BUBT). His research interests are Digital Forensics, Secure and Trustworthy Computing, Data mining, Graph theory, Neural Network.