Security Analysis of CAPTCHA

Anjali Avinash Chandavale[1] and A. Sapkal[2]

[1] Member IEEE
[2] LMIETE
anjali.chandavale@mitpune.edu.in,
ams.extc@coep.ac.in

Abstract. CAPTCHA stands for Completely Automated Public Turing test to distinguish Computers and Humans apart. CAPTCHA is a program which can generate and grade the tests that it itself cannot pass. The security aspect of CAPTCHA should be such that none of the computer program should be able to pass the tests generated by it even if the knowledge of the exact working of the CAPTCHA is known. The effectiveness of CAPTCHA of a given strength is determined by how frequently the guesses of CAPTCHA can be tested by an attacker. This paper proposes a simple and uniform framework for the assessment of security and usability of CAPTCHA that arbitrary compositions of security measures can provide". In this sentence instead of "a simple and uniform framework", use "parameters". This paper proposes parameters for the assessment of security and usability of CAPTCHA that arbitrary compositions of security measures can provide.The pre-processing attack on targeted CAPTCHA is demonstrated having success rate of approximately 97% which in turn helps to build more robust and human friendly CAPTCHA. The universal structure for segmentation attack is framed to analyze security of CAPTCHA.

Keywords: Security, Strength, CAPTCHA.

1 Introduction

In general the security may be defined as the sense of protection from hostile actions. Another way of defining security can be the degree of protection against danger, damage, loss, and crime. In modern day world traditional approach towards security fails to muddle through the ever increasing complexity of security breaches. As we enter into digital world we can no more rely completely on physical aspect of security. Like, if a computer system connected to a network, gets infected with some malicious code. It can cause serious damage to data on memory or might monitor the system user even when perpetrator(s) has no physical access to machine. Also a copyrighted material can get stolen even though original copy is still intact, with nothing actually stolen, and just got copied. Such kind of security breaches cause loss of billions of dollars, identity theft etc. A single isolated computer is less susceptible to be victim of malicious activities in comparison to systems connected to a network such as intranet or internet. In today's age of WEB, more and more people are relying on internet for online information exchange, e- commerce and hence the network security becomes very important. Some of the key issues of network security are

- Privacy
- Authentication

S.M. Thampi et al. (Eds.): SNDS 2012, CCIS 335, pp. 97–109, 2012.

- Authorization

Encryption and Decryption of information

To protect from network threats, security measures are exercised at various levels of computer. Network security starts with authenticating the user, commonly with a username and a password. Since this requires an elaboration authenticating the user name i.e. the password, which is something the user has the knowledge of. This is sometimes termed one-factor authentication. With two-factor authentication, something the user possess is also used (e.g. a security token or 'dongle', an ATM card, or a mobile phone); and with three-factor authentication, a biometric feature of the user is used (e.g. a fingerprint or retinal scan). CAPTCHA systems are widely used to protect various Internet services or applications from unauthorized access of robots or other types of automatic attacks. CAPTCHA stands for Completely Automated Public Turing test to distinguish Computers and Humans apart. CAPTCHA is a program which can generate and grade the tests that it itself cannot pass [3]. The security aspect of CAPTCHA should be such that no computer program should be able to pass the tests generated by it even if the knowledge of the exact working of the CAPTCHA is known. This type of security systems are also called "reverse Turing tests" and used extensively, such as in blogs to prevent spam comments, in forums to stop multiple postings, in email service registration to prevent multiple accounts creation and so on. The role of a CAPTCHA is to make the difference between a bot and a human, through the validation test which can be easily be understood by humans, and nearly impossible to understand by robots. This principle of working of a CAPTCHA is a theoretical view towards understanding the capabilities of robots (artificial intelligence). In practice, many CAPTCHA systems, without having the test generation algorithm, were made public and were broken by researchers as mentioned in [5] [6]. Most of the CAPTCHA systems require the user to type some letters or numbers dynamically generated as a picture by the server. Depending on the directives used in the generation of an algorithm, the letters are rendered in various ways with different types of clutters in background, having the role to make the optical character recognition more difficult for robots. To quote an example, the characters could be rotated, distorted and scaled with different types of background clutters such as Crisscrossing straight lines and arcs, background textures, and meshes in foreground and background colors. Over the past few years, there have been some researches on character recognition dealing with CAPTCHA characters [1]. They emphasized on creating different techniques for different aspects of the problem. Some of the researches on CAPTCHA characters focused on segmentation. Also, most of the methods presented were specific to their own set of problems, but cannot be generalized for common usage. In our research, we use different kinds of CAPTCHA as our subject of experiment and try to analyze security aspect of CAPTCHA. The said paper is organized as follows. Section 2 mentions our simple framework. This includes the parameters to measure the strength of a CAPTCHA for different social web sites. Section 3 focuses on architecture overview to analyze CAPTCHA. Section 4 gives experimental results and section 5 gives the conclusion of the research paper.

2 The Parameters to Measure Strength of CAPTCHA

The strength of CAPTCHA is its effectiveness in resisting and guessing the attacks of BOTS. [10] Specifically, it estimates how many trials an attacker, who does not have direct access to the CAPTCHA, would need on an average to correctly guess it. The security aspect of CAPTCHA should be such that no computer program should be able to pass the tests generated by it even if the knowledge of the exact working of the CAPTCHA is known. The main constraints encountered by most of the CAPTCHAs are:

1. Readable: The CAPTCHA must be easily legible and should be possible to be decoded by humans.
2. Ungues sable: The CAPTCHA message cannot be guessed at random with any real confidence.
3. Order-able: Characters are read left to right, top to bottom (exceptions could include Hebrew or Arabic CAPTCHAs). If a CAPTCHA is readable, its character ordering should be apparent.

These constraints are important because difficult CAPTCHAs can dissuade potential customers, which is not the intent of using CAPTCHAs. Typically, the basic task that a CAPTCHA imposes to users is intuitive, easy to understand and easy to remember. Thus, CAPTCHA has a relatively good learning ability. The nature of CAPTCHAs determines the parameters applicable to address the level of efficiency, errors and satisfaction:

1. Accuracy: How accurately can a user pass a CAPTCHA challenge? For example, how many times he/she has to try in order to pass a test.
2. Response time: How long does it take for a user to pass the test?
3. Perceived difficulty/satisfaction of using a scheme: How difficult to use do people perceive a CAPTCHA is. Are users subjectively satisfied and would they be willing to use such a scheme?

This set of parameters can be a key for quantitatively evaluating the strength of CAPTCHAs. However, this set offers partial guidance on how to improve accuracy, response time or difficulty/satisfaction. Instead, we propose the following parameters for measuring the strength of CAPTCHA which in turn will be helpful to analyze its security.

1. Noise: This examines the form and amount of noise employed in CAPTCHA.
2. Characters: This dimension examines contents embedded in CAPTCHA challenges (or tests) and their impact on its strength. For example, how should the content be organized, and whether the content is appropriate?
3. Response Time/Speed: Duration it take for a user to pass the test. With these parameters strength of CAPTCHA can be measured and thus in turn will help to build more robust CAPTCHA, which is resistant to attack. [11].

2.1 Noise

Noise is an important parameter from security aspect of CAPTCHAs, since it is difficult or impossible for human users to recognize over distorted characters. To cope with this problems caused by distortion, a system will have to allow multiple attempts for each user. Typically a new challenge is used for each attempt. This will not only annoy the users, but also lowers the security of the system by a factor of the number of allowed attempts. The following section describes variations in amount and type of Noise observed in different social web sites like MSN, Google, Badongo, Government services such as Indian Railways etc.

2.1.1 The Use of Color

Badongo uses colored lines to distort the image, and Youtube uses colored blocks; whereas RapidShare uses smaller colored characters as image noise to increase the security. Generally in user interfaces color is extensively used. Using color has also been common in CAPTCHAs, mainly for the following reasons [8]

- Color is a strong attention-getting mechanism.
- Color can provide variation to fit different user preferences [12].
- Color is appealing and can make CAPTCHA challenges interesting.
- Color can facilitate recognition, comprehension and positive effect.
- Color can make CAPTCHA images compatible with the color of web pages and make them look less intrusive [12].

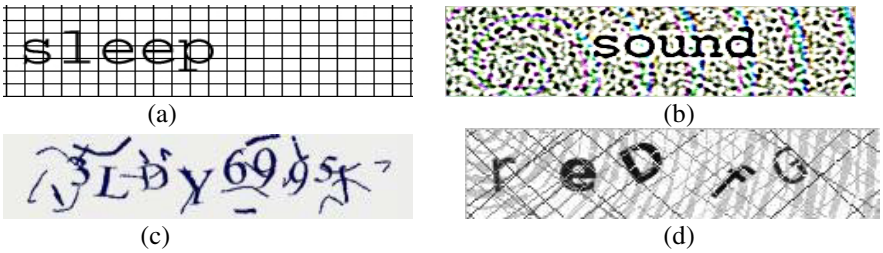
In addition, color schemes might also be expected to work as an additional defense against OCR software attacks in some schemes. Since typically OCR software performs poorly in recognizing texts in color images, particularly they do not do well in segmenting color images. However, we have seen many CAPTCHAs, (refer to Fig. 1) in which the use of color is not effective in context of the concerned security. It has caused negative impact on security, or is problematic in terms of both usability and security. To make challenge images appear to be interesting; some CAPTCHAs generate images in which adjacent characters have distinct colors.

(a) (b)

Fig. 1. a) Google CAPTCHA b) Hollow CAPTCHA (CAPTCHA images where color is used as noise)

2.1.2 Clutters

Crisscrossing straight lines and arcs, background textures, and meshes in foreground and background colors are common examples of clutter used in CAPTCHAs. The representatives of them are the MSN and Yahoo systems, which are used as the basis for the main discussion as shown in Fig. 2



(a) (b)

(c) (d)

Fig. 2. a) Ez-Gimpy (i.e. mesh) CAPTCHA b) Dotted CAPTCHA c) MSN Hotmail d)Digg CAPTCHA

2.1.3 Unwanted/Confused Characters:

Distortion often creates ambiguous characters, which results the inability of users to recognize such characters. In certain cases of characters this distortion may lead to confusion in recognition of characters. Enlisted below are few characters which we studied and found to be resulting into confusion among readers.

Letter vs digits: hard to tell distorted O from 0, 6 from G and b, 5 from S/s, 2 from Z/z, 1 from l.

- Letter vs letters: Under some distortion, “vv” can resemble “w”; “cl” can resemble “d”; “nn” can could resemble “m”; “rn” can resemble “m” ; “rm” can resemble “nn”; “cm” can resemble “an”. Fig. 3b shows some such confusing example that we observed in the Google CAPTCHA (used for its Gmail service). We observed that about 6% of challenges generated by this Google scheme contained such characters.
- Characters vs clutters: In CAPTCHAs such as the MSN schemes, random arcs are introduced as clutters. Confusion between arcs and characters is often observed in this Microsoft scheme. For example, it is difficult to tell an arc from characters such as ‘J’, ‘7’ and ‘L’. In particular, the confusion between an arc and ‘J’ was observed regularly in this scheme.
- Check board characters: The characters are organized in chess board form (as shown in Fig. 3a) which annoys the user.
- Note: characters that look similar in one typeface can look differently in another typeface.

(a) (b)

Fig. 3. a) BOTdetect (Chess board) CAPTCHA b) Confusing characters (starting character is cl or d)

2.2 Characters Used in CAPTCHA

The choice of content materials used in each CAPTCHA challenge can also have significant impact on security.

2.2.1 Character Set

The size of the character set used in a CAPTCHA matters for security. It specifies whether TBC image contains only digits, only alphabets or combination of both. Typically, the larger the character set, the higher resistance to random guessing attacks each challenge can have. However, a larger character set can also imply a higher number of characters that look similar after distortion, causing confusion.

2.2.2 String Length

If both the character set size and the string length are small, random guessing would have a high chance of passing the CAPTCHA. Typically, the longer the string is used in a challenge, the more secure is the result. For example, assume that the state of the art techniques can achieve an individual character recognition rate of r (<1), the chance of recognizing the whole challenge of n characters can be r* n, which decreases as n grows. Whether the length of strings used in a scheme is predictable or not, is again a design issue. Some schemes choose to use a fixed length. For example, in the MSN scheme, each challenge uses eight characters. In some other schemes such as Google's CAPTCHA, the string length is variable, i.e. each challenge uses a different number of characters and the string length for each challenge is unpredictable. This design issue turns out to have implications on both security and usability. For example, the use of a fixed string length in the MSN scheme has a negative impact on its security. The knowledge as to how many characters can be expected in a challenge, can be used for locating connected characters and estimating the number of such characters in the challenge, which is a crucial step in segmentation attack on the MSN scheme [7].

2.2.3 Recognition Rate

The number of characters recognized by bot (automated software program)/system correctly.

2.3 Response Time

How long does it take for a bot to pass the test? One way to judge the strength of a CAPTCHA is to estimate the time and computing power required for its cracking.

3 Attack on CAPTCHA to Determine Strength Measurement Parameters

While focusing towards the task of attacking a CAPTCHA, we observed three procedures namely preprocessing, segmentation and recognition as shown in fig. 4. The preprocessing procedure removes different types of noise (mentioned in section 2), and thus determines the type and amount of noise present in an image. The length of characters in CAPTCHA image is determined by the segmentation procedure which requires identification of the correct positions for each character where as recognition rate is determined by character recognition technique. The recognition procedure identifies which character is in each position. Lastly once the CAPTCHA is broken in

the sense the system gets success in correctly guessing the characters embedded in CAPTCHA image, response time is calculated. In recent research, [9] shows "segmentation" is a much more difficult problem than "recognition" since machine learning algorithms can efficiently solve the recognition problem, but currently we do not have any effective general algorithm to solve the segmentation problem caused by these added clutters. Therefore, this paper proposes an efficient preprocessing algorithm for attacking CAPTCHAs which cleans image to enable the OCR to recognize.

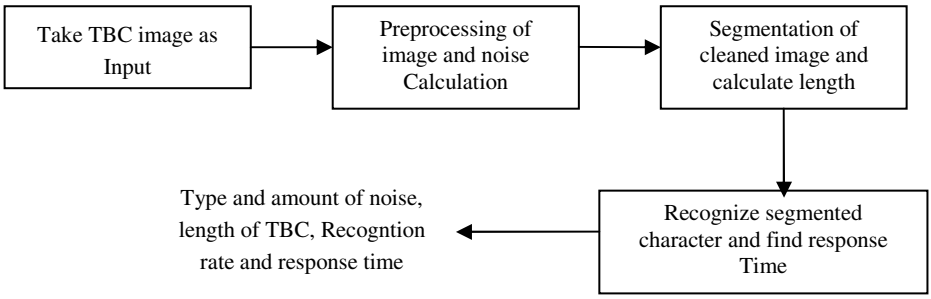


Fig. 4. Procedure for attacking CAPTCHAs

3.1 Implementing Preprocessing Attack

The preprocessing attack implemented in this paper is based on flow chart shown in Fig. 6. CAPTCHA image contains many colors and to work on each of them is quite difficult, so initially it is converted into grey scale using eqn 1 which helps only to work on 256 intensity values. Binary images are often produced by thresholding a grey scale or color image, in order to separate an object in the image from the background. The color of the object (usually white) is referred to as the foreground color.

$$\text{Grey Color} = (0.299*R+0.58*G+0.114*B) \quad (1)$$

The rest (usually black) is referred to as the background color. By analyzing the various color of CAPTCHA images, we found that:

1. Number of pixels of same or similar color in background always dominates number of pixels of characters to be recognized.
2. There is usually color difference of at least 35-40 pixel levels between foreground colors and background colors, considering readability factor of human being to identify characters from background.

In static threshold method of binarization process, pixels having value below predefined threshold are converted into black and the pixels having value above predefined threshold are converted into white. Generally it is observed that static threshold method doesn't give satisfactory results for images having variations in color. It is possible that sometimes it may erode the characters as shown in Fig.5. The binarization process in our preprocessing attack calculates threshold value during run time, depending on dominating color intensity value.

(a) (b) (c)

Fig. 5. a)Color image b)Grey scale image c) binarized image with static threshold=127

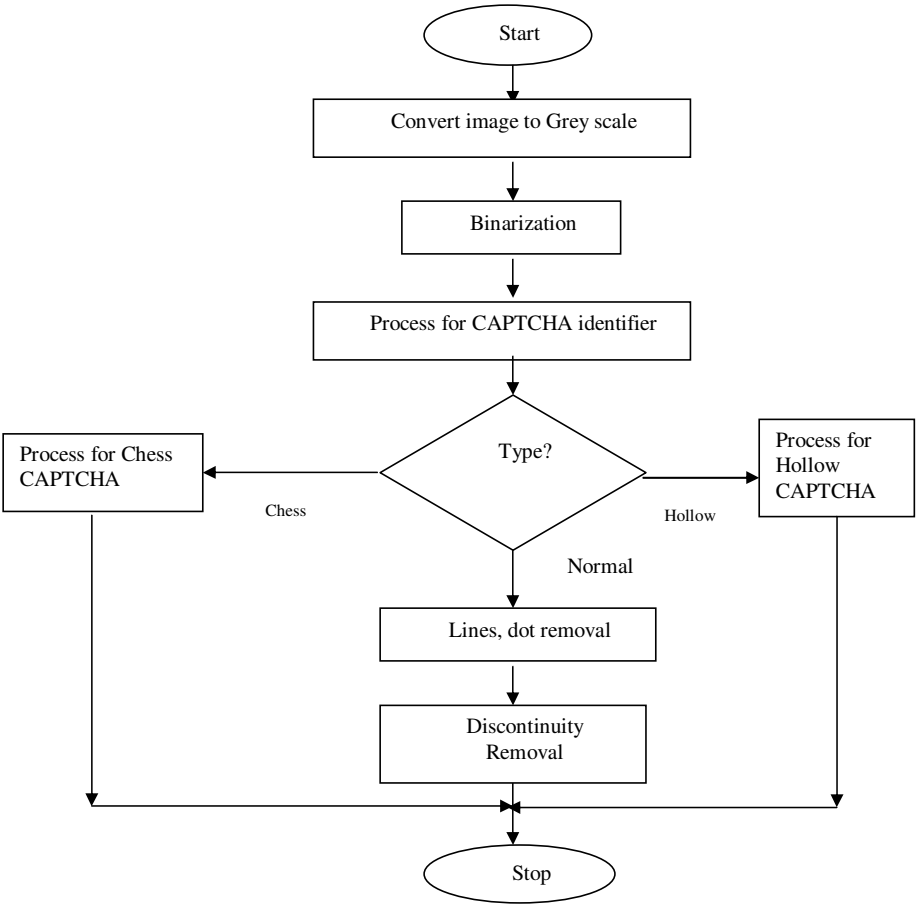


Fig. 6. Flow chart of Preprocessing Attack

(a) (b)

(c) (d)

Fig. 7. a) Color Image (b) Grey scale image (c) Color 255 replaced with white (d) Color 133 replaced with black

Dominating color is a color of maximum number of pixels has. First it scans the image and searches for such dominating color. If color difference between this dominating color and previous color is less than 35-40 pixel level then replace dominating color and similar color (similar color is calculated as ± 8 % of dominating color) with white and again scan the image otherwise replace with black. The binarization process is illustrated with example in Fig. 7.

Once image is converted into binary, it is passed through CAPTCHA identifier which identifies type of CAPTCHA and accordingly passes through particular processing algorithm. We have concentrated mainly on three types of CAPTCHA namely Chessboard i.e. Bot detect, Hallow and CAPTCHA with various kinds of clutter as noise such as Yahoo, MSN Hotmail, Digg CAPTCHA etc. Chess board CAPTCHA is determined using following algorithm:

1. Calculate width and height of alternating color sections.
2. Calculate average measure using eqn. 2

$$avg = \frac{\sum_{i=1}^{k} width[i]}{k}, \tag{2}$$

Where K = no. of width sample taken.

3. Calculate deviation of each measure from the average measure using eqn.3

$$Deviation = \frac{\sum_{i=1}^{k} |avg - width[i]|}{k} \tag{3}$$

4. Finally, obtain ratio of Deviation and avg as shown in eqn.4

$$ratio = \frac{Deviation}{avg} \tag{4}$$

5. If the ratio obtained lies between 0.9 to 1.1, then dimensions are considered equal or nearly equal.
6. Repeat steps 2-5 for height. If it is successful for width as well as height then it can be concluded that these measures are of fairly equal sized cells and hence this CAPTCHA is Botdetect CAPTCHA else test fails and algorithm proceeds with hollow CAPTCHA check.

To process Botdetect CAPTCHA, the algorithm first detects each chess box. If the majority of pixels in a box are black, all the pixels that are originally black are changed to white, and the pixels that are originally white, if any, are changed to black. If the majority of pixels in the box are white, then no color change is done. The result of Botdetect/Chess CAPTCHA process algorithm is shown in fig. 8.

Hollow CAPTCHA is determined based on observation that the contour of characters is formed by black pixels in such a way that number of black pixel count is always 25%-35% less than white pixel count. To determine Hollow CAPTCHA, ratio of total number of black pixels to total number of white pixels is calculated. If ratio is less than 0.35 then, algorithm proceeds with further check or else this check fails. The algorithm for Hollow CAPTCHA identification is as follows:

1. The background of image is filled with black color using boundary fill algorithm.
2. Calculate no. of pixels inside the characters using eqn.5

$$No.of\ pixels\ inside\ characters = (total\ no.\ of\ pixels\ in\ image) - (x + y) \tag{5}$$

Where total no. of pixels in image = height * width of image in pixels.

X is no. of pixels filled during boundary fill.

Y is no. black pixels initially in the binary image.

3. Calculate ratio of pixels inside the characters and total no. of pixels in the image as indicated in eqn.6

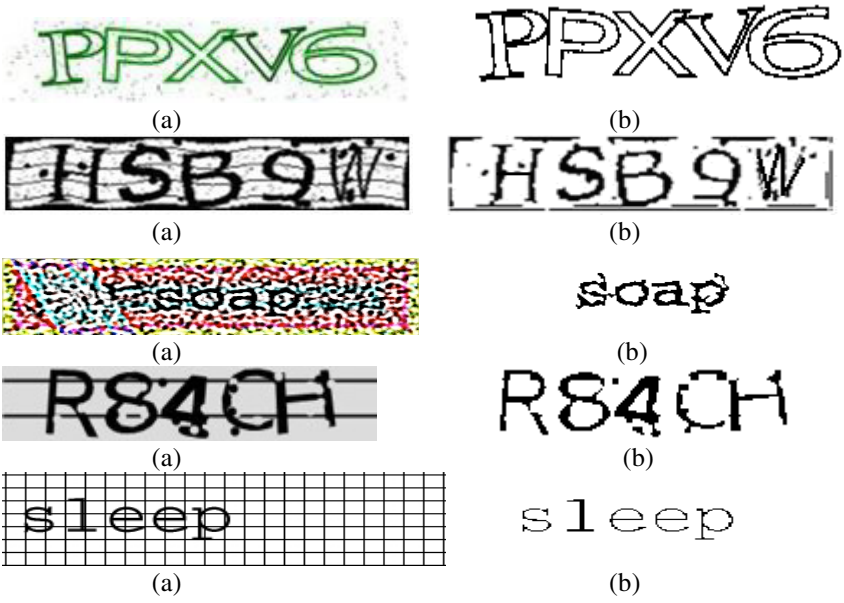
$$ratio = \frac{no.\ of\ pixels\ inside\ the\ characters}{total\ no.\ of\ pixels\ in\ the\ image} \tag{6}$$

4. If calculated ratio satisfies the observation then the CAPTCHA is considered as Hollow, or else check fails and CAPTCHA is considered as CAPTCHA with clutter.

Most of the social web sites like Yahoo, hotmail uses straight lines, slanting lines, wavy lines and arcs as clutter to confuse bots but at the same time maintaining human friendliness. The width of such type of clutter is less than width of characters so taking into consideration clutter width, our preprocessing attack easily cleans up the image for character recognition. The breaks in characters are removed using discontinuity algorithm which reconnects background pixel if any of its four neighbors in the direction of East, West, North and South pixels are of fore color.

(a) (b)

Fig. 8. a) Botdetect CAPTCHA b) Result of Botdetect CAPTCHA processing algorithm



(a) (b)

(a) (b)

(a) (b)

(a) (b)

(a) (b)

Fig. 9. a) Original image b) image cleaned by preprocessing attack

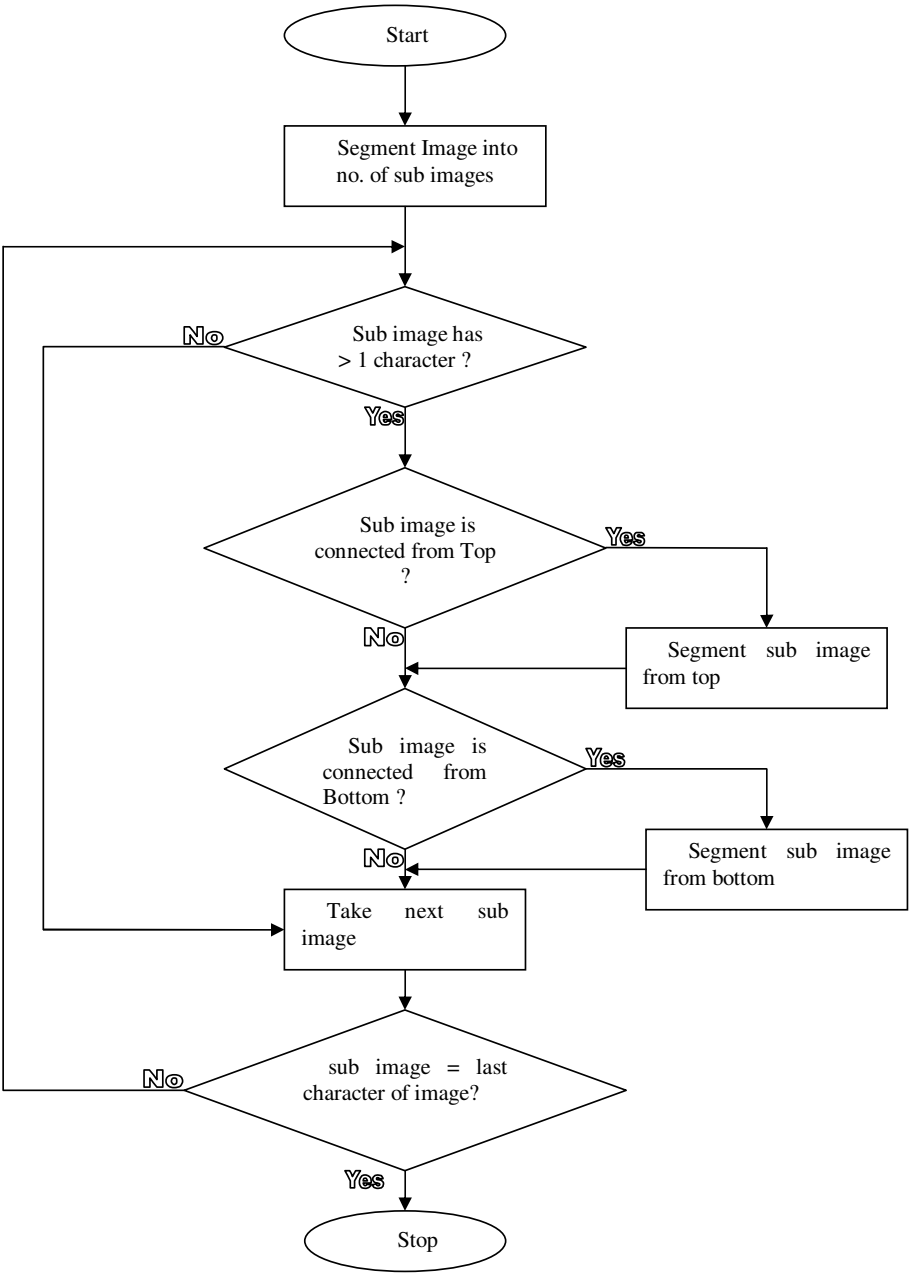


Fig. 10. Segmentation attack

3.2 Segmentation Attack

Segmentation attack extracts characters from image along with its position in image. We have developed universal structured flow chart for segmentation attack (shown in Fig.10) applicable to most of characters (i.e. connected, overlapped and disconnected characters) present in CAPTCHA image. Our future work will concentrate on implementation of structure which will be based on projection value of characters and vertical slicing process.

4 Results

In this section we present quantitative analysis of our preprocessed attack on various samples of standard database (approximately 200 images) obtained from social websites such as yahoo, Google etc. To show the effectiveness of our algorithm, we added various colors and random noise to images to generate distorted images and tested our algorithm on these images along with standard database obtained from social websites. The success rate is calculated depending on number of images cleaned. For example, if total number of images are 10 with each image having 6 characters and number of images converted to two valued color are 8 then success rate for binarization is calculated as (8*6)/ (10*6) = 80%. Similarly, the success rate for CAPTCHA identifier is calculated when type of CAPTCHA is identified correctly. The binarization gives approximately 95% success rate where as CAPTCHA identifier, Chess board and hollow CAPTCHA processing algorithm gives 94% result. The CAPTCHA with various types of clutter such as arcs, wavy and straight lines and dots are removed in 98% images of sample set. In all these experiments, we use same parameters specifically standard deviation between foreground and background in process of binarization .Fig.9 shows preprocessing attack is implemented successfully on targeted CAPTCHAs giving overall success rate as 97%.

5 Conclusion

There are two main explanations towards the shortcomings of security in the CAPTCHAs we analyzed. First, their design was almost exclusively based on research in computer vision, document recognition, and machine learning. However, our attacks did not rely on sophisticated, specialized algorithms. Instead, we applied our training in security engineering to identify critical vulnerabilities in each of the schemes, especially invariants at the pixel levels, and then design simple but novel methods to exploit those flaws. Second, a good CAPTCHA requires striking the right balance between robustness and usability, which often have subtle influences on each other. Our preprocessing attack has shown that a key strategy involved in preventing automated attacks is to incorporate random distortions in such a way that width of characters and distortion should not be noticeable. The experimental result proves that use of color doesn't enhance the security features but sometimes it annoys users

leading to usability issues. This paper has mentioned the effect of noise (one of the parameters of framework to measure strength of CAPTCHA) causing security as well as usability issues of CAPTCHA. Our future work will concentrate on next set of parameter used to measure strength of CAPTCHA through segmentation attack and hence to analyze its security. We don't claim the list of parameters we have discussed for security analysis of CAPTCHA is complete and encourage the researchers to identify more parameters using our framework.

Acknowledgements. The authors wish to thank the anonymous reviewers for their useful suggestions that helped in improving the quality of this paper. This work was supported in part by MAEERs MIT, Pune in association with University of Pune.

References

[1] Kato, N., Suzuki, M., Omachi, S., Aso, H., Nemoto, Y.: A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(3), 258–262 (1999)

[2] Lu, Y.: Machine Printed Character Segmentation-An Overview. Pattern Recognition 28(1), 67–80 (1995)

[3] von Ahn, L., Blum, M., Langford, J.: Telling humans and computers apart (automatically), CMU Tech. Report CMUCS-02-117 (2002)

[4] von Ahn, L., Blum, M., Hopper, N.J.: CAPTCHA: Using Hard AI Problems for Security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003)

[5] Mori, G., Malik, J.: Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 134–141 (2003)

[6] Moy, G., Jones, N., Harkless, C., Potter, R.: Distortion Estimation Techniques in Solving Visual CAPTCHAs. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), vol. 2, pp. 23–28 (2004)

[7] Chellapilla, K., Larson, K., Simard, P.Y., Czerwinski, M.: Building Segmentation Based Human-Friendly Human Interaction Proofs (HIPs). In: Baird, H.S., Lopresti, D.P. (eds.) HIP 2005. LNCS, vol. 3517, pp. 1–26. Springer, Heidelberg (2005)

[8] Yan, J., Ahmad, A.E.: A Low-cost Attack on a Microsoft CAPTCHA. Technical report, School of Computing Science, Newcastle University, UK (2008)

[9] Rabkin, A.: Personal knowledge questions for fallback authentication: Security questions in the era of Face book. In: IEEE Symposium on Usable Privacy and Security, SOUPS 2008 (July 2008)

[10] Chandavale, A.A., Sapkal, A.M., Jalnekar, R.M.: A framework to analyze security of Text based CAPTCHA. International Journal of Forensics and Computer Application (February 2010)

[11] Converse, T.: CAPTCHA Generation as a Web Service. In: Baird, H.S., Lopresti, D.P. (eds.) HIP 2005. LNCS, vol. 3517, pp. 82–96. Springer, Heidelberg (2005)

[12] Ahmad, A.E., Yan, J.: Colour, Usability and Security: A Case Study. Tech. report CS-TR 1203, School of Computing Science, Newcastle Univ. (May 2010), http://www.cs.ncl.ac.uk/publications/trs/papers/1203.pdf