Picture CAPTCHAs With Sequencing: Their Types and Analysis

Aditya Raj¹, Ashish Jain¹, Tushar Pahwa¹ and Abhimanyu Jain²,

Department of Information Technology

Netaji Subhas Institute of Technology, New Delhi, India¹

Delhi College of Engineering, New Delhi, India²

Abstract

CAPTCHAs are employed on web systems to differentiate between human users and automated programs which indulge in spamming and other fraudulent activities. CAPTCHAs currently in use have been broken and rendered ineffective as a result of continuous evolution in CAPTCHA breaking. Thus, there is a need to employ stronger CAPTCHAs to keep these breaking attacks at bay while retaining ease of implementation on websites and ease of use for humans. In this paper, we introduce Sequenced Picture Captcha (SPC). Each CAPTCHA round comprises of object pictures, each of which may be accompanied by a Tag. The user is required to determine the logical sequence of the displayed object pictures based on the Tags. We identify two generation schemes - one in which object pictures indicate an inherent sequencing and one in which explicit Tags are displayed for determining the sequencing. We also analyze all these schemes. The advantages of high user convenience and simplicity of operation are retained in both generation types.

Keywords: Picture CAPTCHA, SPC, Sequencing, Tag, Inherent, Non-inherent, Spamming, Usability, Security, HIP

1. Introduction

CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Human Apart) consist of Human Interactive Proof (HIP). HIP system is a broad set of protocols to distinguish between computer programs and human users. CAPTCHAs are employed as means to prevent bot programs (posing as human users) from indulging in spamming [13] and other unscrupulous activities like phishing [5], credit-card frauds and unauthorized access to information. The essence of a CAPTCHA is that it should be identified easily by a human but not by a bot. Broadly, CAPTCHAs can be classified into two groups - OCR-based

CAPTCHAs and Non-OCR based CAPTCHAs [6].

The current OCR-based CAPTCHA generation schemes remain insecure and exploited. Also, OCR-based CAPTCHAs have a limitation that the strength of these CAPTCHAs significantly depends upon the degree of distortion of the displayed text. We observed that, in general, higher distortion is aimed at to render breaking attacks useless, But high distortion may lead to failure of recognition by humans, thus making the CAPTCHA ineffective [18]. Further, for mobile phones and devices like PDA's and palmtops, the use of keyboard is infeasible thus making OCR-based CAPTCHAs inconvenient.

These weaknesses can be resolved by using Non-OCR based CAPTCHAs. These include picture CAPTCHAs which exploit two facts:

- The human eye is naturally very good in recognizing pictures of objects.
- The large variety of objects present in the world makes recognition of pictures infeasible for a computer.

SPC is a picture CAPTCHA which consists of object pictures and possibly their Tags which appear in the form of text-based CAPTCHAs. We explain this idea in detail later in the paper. The following section discusses some of the related work already performed in the generation and breaking of OCR and Non-OCR based CAPTCHAs.

2. Related Work

In this section, we discuss the research work already done in OCR-based and Non-OCR based CAPTCHAs.

2.1 Research Work Conducted in OCR-based CAPTCHAs

The MSN CAPTCHA (refer Figure 1(a)) has been broken by Yan et al.[17] with overall success of 60%. Yan et al.[16] used segmentation and pixel count lookup to

Figure 1. (a) An image of MSN CAPTCHA Figure 1. (b) An image of CAPTCHA from captchaservice.org Figure 1. (c) An image of google CAPTCHA Figure 1. (d) An image of EZ-gimpy CAPTCHA

break CAPTCHAs from the web-site captchaservice.org as shown in Figure 1(b). Dictionary attack was also applied and with that, 92% success rate was obtained. With snake segmentation and simple geometric analysis schemes, the success rate was boosted to 99% even when simple segmentation was not possible. The Google HIP (refer Figure 1(c)) has been broken by [2]. The EZ-Gimpy CAPTCHAs (refer Figure 1(d)) employed by Yahoo! were broken by Mori et al.[8]. Moy et al.[9] developed distortion estimation techniques to break EZ-gimpy with a success rate of 99% and 4-letter Gimpy-r with a success rate of 78%.

OCR CAPTCHAs used by Google and Yahoo! (shown in Figure 2(a) and (b)) suffer from usability issues [18]. Thus, in spite of good security, if a CAPTCHA scheme suffers from usability issues, the scheme is surely not the best.

Gupta et al.[6] introduced an OCR-based CAPTCHA generation scheme called Sequenced Tagged Captcha (refer Figure 3) which incorporated Tagging. In this, the characters of the CAPTCHA have a Tag (or embedded number) associated with them. The user has to enter the characters in a sequence which is determined by the Tags.

Chow et al.[4] proposed the idea of Clickable CAPTCHAs (refer Figure 4) which were primarily aimed for mobile phones and required the user to identify English words. In which they have created twelve Google CAPTCHAs and tiled them in a 3-by-4 grid. Of the twelve CAPTCHAs, three represented real English words (chosen at random from a dictionary of English words) whereas the remaining nine did not correspond to any English word. The user's task is to identify the three valid English words. To

Figure 2. (a) Image of google captchas showing usability problems Figure 2. (b) An image of yahoo CAPTCHA

Figure 3. A sample STC image

solve this clickable CAPTCHA, the user must click only on the three cells which contain English words. Any click on another CAPTCHA cell invalidates the solution.

Von Ahn et al.[15] introduced reCAPTCHA (refer Figure 5). Whereas standard CAPTCHAs display images of random characters rendered by a computer, reCAPTCHA displays words taken from scanned texts/old printed materials. The solutions entered by humans are used to improve the digitization process. To increase efficiency and security, only the words that automated OCR programs cannot recognize are sent to humans. However, to meet the goal of a CAPTCHA (differentiating between humans and computers), the system needs to be able to verify the user's answer. To do this, reCAPTCHA gives the user two words, the one for which the answer is not known and a second "control" word for which the answer is known. If users correctly type the control word, the system assumes they are human and gains confidence that they also typed the other word correctly.

2.2 Research Work Conducted in Non-OCR based CAPTCHAs

Non-OCR CAPTCHAs include audio [14], video [7], picture and logical [11] CAPTCHAs. But since we are mainly concerned with Picture CAPTCHAs, we mention the research work already conducted in this field.

Chew et al.[3] proposed Naming CAPTCHA (refer Figure 6(a)) in which the user is shown a group of object pictures and is required to identify the common theme among the pictures and type it as an answer. They also proposed Anomaly CAPTCHA (refer Figure 6(b)) in the same paper in which the user is shown a group of object pictures. The user is required to click the anomalous picture from among those displayed.

Shirali-Shahreza et al.[10] proposed a picture CAPTCHA called Collage CAPTCHA (refer Figure 6(c)). In this CAPTCHA, the user was shown pictures of some objects and was required to click on the picture of some specific object from among the displayed ones.

Shirali-Shahreza et al.[12] proposed Advanced Collage CAPTCHA. This CAPTCHA was identical to Collage CAPTCHA but the user now had to identify the required picture from a group of pictures appearing on the right of the screen also.

Baird et al.[1] proposed Implicit (refer Figure 7) in which certain hot spots were identified in each image. The user was asked to click on these hot spots in different rounds. This scheme suffered from the limitation that images had to be manually marked for such areas.

3. Weaknesses of Current Non-OCR based CAPTCHAs

In this section, we will discuss some terminology used throughout the paper which would be useful in understanding the weaknesses of currently proposed Non-OCR CAPTCHAs.

Figure 4. A sample image of Clickable CAPTCHAs

Figure 5. A sample re-CAPTCHA image

- Breaking Attacks - We observed that two types of attacks are feasible in the breaking of Picture CAPTCHAs - Random guessing attack and Pictionary-based Attack.

In Random guessing attack [3], we observed that the random guessing attack is the simplest attack that could be employed to break CAPTCHAs. It requires no computation time to solve the CAPTCHA and no significant resources on the hacker's part. Therefore, the random guessing attack can be used with a high frequency and even a low success rate like 16-20% results in a huge volume of spam. Hence, a robust CAPTCHA generation scheme must provide high security against this attack. Therefore, we have included this parameter in our analysis.

In Pictionary-based Attack, the bot maintains a 'picture dictionary' (Pictionary) or a 'look-up table' of object pictures along with their object names and any associated information useful in picture matching. Hence, whenever the bot comes across a picture, it searches for the picture by comparing properties like color/intensity, edge detected pattern, pixel pattern etc. with those of every picture it has in its table. If a match occurs i.e. the picture was 'old', the bot simply 'looks up' the answer. If there is no match, i.e. the picture is 'new', this picture gets added in the table for future look-up. Pictionary-based attack exploits the fact that some pictures may be repeated in CAPTCHA rounds after some initial rounds. This assists the hacker by boosting the average CAPTCHA breaking success rate. Therefore, we considered the Pictionary-based attack in our analysis.

- Implementation strategy - There are two general models of selecting the object pictures - Database Model and On-the-Fly Model. We consider both and discuss their advantages and disadvantages:

(a)

(b)

(c)

Figure 6. (a) An image of Naming CAPTCHA
Figure 6. (b) An image of Anomaly CAPTCHA
Figure 6. (c) An image of Collage CAPTCHA

Figure 7. A sample Implicit CAPTCHA image

In Database Model, a huge number of pictures are collected from the Internet and other sources and stored in a database. Pictures are randomly chosen from this database for display in CAPTCHAs. There are two advantages of using this model. Firstly, high quality pictures can be selected because the pictures are chosen and stored beforehand. Thus, the pictures do not pose any problem in user identification as each picture perfectly displays the required object. Potential problems like mislabelling [3] and low picture quality are eliminated. Secondly, since the pictures are stored beforehand, CAPTCHA generation is time-efficient. It requires little time to retrieve and load the images as suitably designed hashing functions could be employed. Hence, the loading time of the CAPTCHA decreases which increases the user-friendliness of the website and causes little inconvenience to users. However, some disadvantages associated with this model are, firstly, the cost of updating the database with new pictures will be high. This is because it will be a time consuming process since each picture will be accompanied by a 'label' and associated information (used during look-up) that will have to be entered manually or generated by some automated means. This process will have to be repeated periodically and hence will be costly. Secondly, the costs associated with the storage of a large number of pictures and database maintenance may become too high and unacceptable.

In On-the-Fly Model, first the names of the objects to be displayed are selected out of a bank of N object names. Then, random pictures of these selected objects are searched and selected from the Internet using standard search engines like Yahoo!, Google etc. These images are then displayed in the CAPTCHA. One of the benefits of this model is that since the pictures are obtained dynamically from the web, the costs associated with storage of a large number of pictures

Table 1. Mislabelling table

Keyword	Percentage of Mislabelling
Cat	41.11
Flower	32.22
Aeroplane	33.33
Computer	65.56
Book	60
Apple	70
Ball	68.89

and maintenance of a database is eliminated. Another advantage is that the web is completely dynamic and is updated constantly. Since in on-the-fly model, pictures are being searched on the web, the effort required to keep a static database, updating it with new pictures constantly is not a factor here. Thus, the costs associated with constant upkeep of a database are eliminated. However, there are some disadvantages associated with this model. Since the pictures are searched and chosen from the Internet, they may not completely match their object. Thus, mislabelling is present which reduces the user-friendliness of the CAPTCHA. Also, this model is not time-efficient. Time required to search for pictures on the web and to include them in the CAPTCHA may be significant enough to adversely affect the loading time of the CAPTCHA. This decreases its user friendliness. Further, if the website attracts high amount of traffic, pictures may start repeating themselves. This is because search engines only provide limited search results (1000 for Google). This may assist the hacker in breaking the picture CAPTCHA.

We have analyzed all the CAPTCHAs in this paper with respect to the following aspects:

- The success rate of Random guessing attack on the CAPTCHA.
- The success rate of Pictionary-based attack on the CAPTCHA.
- Mislabelling, mis-spelling, polysemy and synonymy as mentioned in [3].
- Requirement of keyboard.

We present the analysis of various currently known Non-OCR CAPTCHAs below.

- **Naming CAPTCHA**

The biggest weakness of Naming CAPTCHA is that since all the pictures depict a common theme, the bot

has to identify only one picture correctly to break the CAPTCHA. This can be realized with the following example. We assume that the hacker maintains a Pictionary for breaking. Suppose out of 6 shown pictures, the first picture scanned by the hacker was an old image (i.e., a 'hit' occurs in the Pictionary), then breaking requires one database search to obtain a success of 100%. If instead of the first, the second picture is the old one, only one additional database search will be required. Only if all the images are new for the bot would this method fail. Then the bot would have to randomly guess the answer from a dictionary (we assume a standard online Oxford dictionary consisting of 83,000 English nouns) giving a success rate of 0.0012%. Thus, assuming 'n' images, the average breaking success rate S by Pictionary-based attack is given by:

$$S = \frac{1}{2^n} \times 0.0012\% + \frac{2^n - 1}{2^n} \times 100\%$$

Taking $n=6$, this evaluates to 98.44%. Thus, it can be observed that if even one picture in the CAPTCHA round is an old one, it has serious security implications. Moreover, the probability of at least one image being old is quite high, more so after a few initial rounds.

The problems of Mis-spelling, Mislabelling, Polysemy and Synonymy are present in Naming CAPTCHA [3]. The solution to these problems was proposed by Chew et al. by increasing the number of rounds for users to pass in order to authenticate themselves. However, by increasing the number of rounds, the user-friendliness of the scheme is lost as solving more than one CAPTCHA becomes very annoying for users. A scheme which requires users to solve the CAPTCHA only once is surely better than a scheme requiring them to pass a number of rounds. It is customary to note that confusion will result in solving a CAPTCHA round if at least half of the shown pictures depict different objects than intended as a result of mislabelling. We documented the results of mislabelling of pictures from the Internet in Table 1. We obtained the values shown in the table by observing first 100 hits from Google search engine. The mislabelling increases further after the first 100 hits. However, these results are unreliable as the pictures from the web are subject to change. Therefore, the magnitude of mislabelling will vary according to time. However, these results are quite significant when one takes into account the huge number of Internet users. In such a scenario, re-generating the rounds to avoid mislabelling wastes Internet resources and further contributes to network clogging, which is the reason CAPTCHAs were employed in the

first place. Also, Naming CAPTCHA requires the use of keyboard which is infeasible in hand-held devices.

- **Anomaly CAPTCHA**

For 'n' displayed pictures, the probability of success by random guessing method is $\frac{1}{n}$. Usually, six images are shown. This gives us a success rate of 16.67%. This is a glaring weakness of Anomaly CAPTCHA. The success rate of bot can further increase if the bot is maintaining a Pictionary and some CAPTCHA pictures turn out to be old, more so, after a few initial rounds. To calculate the success rate of the bot, firstly, we assume that the number of pictures that turn out to be old is arbitrary and ranges from zero to 'n' (assuming a total of 'n' pictures are displayed). We obtain four cases. First, if the anomalous picture and at least two other pictures are identified, then the CAPTCHA is broken with 100% success. Second, if the bot identifies only the anomalous picture (i.e. only the anomalous picture was old) then the bot would have to make a random guess among all the displayed pictures, giving a $\frac{100}{n}\%$ success rate. Third, if the bot identifies the anomalous and one other picture only, then since they correspond to different objects, one among them must be anomalous. Thus the bot chooses among the two, giving a 50% success rate. Fourth, if the anomalous picture is not identified then the bot would have to guess among the unidentified pictures.

We used binomial distribution to obtain the probabilities of occurrence of each value of 'k', k being the number of old pictures and applied it to the above cases. For a total of six images the average success rate came to 59.36%. This is quite high and is therefore a major drawback. Also, Mislabelling and polysemy will be present in Anomaly CAPTCHA.

- **Collage CAPTCHA**

For 'n' displayed pictures, the probability of success by random guessing method is $\frac{1}{n}$. Assuming six pictures, this evaluates to 16.67%. This breaking rate is too high as mentioned earlier. If we use a Pictionary-based attack, we get much better success rates. To calculate the success rate of the bot, firstly, we assume that the number of pictures that turn out to be old is arbitrary and ranges from zero to 'n' (assuming a total of 'n' pictures are displayed). We obtain two cases. First, if the bot identifies the 'target' picture (the one that has to be clicked by the user for authentication), then obviously the success rate is 100%. In other cases, the bot would have to guess from among the pictures that have not been identified.

We used binomial distribution to obtain the probabilities of occurrence of each value of 'k', k being the

number of old pictures and applied it to the above cases. For six images, the average breaking success rate comes to 66.41% which is quite high and therefore a major drawback. Also, Mislabelling and polysemy will be present in Collage CAPTCHA.

After studying the generation and breaking of the above mentioned CAPTCHAs and other OCR-based CAPTCHAs, we observed that there is a strong need to employ a CAPTCHA which provides better security. Moreover, the CAPTCHA should be very convenient for humans to solve to retain user-friendliness. These observations serve as our motivation to present Sequenced Picture Captcha (SPC). We discuss SPC in the following section.

4. SPC Generation

SPC consists of a number of object pictures. Each picture may be accompanied by a Tag which is a text-based CAPTCHA. These object pictures are displayed to the user and the Tags are seen near their 'owner' picture. For solving SPC, the user is required to determine the logical sequence of the displayed objects based on their Tags. If the final answer as selected by the user is correct, only then is the user authenticated.

We now discuss the various stages involved in SPC generation.

- **Displaying Object Pictures** - Pictures of common noun objects like aeroplane, car, flower etc. are displayed. Object pictures can be displayed by either the database implementation or by on-the-fly implementation.
- **Generation of Tags** - We have considered STCs as Tags. Thus, the Tags offer high resistance to current breaking techniques. However, Tags may be chosen as desired. The type of Tags which appear along with the object pictures depend on the type of implementation scheme.

We now propose two different types of generation schemes of SPCs.

- **Inherent Scheme** - In this scheme, the object pictures themselves indicate the logical sequencing required to solve the CAPTCHA. Therefore, in this scheme, there is no need to display Tag images as the object pictures themselves are self-sufficient. This scheme has the advantage of minimal loading time because in each round, only the object pictures are displayed. We now illustrate two different types of Inherent Schemes.

Scheme-I - In this scheme, a picture of an object is selected and is dissected into a number of component pictures, each depicting a part of the original picture. Thus, each part indicates an inherent sequence to be followed. The user has to solve the

Figure 8. Sample images of inherent scheme-I of SPC

CAPTCHA with this ordering. For example, we could choose to show an image of a human body. The picture is horizontally cut (dissected) into three parts - Top (which contains the upper body parts like head, face etc.), Middle (containing the torso) and Bottom (containing legs and feet). In addition to these pictures, two pictures of random objects (other than the human body) are displayed. The user is then asked to click the pictures pertaining to the dissected object (in this case, the human body) in a particular order, say, in a top-to-down fashion, i.e., top, middle and then bottom. The user is validated if the clicking sequence is correct. A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spcin1.htm

Scheme-II - In this scheme, object pictures are selected such that they indicate an inherent sequencing based on some property of the object. The user has to solve the CAPTCHA by selecting object names from drop-down menu in a sequenced fashion which will be clear from the object pictures themselves. For example, we could choose to show pictures of a tree, two dolls, three coins, four pens etc. The user is validated if he/she clicks the pictures in a particular order - say, in increasing order (first the picture of a tree, followed by two dolls, three coins and four pens). The generation algorithm for scheme-II is as follows:

1. Select a random object.
2. Generate a random Tag.
3. Load a blank image.

4. Plot the pixel pattern of the selected object in the blank image as many times as the Tag digit indicates.
5. Repeat steps 1 through 4 as many times as there are objects displayed.

A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spcin2.htm

It is customary to note that each object picture will be displayed in a different manner each time. This is because the location where the pixel patterns of the object are plotted in the image is variable. However, overlapping between two pixel patterns of the object is controlled so that the user can distinguish between the two patterns and recognize them as two copies of the object. The copies may also be transformed to further increase security.

- **Non-Inherent Scheme** - In this scheme, Tag images are shown along with the object pictures to indicate the logical sequencing required to solve the CAPTCHA. Therefore, in this scheme, the presence of Tags augments the security of the CAPTCHA scheme. We now illustrate four different types of Non-Inherent Schemes.

Scheme-I - In this scheme, the Tags consist of numbers written in English words. For example, a picture of an apple may have the number 11 as its Tag which is displayed to the user as a text CAPTCHA with text 'eleven'. The picture of the apple then becomes the 'owner' picture of the Tag 'eleven'. Figure 9 (a) shows an example of such a Tag. The Tags may also be text CAPTCHAs which contain random English characters along with a number. The English characters act as noise to identification of the number and thus, prevent the breaking of the text CAPTCHA. Here, the picture of the apple may have the number 8 as its Tag, which may appear among the characters. Figure 9 (b) shows an example of such a Tag. The picture of the apple then becomes the 'owner' picture of the Tag '8'. The user is provided with drop-down menus containing a list of object names. There is one menu for every object picture so the number of drop-down menus is equal to the number of object pictures displayed. Some of the object names appearing in the menus are of the objects which appear in the displayed pictures and others are random names which are incorrect if selected as answers. The user is required to determine the logical ordering of the displayed objects based on the Tags. To submit an answer, the user selects the names of objects whose pictures are shown from the drop-down menus, in the order specified by the Tags of each picture. For example, the user may be required to order the objects in increasing order of

(a)

(b)

(c)

Figure 9. (a) Tag of SPC non-inherent Scheme-I, Figure 7. (b) Tag of non-inherent SPC Scheme-I, Figure 7. (c) Tag of non-inherent SPC Scheme-II

the numbers denoted by their respective Tags. This means that in the answer, an object, say apple, with Tag 'eleven' should be selected before another object, say tree, with Tag 'twenty'. A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spc1.php

Scheme-II - In this scheme, the Tags contain the names of objects whose images are displayed to the user. These Tags are shown in an arbitrary sequence to the user. The user is required to simply click on the object pictures in the order in which their corresponding Tags appear. Figure 9 (c) shows an example of such a Tag. This is a very user friendly scheme. A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spc2.htm

Scheme-III - In this scheme, the object pictures and their Tags are merged together in a single image. The Tags appear in the form of English words for numbers (eg., eleven) and overlap with the object

(a)

(b)

Figure 10. Sample images of non-inherent scheme-III of SPC

Figure 11. A sample image of non-inherent scheme-IV of SPC

picture. Each such image (containing an object together with a Tag) has variable transparency so that while the user has no problem in identifying the objects and Tags, the hacker's task is made tougher due to the increased complexity. Moreover, the location where the Tag is displayed in the object picture is selected randomly, so that it is different for each object and Tag pair in each round. The user is required to select the objects from a drop-down menu in the order in which their corresponding Tags appear on the object picture itself. Figure 10 shows an example of such captcha. A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spc3.php

Scheme-IV - In this scheme, all the Tags appear together in a bigger image. The Tags are in the form of English text. Some Tags correctly identify some objects displayed in the round while other Tags do not. Each Tag also contains an embedded number which determines the sequence of its 'owner' object. The user has to select the names of objects (which are correctly identified by some Tag) from a drop-down menu in the order determined by the embedded number in the object's Tag. Tags undergo transformation so the hacker's task is made tougher due to the increased complexity. Figure 11 shows an example of such a Tag. A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spc4.php

Scheme-V - In this scheme, Tags appear in the form of digits/numbers. However, Tags appear in their 'owner'

Figure 12. A sample image of non-inherent scheme-V of SPC

object picture itself rather than as separate images. Also, the Tags are colored. The Tags are embedded intricately in the object pictures so that they are clearly visible to users while increasing breaking complexity at the same time. The Tags could appear anywhere in the object pictures and indicate the sequence. The user is validated if he/she selects the object names in the correct sequence from the drop-down menu. We embed the Tag in the object picture using the following algorithm:

1. Select an object picture.
2. Generate a random Tag.
3. Flood-fill the object picture to obtain differently colored connected chunks.
4. Apply transformations on the Tag.
5. Flood-fill the Tag to obtain differently colored connected chunks.
6. Calculate the pixel count of the smallest connected chunk of Tag and also that of the largest connected chunk of Tag.
7. Locate all chunks in the object picture with pixel count in the range obtained in step 6 above.
8. Identify the color with which maximum number of chunks are present in the object picture. Select this color as the color of the Tag.
9. Locate a chunk in the object picture such that the difference between the color vector of the chunk and that of the Tag's color is maximum. Select this chunk as the background for Tag.
10. Display the Tag on the chunk identified in step 9 above.

Figure 12 shows an example of this type of SPC scheme. A sample implementation of this scheme is available at: http://stccaptcha.zymichost.com/form_spc5.php

In the following section we discuss the security aspects of the above proposed SPC schemes.

5. Security Analysis of SPC Schemes

• Inherent Schemes

Scheme-I - If the hacker employs random guessing to solve the CAPTCHA, then since an object will be dissected into parts in the CAPTCHA round, he/she will choose to click a minimum of three (corresponding to the case where some object pictures are of random objects) and a maximum of all the object pictures displayed in the round (corresponding to the case where all the object pictures shown pertain to the dissected object). This is because an object is dissected into at least three pictures. Since there will be only one correct sequencing, the random guessing probability evaluates to $\frac{1}{\sum_{t=3}^n \times \binom{n}{t} \times t!}$ where 'n' object pictures are displayed in a round and out of those 'n' pictures, 't' pictures pertain to the dissected object. Hence, to solve the CAPTCHA, 't' pictures are required to be clicked in a particular sequence.

The Pictionary-based attack is not feasible to break this scheme because the dissection points of object pictures are dynamic and chosen at run-time. Hence, even if the same object is dissected in different CAPTCHA rounds, the object will be dissected at different locations and the consequent pictures would be different.

Scheme-II - Since there will be only one correct sequencing, the random guessing probability of this scheme evaluates to $\frac{1}{N^t}$ where 'N' is the number of options for each object name in the drop-down menu and 't' is the number of object pictures displayed. We have assumed that the object pictures may repeat.

The Pictionary-based attack would be very difficult to employ to break this scheme because the object pictures displayed vary each time they are displayed. Even if the same object is selected to be displayed the same number of times in a picture, the location of its copies varies as the pictures are generated dynamically. Hence, random guessing attack seems to be the most feasible for this scheme.

• Non-Inherent Schemes

Scheme-I - The random guessing probability of breaking SPC Scheme-I evaluates to $\frac{1}{N^t}$ where 'N' is the number of options for each object name in the drop-down menu and 't' is the number of object pictures displayed. We have assumed that the object pictures may repeat.

The probability of breaking SPC Scheme-I by Pictionary-based Attack is calculated as follows: We assume the following variables: 'N' is the number of options provided in the menus. 't' is the number of displayed object pictures. 'k' is a variable that varies from zero to 't' and is used to denote the number of new pictures in the CAPTCHA. The probabil-

ity that out of 't', 'k' pictures are new is given by: $\frac{1}{2^t} \times \binom{t}{k}$ (according to binomial distribution). The term $\frac{1}{\binom{t}{k}}$ denotes that there are $\binom{t}{k}$ different orderings of 'old' and 'new' pictures possible and only one of them is correct. Since the Tags are difficult to break, the bot would have to 'guess' the correct order. For the (t-k) old pictures, since the bot knows their object names, it only has to guess their correct order. Since (t-k)! different orders are possible, the probability that the bot guesses correctly is $\frac{1}{(t-k)!}$. For the rest of the k pictures which are new, the bot knows neither their object names nor their correct order. The bot would then have to guess among the $N - (t - k)$ remaining options. The number of possibilities are: $(N - (t - k)) \times (N - (t - k) - 1) \times \dots \times (N - t + 1)$. Thus, the probability that the bot correctly guesses the order of the new pictures is

$$\frac{1}{((N - (t - k)))} \times \frac{1}{(N - (t - k) - 1) \times \dots \times (N - t + 1)}$$

which when simplified, becomes $\frac{(N-t)!}{(N-(t-k))!}$.

To account for repetition of pictures, we assume that we have 'x' distinct old pictures and r1,r2 upto rx are the number of occurrences of the the distinct old pictures. We also assume that 'y' denotes the number of distinct new pictures and c1,c2 upto cy are the number of occurrences of the the distinct new pictures. They are usually equal to one except in cases of pictures that are repeated. We incorporate these factors into our equation and multiply the above terms. Summing from k=0 to k=t, we get the following expression for average success rate of Pictionary-based attack as:

$$P = \sum_{k=0}^t \times \frac{\binom{t}{k}}{2^t} \times \frac{1}{\binom{t}{k}} \times \frac{r1! \times r2! \times \dots \times rx!}{(t-k)!} \times \frac{c1! \times c2! \times \dots \times cy! \times (N-t)!}{(N-t+k)!}$$

After simplifying the above expression and assuming no repetition of pictures such that r1 upto rx and c1 upto cy all become equal to one, we get the following equation

$$P = \sum_{k=0}^t \times \frac{1}{2^t} \times \frac{(N-t)!}{(t-k)! \times (N-t+k)!}$$

Meanwhile, in the case of Tags being numbers present as English words, the Tags can be made secure by increasing overlapping among the characters and increasing the amount of distortion applied. This does reduce readability but since these are English words, it is an acceptable trade-off. In the case of Tags being present as numbers containing random English characters as noise, the Tags are present in the form of words.

Scheme-II - The random guessing probability of breaking of SPC Scheme-II evaluates to $\frac{1}{n!}$ where n is the number of pictures displayed. However, Pictionary-based Attack is not possible in this scheme since even if one or more of the pictures are identified by the bot, the sequence in which the pictures must be clicked remains unclear.

Scheme-III - The random guessing probability of breaking SPC Scheme-III evaluates to $\frac{1}{N^t}$ where 'N' is the number of options for each object name in the drop-down menu and 't' is the number of object pictures displayed. We have assumed that the object pictures may repeat. However, since the object pictures and Tags are merged together dynamically, the location where the Tag is displayed in the object picture varies with each round. Hence, the same object and Tag pair will be displayed differently even if it repeats in some round. Thus, Pictionary-based attack is not feasible to break this scheme.

Scheme-IV - In this scheme, the hacker does not know how many objects out of those displayed will be present in the answer because this depends on whether any Tag correctly identified the object. Now, in every round, a minimum of three pictures would be there which are correctly identified by a Tag. Therefore, assuming 't' to be the number of options provided in the menus, 'n' to be the number of displayed object pictures, the random guessing probability of this scheme evaluates to $\frac{1}{\sum_{p=3}^n \times \binom{n}{p} \times t \times p!}$. Pictionary-based attack is infeasible to break this scheme because even if the hacker identifies an object in a round, it is still unclear whether that object will be present in the answer. Thus, object matching in the Pictionary serves no purpose.

Scheme-V - The random guessing probability of breaking SPC Scheme-V evaluates to $\frac{1}{N^t}$ where 'N' is the number of options for each object name in the drop-down menu and 't' is the number of object pictures displayed. We have assumed that the object pictures may repeat. Since the location of the Tag in the object picture is determined at run-time, the same object and Tag pair will be displayed differently in different rounds. The variable position of Tag will mix with the object picture and such an image will not be conducive for image matching. Hence, Pictionary-based

attack is not feasible to break this scheme.

Since the transformations applied on the Tag include fragmentation, the Tag will not be present as a connected chunk. Moreover, the object pictures themselves contain colored chunks pertaining to the different shades in the object. These chunks, coupled with the disconnected chunks of the Tag, render geometric analysis of the Tag infeasible. Hence, in order to counter shape detection by pixel counting, we employ the algorithm mentioned in the previous section to display the Tag in the object picture.

We have documented the average breaking success rates for all discussed CAPTCHAs in Table 2. We discuss below some aspects related to SPC.

- By increasing the number of options in the drop down menus or by increasing the number of displayed object pictures, we can increase the security of the schemes.
- All the schemes are much more resistant to both random guessing attack and Pictionary-based attack as compared to other picture CAPTCHAs. This can be observed from Table.
- Benefits of Sequencing - SPCs incorporate Sequencing which has the inherent strengths as mentioned in [6].
- The problem of Polysemy is absent in both the inherent SPC schemes while it is present in all other Picture CAPTCHAs. This is because the object pictures themselves indicate a sequence and the final answer depends on this sequence. The user is required to identify the object to know the inherent sequence but not to explicitly name the object.
- Due to the nature of Inherent SPC schemes and Non-Inherent SPC schemes-II,III,IV and V, Pictionary-based attack is not feasible to break these schemes. This is a very significant advantage as each time, either a new picture and Tag pair is generated or the scheme is such that Pictionary-based attack serves no purpose to the hacker. This is a big advantage as compared to other Picture CAPTCHAs.
- Only Non-Inherent SPC schemes-I and II require a Tag for each object pictures. This may increase the overall loading time a little, but this effect can be ignored as the security offered is much higher than other Picture CAPTCHAs.

In the following section, we present the advantages and benefits of employing SPCs on web systems.

6. Advantages of Using SPC

There are some inherent advantages associated with SPCs. These are as follows:

- Being a Picture CAPTCHA, SPC can be used by users of all ages, including children and elderly people. This enables SPC to cater to all types of users accessing the Internet.
- SPCs can be integrated easily on online systems and websites.
- Unlike OCR-based CAPTCHAs, SPCs do not require the user to type anything. This eliminates the need of keyboard. Therefore, SPCs can be solved on hand-held devices or devices in which it is cumbersome to use the keyboard, such as PDAs, mobile phones or palmtops.
- SPC does not require any processing on client side. Thus it is feasible and suitable to use on small devices and devices with limited resources such as mobile phones, PDAs etc.
- The security of SPC can be varied according to convenience and requirement. This can be accomplished by adding more options in the drop-down menus, increasing the number of object pictures and morphing the object pictures. By morphing, the pictures can be sufficiently modified in terms of their color scheme, intensity and introduction of noise and other random shapes and objects such that they convey the same meaning to the user but a bot would not be able to recognize it as an old picture and would consider it as new. This has the dual advantage of forcing the bot to resort to random guessing which is less effective and unnecessarily increasing the size of the database maintained by the bot.

7. Conclusions

OCR-based CAPTCHAs have been broken and remain insecure. Non-OCR based CAPTCHAs retain convenience of operation for humans as they exploit the natural skill of the human eye of identifying pictures. We employed the concept of Sequencing in Picture CAPTCHAs to introduce SPC. SPC generation can be classified into two types - inherent sequencing and non-inherent sequencing. The former does not require Tags while the latter does. SPCs incorporate two levels of security, viz. recognition of objects in pictures and determining their logical sequence. As can be observed from Table 2, SPC offers much higher security as compared to currently known Non-OCR based CAPTCHAs. This, coupled with picture morphing and

Table 2. Comparison of previously proposed Non-OCR CAPTCHAs with the proposed SPC schemes

	Mislabel- ling	Misspel- ling	Synony- my	Polyse- my	Need of Tag Image for Each Object Picture	Success Rate of Random Guessin g Attack	Success Rate of Pictiona- ry- based Attack
Naming CAPTCHA	Present	Present	Present	Present	No	0.0012%	98.44%
Anomaly CAPTCHA	Present	Absent	Absent	Present	No	16.67%	59.36%
Collage CAPTCHA	Present	Absent	Absent	Present	No	16.67%	66.41%
Inherent SPC							
Scheme-I	Present	Absent	Absent	Absent	No	0.0521%	NA
Scheme-II	Present	Absent	Absent	Absent	No	0.0001%	NA
Non-Inherent SPC							
Scheme-I	Present	Absent	Absent	Present	Yes	0.0001%	0.0088%
Scheme-II	Present	Absent	Absent	Present	Yes	0.1389%	NA
Scheme-III	Present	Absent	Absent	Present	No	0.0001%	NA
Scheme-IV	Present	Absent	Absent	Present	No	0.0052%	NA
Scheme-V	Present	Absent	Absent	Present	No	0.0001%	NA

the difficulties associated with breaking by employing Pictionary makes SPCs very hard to break. The usability of SPCs can be increased by using higher quality pictures, avoiding mislabelling even at the cost of repetition of pictures and increasing readability of Tags, all the while retaining same levels of security. SPCs combine the strong points of both OCR-based and Non-OCR based CAPTCHAs in terms of security and ease of use. Hence, SPCs are a new step in the evolution of Picture-based CAPTCHAs.

References

- [1] H. S. Baird and J. L. Bentley. Implicit captchas. In *Proceedings, SPIE/IS&T Conference on Document Recognition and Retrieval XII (DR&R2005)*, San Jose, CA, January 2005.
- [2] K. Chellapilla and P. Simard. Using machine learning to break visual human interaction proofs (hips). In *Advances in Neural Information Processing Systems 17, Neural Information Processing Systems (NIPS)*, pages 265 – 272, 2004.
- [3] M. Chew and J. D. Tygar. Image recognition captchas. In *7th International Information Security Conference, (ISC'04)*, pages 268 – 279, September 2004.
- [4] R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang. Making captchas clickable. In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 91 – 94, New York, NY, USA, 2008. ACM.
- [5] R. Dhamija and J. D. Tygar. Phish and hips: Human interactive proofs to detect phishing attacks. In *Second International Workshop on Human Interactive Proofs, (HIP 2005)*, pages 127 – 141, USA, May 2005.
- [6] A. Gupta, A. Jain, A. Raj, and A. Jain. Sequenced tagged captcha: Generation and its analysis. In *International Conference on Advanced Computing, (IACC'09)*, pages 1286–1291, India, March 2009.
- [7] K. A. Kluever and R. Zanibbi. Balancing usability and security in a video captcha. In *Proceedings of the 5th Symposium On Usable Privacy and Security 2009*, Mountain View, CA, USA, July 2009.
- [8] G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *Conference on Computer Vision and Pattern Recognition, (CVPR'03)*, volume 1, pages 134 – 141, June 2003.
- [9] G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion estimation techniques in solving visual captchas. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR04)*, volume 2, pages 23 – 28, 2004.
- [10] M. Shirali-Shahreza and S. Shirali-Shahreza. Collage captcha. In *Proceedings of the 20th IEEE International Symposium Signal Processing and Application (ISSPA 2007)*, Sharjah, United Arab Emirates, February 2007.
- [11] M. Shirali-Shahreza and S. Shirali-Shahreza. Question-based captcha. In *ICCIMA '07: Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, pages 54 – 58, Washington, DC, USA, 2007. IEEE Computer Society.
- [12] M. Shirali-Shahreza and S. Shirali-Shahreza. Advanced collage captcha. In *Proceedings of the 5th International Conference on Information Technology: New Generations (ITNG 2008)*, pages 1234 – 1235, Las Vegas, Nevada, USA, April 2008.
- [13] M. Siponen and C. Stucke. Effective anti-spam strategies in companies: An international study. In *39th Annual Hawaii International Conference on System Sciences, (HICSS'06)*, volume 6, pages 127c – 136c, January 2006.
- [14] J. Tam, J. Simsa, S. Hyde, and L. V. Ahn. Breaking audio captchas. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, NIPS*, pages 1625–1632, Vancouver, British Columbia, Canada, December 2008.
- [15] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. In *Science Express*, volume 321. no. 5895, pages 1465 – 1468, September 2008.

- [16] J. Yan and A. S. E. Ahmad. Breaking visual captchas with nave pattern recognition algorithms. In *23rd Annual Computer Security Applications Conference, (ACSAC'07)*, pages 279 – 291, USA, December 2007.
- [17] J. Yan and A. S. E. Ahmad. A low-cost attack on a microsoft captcha. In *15th ACM conference on Computer and communications security, (CCS'08)*, pages 543 – 554, October 2008.
- [18] J. Yan and A. S. E. Ahmad. Usability of captchas or usability issues in captcha design. In *Symposium On Usable Privacy and Security, (SOUPS'08)*, pages 44 – 52, USA, July 2008.