The ISC Int'l Journal of Information Security
ISeCure

September 2012, Volume 4, Number 1 (pp. 63–76)
http://www.isecure-journal.org

SEIMCHA: A New Semantic Image CAPTCHA Using Geometric Transformations

Maryam Mehrnejad[1,*], Abbas Ghaemi Bafghi[1], Ahad Harati[2], and Ehsan Toreini[3]

[1] *Information and Communication Security Lab., Computer Department, Ferdowsi University of Mashhad (FUM), Iran*
[2] *Machine Vision Lab., Computer Department, Ferdowsi University of Mashhad (FUM), Iran*
[3] *Computer Department, Faculty of Engineering, Islamic Azad University – Mashhad Branch, Mashhad, Iran*

ARTICLE INFO.

Article history:
Received: 8 December 2011
Revised: 23 March 2012
Accepted: 24 January 2012
Published Online: 11 July 2012

Keywords:
SEIMCHA, Semantic Image CAPTCHA, Geometric Transformation, Upright Orientation, Tagging Image CAPTCHA, Random Guessing.

ABSTRACT

As protection of web applications are getting more and more important every day, CAPTCHAs are facing booming attention both by users and designers. Nowadays, it is well accepted that using visual concepts enhance security and usability of CAPTCHAs. There exist few major different ideas for designing image CAPTCHAs. Some methods apply a set of modifications such as rotations to the original image saved in the data base, to make the CAPTCHA more secure.

In this paper, two different approaches for designing image based CAPTCHAs are introduced. The first one—which is called Tagging image CAPTCHA—is based on pre-tagged images, using geometric transformations to increase security, and the second approach tries to enhance the first one by eliminating the use of tags and relying on semantic visual concepts. In fact, recognition of upright orientation is used as a visual cue. The usability of the proposed approaches is verified using human subjects. An estimation of security is also obtained by different kinds of attacks. Further studies are done on the proposed transformations and also on the properness of each original image for each approach. Results suggest a practical Semantic Image CAPTCHA which is usable and secure compared to its peers.

1 Introduction

Completely Automated Public Turing Test to Tell Computer and Humans Apart (CAPTCHA) offers a way to make distinction between a human and an artificial agent. Nowadays, with an increasing rate of free web services the problem of misuse through spammers and automated soft-bots is getting worse on regular basis. Therefore, it is crucial to make such a distinction.

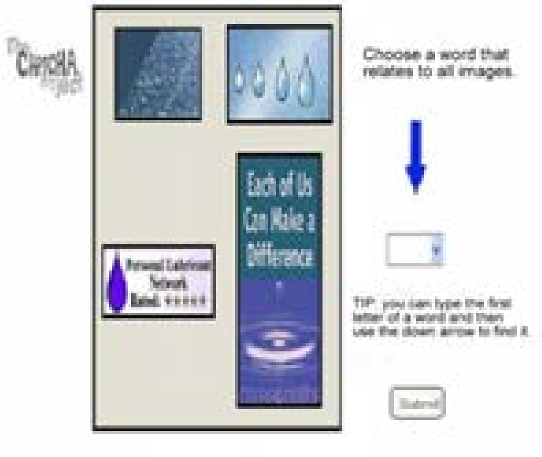
Various criteria have been proposed in the literature for evaluating CAPTCHAs. We will consider the following four properties (originally reported in [1]) in development of CAPTCHAs:

(1) Automated: Tests should be easy to be automatically generated and graded by a computer.
(2) Open: The underlying database(s) and algorithm(s) used to generate and grade the tests should be public. This property is in accordance with Kerckhoffs's Principle, which states that a system should remain secure even if everything about the system is public knowledge.
(3) Usable: Tests should be easily solved by humans in a reasonable amount of time. Furthermore, the effect of a user's language, physical location,

[*] Corresponding author.

Email addresses: maryam.mehrnejad@stu-mail.um.ac.ir (M. Mehrnejad), ghaemib@ferdowsi.um.ac.ir (A. G. Bafghi), a.harati@ferdowsi.um.ac.ir (A. Harati), etoreini@mshdiau.ac.ir (E. Toreini).

ISSN: 2008-2045 © 2012 ISC. All rights reserved.



Choose a word that relates to all images.

TIP: you can type the first letter of a word and then use the down arrow to find it.

Figure 1. PIX CAPTCHA

education, and/or perceptual abilities should be minimized.

(4) Secure: Tests should be difficult for machines to solve algorithmically.

The first CAPTCHA was a text based one and was proposed in 2000 for Yahoo in Carnegie Melon University [2]. After that, text based CAPTCHAs began to be considered by researches widely in the last 10 years. Since designing and implementation of text based CAPTCHAs is simple, they are being used wide-spreading today. But still some people find the current text-based CAPTCHAs annoyingly difficult [3]. Also there are different ways to attack a text based CAPTCHA based on Optical Characters Recognition (OCR) algorithms. Chandaval et al has developed a framework to attack text based CAPTCHAs [4]. They discussed various ways of CAPTCHA breaking using bots and proposed a framework for examining strength of these CAPTCHAs.

Different methods are proposed to replace text-based CAPTCHAs including Image CAPTCHAs, Video CAPTCHAs and Audio CAPTCHAs [1, 5–9]. Also some combinations of these methods are being used. Recognizing these media has more difficulty for computers compared with Text based ones. Image CAPTCHAs are facing booming attention both by users and designers due to more security and usability. Therefore they are used as a good alternative for text based CAPTCHAs. The first idea using image to tell humans and machine apart is used in ESP-PIX [10]. In this CAPTCHA which uses a limited database of tagged images, some photos of a similar topic are chosen and the user should guess the topic and select it from a given list. Figure 1 shows an example of this CAPTCHA.

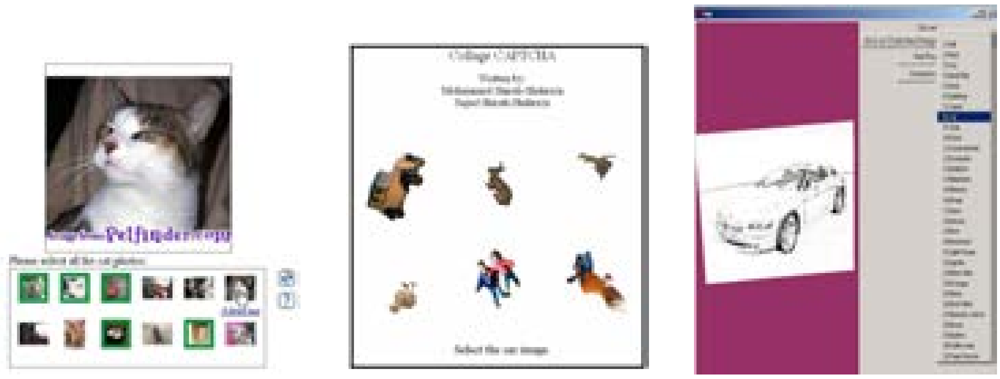
Text based CAPTCHA designers use a random generator to produce a word containing some random characters -in some cases meaningful words from dictionaries- to increase the number of outputs [6], but in image based CAPTCHAs it is not possible to produce a meaningful image easily. Using a limited database is an inevitable solution for image based designers. Therefore, there is a tight-coupling between the general security of an image based CAPTCHA and the security of image database. Note that even if the database is not open—which breaks the rules of an ideal CAPTCHA—the attacker can still or acquire all or a part of the images by frequent use of CAPTCHA and attack the CAPTCHA by using machine learning or direct matching techniques—which are discussed in detail later. So it is essential to develop some solutions to improve the security of CAPTCHA independent of its database [3]. In this way, we could prevent a successful attack, even if we miss the security of data base. These solutions could be:

- Using unlimited databases
- Updating the limited database
- Showing a transformed image instead of the original one.

The third solution is the more common method and a combination of all mentioned methods could be applied too. In simple image CAPTCHAs no changes are made to the images and the user is asked to type or select the name of the image from a list. In more sophisticated CAPTCHAs, some modifications like image rotation is applied and the user is asked to determine these changed images. Examples of such CAPTCHAs will be introduced in next section.

There are different kinds of image based CAPTCHAs. Some uses tagging or labeling which is assigning one or some words to some objects in an image. Automated meaningful tagging by machine is a challenge in this area. Content Based Image Retrieval (CBIR) and Knowledge Based Image Retrieval (KBIR) algorithms can be used to obtain meaningful tags for images automatically, though it is very hard [11]. One of the reasons why machines are weaker in tagging than humans is the fact that humans use the background of image to tell the tagging, which machines are incapable of.

In this paper, first we introduce a new tagging image CAPTCHA using geometric transformations as a more complex method. Indeed, we consider a set of 3D shapes such as sphere, cone and other shapes and wrap the original image on to one of them. Then a 2D projection from a random viewpoint gives the final image and helps to generate many new various images of each original image. The user should recognize the transformed picture and finds an appropriate tag from a proposed list.



(a) Assira CAPTCHA system (b) Collage CAPTCHA (c) 2D CAPTCHAs from 3D models

Figure 2. Example of image based CAPTCHAs

Tagging CAPTCHA systems are difficult since users require a priori knowledge of the image tags and it is a language-dependent method. In addition, Machine's weakness in automatic tagging, not only counted as an advantage for designers since bots cannot attack easily, but also is a disadvantage too as a wide range of reliable labels that are not available for most images on the web to create a random challenge. So as addressed in [5] there are some common techniques used to gather proper labels for images:

(1) Using the label assigned to an image by a search engine,
(2) Using the context of the page to determine a label,
(3) Using images that were labeled when they were encountered in a different task, or
(4) Using games to extract the labels from users (such as the ESP game).

It is obvious that there are some limitations to obtain labels in these methods since asking user to solve tagging CAPTCHA system. Noisy labels, unreliable and unrelated labels, misspelling, synonymous words, linguistics problems and etc, are the main obstacles of these methods.

One way to escape tagging CAPTCHAs is to apply semantic content in images. Semantic cues could be identified by users instead of selecting and/or mapping tags. It is a new solution in which limited CAPTCHAs applied it so far. Upright orientation of an image is a semantic which is easy for human to comprehend and hard for machine. Currently, automatic detection of such concepts is possible only for a small subset of images [12, 13].As reported in [5] 68.75% of users preferred rotating images as CAPTCHA, and 31.25% of users preferred deciphering text. So it seems that upright concept of an image is a potential choice to use as CAPTCHA. There are a few works based on this idea which will be explained in next section.

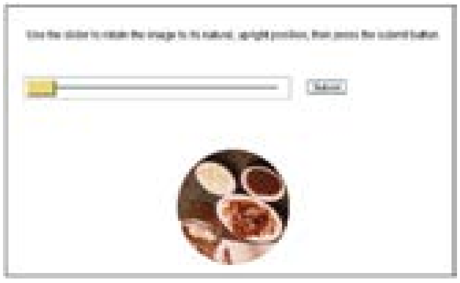
We have extended the proposed Tagging CAPTCHA based on upright orientation concept and designed a novel semantic image CAPTCHA named SEIMCHA in which the user should click on the upper area of the transformed image. This is the first time that geometric transformations and upright orientation concept are applied to design a CAPTCHA system. This combination leads to a more secure and more usable CAPTCHA.

Section 2 introduces related works containing prior tagging CAPTCHAs and upright orientation based CAPTCHAs. In Section 3, the proposed Tagging CAPTCHA is described. Furthermore it presents all applied transformations including geometric functions, Also security and usability analysis on the proposed Tagging CAPTCHA are described in this section. Section 4 presents SEIMCHA based on upright orientation and geometric transformations and includes all security and usability analysis. In Section 5, we make some comparisons between the proposed methods and similar works. Finally, Section 6 is conclusion and suggested future works.

2 Related Works

Since this paper introduces two separated CAPTCHA systems—the Tagging CAPTCHA and SEIMCHA-related work falls into two main groups. The former introduces some non-semantic image based CAPTCHAs from different level of distortion on images and the latter group contains all previous works have been done based on upright orientation.

There are various CAPTCHAs from basic to advanced which uses images without any changes, few changes and sophisticated ones. Microsoft Assira is a famous example of these CAPTCHAs in which the user must choose cats in a 12 image set of cats and dogs [7]. Figure 2a shows a screen shot of Assira. Col-



Use the slider to rotate the image to its natural, upright position, then press the submit button.

Figure 3. What's Up CAPTCHA interface [5]

lage [14] is another instance which displays some rotated images and the user has to find an object which the algorithm requests (Figure 2b). Improved Collage [8] is a promoted version of Collage in which a random number of images were chosen and edited, and then the user assigns each photo to their names on the other side of the page. In more advanced systems, more changes are made to the pictures. In [9] a 2D CAPTCHA is proposed using 3D models. A limited database of 3D images is applied in this CAPTCHA, and these pictures are converted with changes such as rotation, brightness, size and etc. to produce an unlimited number of 2D images for showing to user. The Graphic User Interface (GUI) asks the user to decide which tag is most suitable for the picture (Figure 2c). It is important to notice that there are several types of such image CAPTCHAs. But since in the proposed Tagging image CAPTCHA the focus is on the transformations, so Assira, Collage and 2D CAPTCHAs from 3D models are appropriate nominations for making comparisons.

On the other hand, based on our knowledge, there are two main works based on upright orientation of an image. Gossweiler et al. proposed the idea of image orientation as a basis for an image based CAPTCHA [5]. They called their work "What's UP CAPTCHA" and introduced it in this way: "This experiment will present a series of images one at a time. Each image will be rotated to a random angle. Use the provided slider to rotate the image until you believe it is in its natural, upright position, then press submit to go to the next image. This process will continue until you have adjusted ten images." [5]. Figure 3 shows a screenshot of What's UP CAPTCHA system.

As an extension of What's Up CAPTCHA, Ross et al. introduced a new CAPTCHA based on upright orientation of line drawing rendered from 3D models which is called Sketcha [3]. They download their models from Google3D Warehouse and render a collection of images from various angles. A screenshot of Sketch is shown in Figure 4. They explain Sketcha's response mechanism in this way: "The user's goal is to rotate each image until it is upright, choosing among four orientations by clicking on the image. Each line drawing was automatically rendered from a 3D model using a randomized point of view, providing for many possible images from each model" [3].

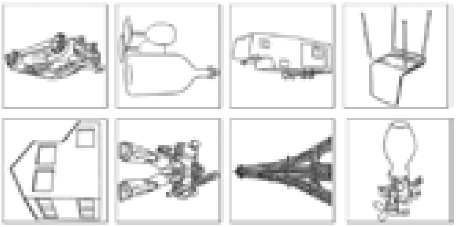


Figure 4. Example CAPTCHA based on line drawings.

We will discuss the pros and cons of each mentioned CAPTCHA and the proposed ones in Section 5.

3 Tagging Image CAPTCHA based on Geometric Transformations

As it was mentioned before, first, a new method is proposed which doesn't need any huge image database or having a large number of saved tags for CAPTCHA. A constant number of 30 images are used in the database. Each image is tagged by its own name. These tags which contain different subjects like animals, foods, different scenes and etc. are selected the same as in [9]. Furthermore, the images and tags are not ambiguous for humans. The images are transformed by some geometric transformation functions which are a novel approach to create a large search space from a finite image database. This section explains whole steps for developing, testing and evaluating the proposed Tagging image based CAPTCHA.

3.1 Transformations

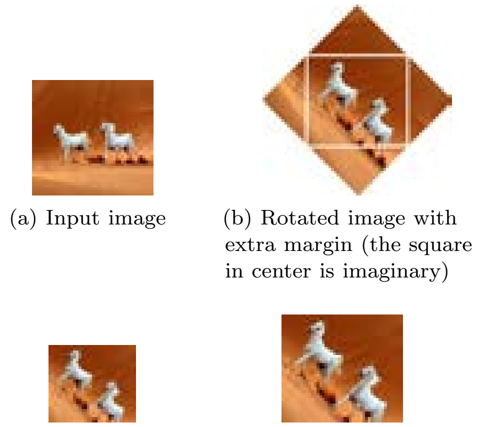
We apply some transformation functions to modify an input image. These functions include simple rotations and geometric transformations. Then we convert 3D object to 2D images by capturing from a random viewpoint. The algorithm below describes the approach better:

(1) Randomly select an input image
(2) Randomly rotate input image
(3) Randomly select a 3D geometric transformation and transform image on it
(4) Capture a 2D image from a random viewpoint

These transformations are implemented in Matlab software. The steps are described in the following sections.

3.1.1 Rotating Input Images

When an image is rotated by a random degree, an extra white margin in produced in the final image as



(a) Input image

(b) Rotated image with extra margin (the square in center is imaginary)

(c) Output image by ROIrotate

(d) Final output image

Figure 5. The process of rotating input image

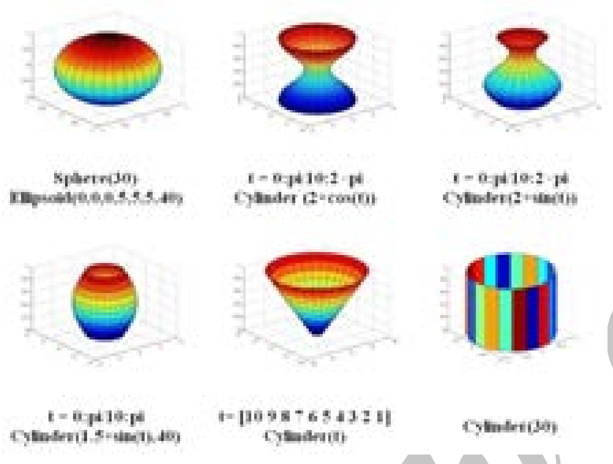


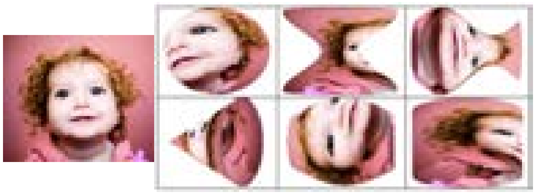
Figure 6. Selected geometric functions

can be seen in figure 5-b. To remove this extra margin some parts of output image should be cut by selecting a Region Of Interest (ROI) and rotate image to the wanted angle. For this purpose, ROIrotate function is applied to the algorithm. This function is accessible from the Matlab website [15]. Figure 5 shows the process of achieving the final rotated image step by step.

3.1.2 Geometric Transformations

Geometric transformations are a subset of mathematical transformations. A mathematical function transforms the pixels of an image to another position in page or space. These functions are various with several variables. However, we only have used 6 fixed 3D of them in this paper which is shown in Figure 6.

We used Warp function in Matlab to transform images on these 3D shapes. One input image and output instances is shown in Figure 7.



(a) Input image

(b) Output instances

Figure 7. Input image and final warped images

3.1.3 Rendering 2D images from 3D objects

The next step is creating a 2D image from the produced 3D object. When capturing a 2D image from the 3D object, we imagine the camera sight is always adjusted to the center of 3D object and the camera is turning on a fixed sphere around the 3D object. Matlab uses 2 angles to turn around a 3D object; Azimuth and Elevation. As we examined different viewpoints for acquiring usable 2D images, we realized that some of output images from particular angles are not usable for users. To prevent producing some of those unusable pictures, Elevations is adjusted between 50 and −50 degrees. Figure 8 illustrates some examples of usable and unusable images which are taken from different angles.

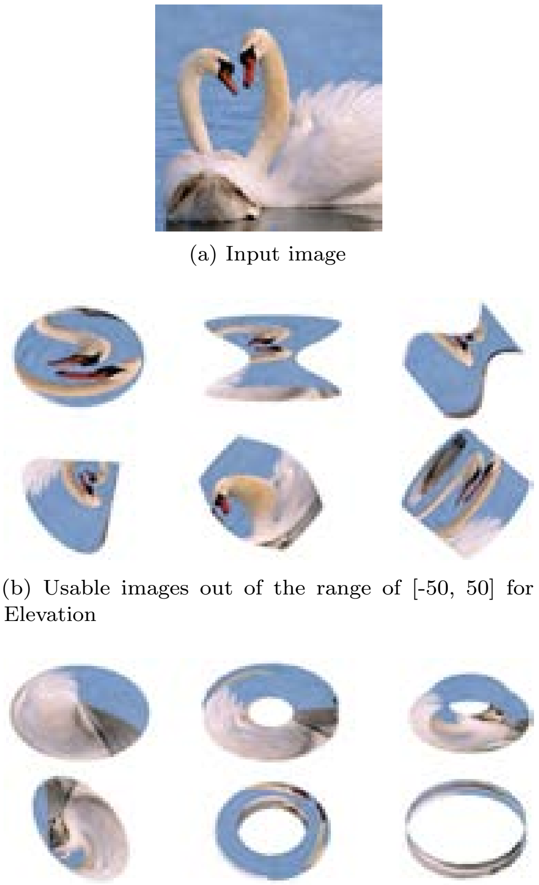
3.1.4 Improving Usability of Geometric Transformations

In order to improving usability of geometric transformations, in addition to adjusting camera angles, we proposed a heuristic function called H0 which applies a heuristic image which is a white plain image with a black mark in the center as shown below (Figure 9).

H0 image is transformed with input image concurrently. If H0 final image contains some part of this black mark, the main final image is tagged as usable in database, otherwise it is unusable. Since the center of an image is more important for human to identify the whole image, as it was predicted, H0 works. Figure 10 shows some usable and unusable images which are produced by H0. In order to a better display and understanding of final H0 images, the background color is changed to yellow and some imaginary black lines are added to the H0 image.

3.2 Tagging CAPTCHA system

As discussed before, in all steps of the algorithm, some random variables are applied to expand the search space. The original images are stored in a file. For every input image, the preprocessing algorithm generates 4 random images for all 3D transformation. Therefore, the six 3D transformation functions give us 24 final images for every input image and 720 final images are



(a) Input image

(b) Usable images out of the range of [-50, 50] for Elevation

(c) Unusable images in the range of [-50, 50] for Elevation

Figure 8. Usable and unusable images by setting camera angles

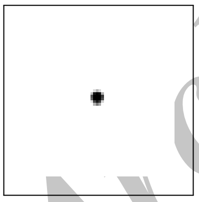
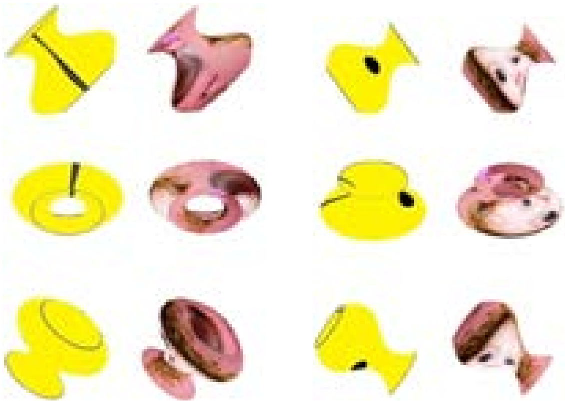


Figure 9. H0 image

created for 30 input images. Please consider that, these 720 images are only a possible subset of all images can be produced by this algorithm.

In order to evaluating the proposed methods, we tested a practical CAPTCHA system in this phase. The GUI is implemented in ASP.net and it displays an image per challenge to the user with a menu contains 30 labels. The user selects the label relating to the concept of the image. We asked 20 users to response to Tagging CAPTCHA including 10 male and 10 female, 18 to 30 years old, which were undergraduate and graduate students. The proposed Tagging CAPTCHA was new to all users. 30 images in 2 rounds were displayed to each user and feedbacks were logged into an Access Database containing 3 tables:



(a) Unusable output (b) Usable output

Figure 10. Usable and unusable output and heuristic images

- User Information Table: includes user information (Age, Sex, Field and Grade) with 20 records.
- Picture Information Table: contains image information (Name, Transformations and result of applying Heuristic H0) with 720 records.
- Feedback Table: includes users feedbacks per test (Image Name, Passed or Failed and Response Time) that has 1200 records.

These records were analyzed by a program and the results are presented in Section 4. However some interesting facts about the input images were exploited from this data which are presented in this section.

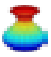
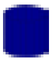
3.3 Usability of Heuristic H0

Selected input images, selected transformations and the proposed heuristic have some effects on the success rate which are discussed here:

(1) Non-centric object: H1 doesn't have a good efficiency on the images that the main subject is not in the center. For instance, in the Lighthouse image in Figure 11a, since the main distinguishable concept (the tower) is in the right side of the image, applying H0 doesn't make it too more usable.
(2) Multi concept images: images including more than one subject- causes users to select wrong tag for the final image. As it can be seen in Figure 11b, there are some Fish (the secondary object) around the Dolphin (the main subject). After applying transformations, users select the second subject in some cases.
(3) Repetitive images: when the main subject is copied several times in an image, users can response easier to it. For example in Figure 11c, 4 fishes are in one image. This image is one of the most successful images in its response rate.

Considering these tips in designing heuristic functions and in selecting input images makes the final

Table 1. Success rate of 3D objects

Number	1	2	3	4	5	6
3D object						
Without H0	77%	85%	76%	80%	82%	78%
With H0	81%	88%	87%	86%	91%	81%

CAPTCHA more usable.

3.4 Response time and Success rate

Usability of the system means how it is easy for human to response. So they are named human metrics too. Two main factors are considered as usability metrics in CAPTCHAs; Response time and Success rate. First we present the success rate of each 3D objects in Table 1. As it can be seen cone and sphere have the best and the worst response rate with H0, respectively. In addition, generally there is a rise of 7% in usability after applying H0.

Table 2 shows the information about usability metrics for the proposed Tagging image CAPTCHA in 2 different rounds. Since users are more familiar with the challenges and images in round 2 the results is better than round 1. And it is predictable it would be better in next rounds too. Users can pass the proposed Tagging CAPTCHA after one training round in 6 seconds and with the success rate of 91% which generally are good results in usability metrics for CAPTCHAs. On the other hand, these transformations are interesting for users too and they interact with the challenge as a game we are solving that. In conclusion, geometric transformations could be considered as potential options for using in image based CAPTCHAs.

3.5 Security Analysis

Security metrics are about the security of the CAPTCHA systems. Since they measure the strength of the system against the machine bots, they are called machine metrics too. Security metrics are divided into three main types; Random guessing, Direct matching and Learning attacks. We present the first and second attacks here and the third one will be discussed later.

If an attacker selects one of the tags in the menu randomly, and (s)he responses to the challenge correctly, (s)he performs a successful random guess attack. We should calculate the probability of this selection as the probability of random guessing. Since there are 30 tags in the proposed GUI, this probability is equal to 3.3%. If the challenge repeats 2 times for the user, it decreases near to 0.1% and if the GUI presents many images for example 12 ones, this could plummet to $1.66 \times e^{-18}$.

Table 2. Usability metrics

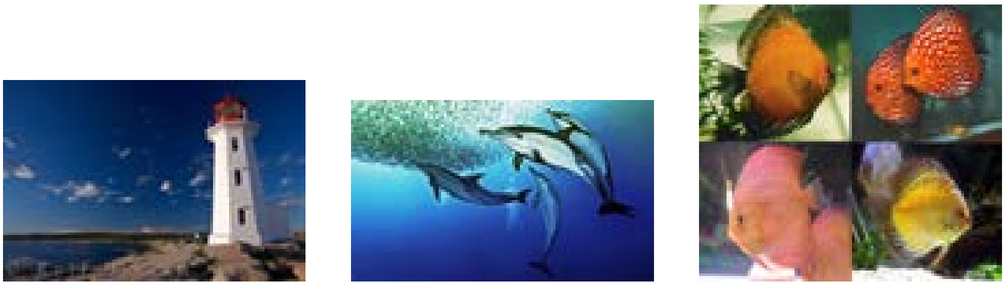
Round no.	Success rate		Average response time
	Without H0	With H0	
1	82.92%	76.83%	8.99 seconds
2	91.06%	84.33%	6.1 seconds

Table 3. Needed operations for direct matching attack

Variable name	**Number of modes**
Input Images	30
Rotation in step 2 of algorithm	360 degrees of freedom
Geometric transformations	6 — at least
Camera viewpoint	$100 \times 360 \times 360$ degrees of freedom One of the degrees is set to $[-50, 50]$ for more usable images
All possible final images without H0	840 milliards
All possible final images after applying H0	630 milliards
Needed operations for comparison 2 images	$\log_2(1200 \times 900) = 20.043$
All needed operations without heuristics	16.8×10^{12}
All needed operations after applying heuristics	12.6×10^{12}

If an attacker uses the CAPTCHA too many times or (s)he steals all or some parts of input images from database, (s)he can ask several users to exploit correct tags by paying them a reasonable cost or credit—Mechanical Turk Attack. Saving main images in DB instead of transformed ones reduces this attack considerably. Now imagine the attacker has gained some or all parts of the DB in some way. S(he) can construct a new DB as a lookup table and save all possible output images after applying transformations. S(he) should compare the displayed image with all images saved already. The minimum required operation for compacting two images is equal to $\log_2$ (image pixels). The output images in the proposed CAPTCHA system are 900*1200 pixels and based on the statistics, about 25% of images are removed after applying heuristic. Table 3 calculates all needed operations for direct matching attack.

Note that geometric transformations have much variety which is limited in this work. As it can be seen in the table, although many variables are fixed for the practical tests, there is a huge search space for the proposed Tagging CAPTCHA which can be used in



(a) Main subject on the side (b) Multiple concept (c) Repetitive image

Figure 11. Some example of input images

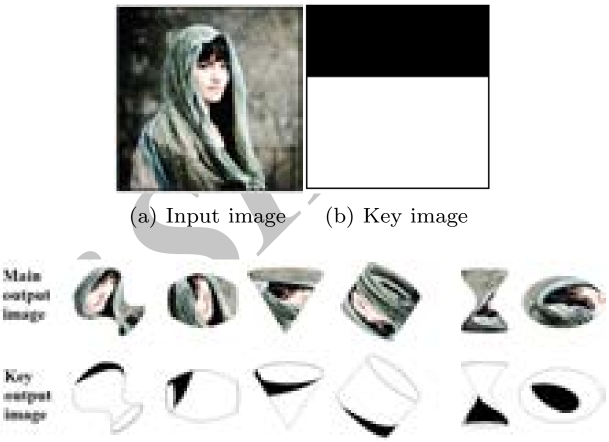
other image CAPTCHAs too.

4 SEIMCHA

From previous section, we found geometric transformations as a potential solution to use in a more practical CAPTCHA which is more usable and secure. It is undeniable that increasing round number to improve security decreases usability because of longer response time. Furthermore, the proposed Tagging CAPTCHA suffers from the general problems of the other tagging CAPTCHAs. Today users look forward to faster response mechanism to pass CAPTCHAs. Based on the previous works on upright orientation CAPTCHAs, we found this concept appropriate for designing more secure and usable CAPTCHAs. In this section we present SEIMCHA as a new semantic image CAPTCHA which is a combination of upright orientation concept and geometric transformations.

4.1 Identifying Upright Oreintation

As the first challenge in using upright orientation in a CAPTCHA, we face the problem of producing the correct answer for the test. It means that the server—which is sending the tests, should know the answer. Considering the issue that we do not save the upright orientation as a tag in DB, the server should produce it dynamically. One of the most important advantages of using upright orientation instead of labeling is that there is no need to keep something as key in database. In fact, we can design an algorithm to produce the key on the fly. We suggest a key image transforming exactly like the input image which is divided into three parts and the corresponding top part of it is considered as the answer area which could be clicked by user as correct answer (Figure 12b). Indeed, Users should click on the logical upright orientation of an image as right answer which is a specific area for the server. Please note to the fleshes in Figure 12c showing the area could be clicked by user.



(a) Input image (b) Key image

(c) Examples of main output images and key output images

Figure 12. main and key images

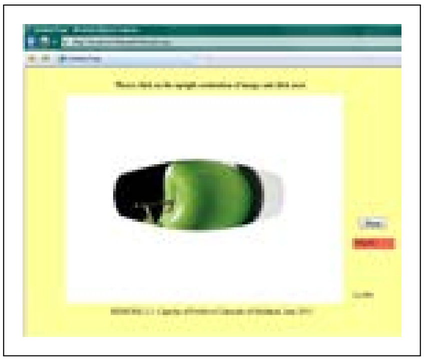


Figure 13. SEIMCHA interface

4.2 SEIMCHA System

We implemented a beta version of SEIMCHA in ASP.net. The GUI presents a series of images to users in several rounds and asks them to click on the upright orientation. The clicked point is returned to the server and is checked in the corresponding key image. Then, the user will be announced if s(he) passes or fails the challenge (Figure 13).

Again we asked 20 users to take part in SEIMCHA, 10 male and 10 female, 22 to 30 years old. They study as undergraduate, graduate and Ph.D. students in en-

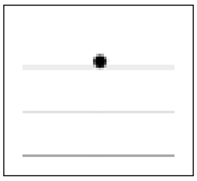


Figure 14. H1 image

gineering faculty of Ferdowsi university of Mashhad[1]. SEIMCHA was new to all users and 60 images were displayed to each user and all feedbacks were logged into an Access database containing 3 tables:

- User Information Table: includes user information with 20 records.
- Picture Information Table: Contains information about images (Image name, Transformations and H2 and H3 tag—which are two heuristic functions) with 720 records.
- SEIMCHA table: includes users feedbacks (Image name, Passed or failed and Response Time) with 1200 records.

4.3 Improving Usability of SEIMCHA

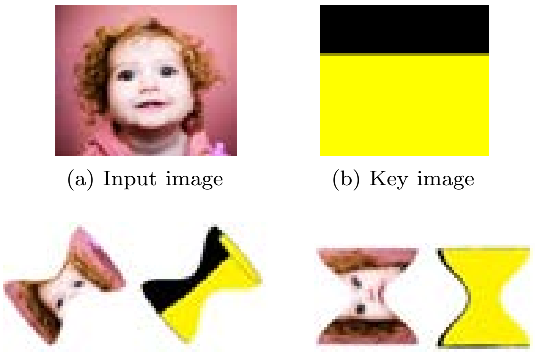
Adding upright orientation concept changes the style of response mechanism and needs new solutions to raise usability. We applied two new heuristics to improve usability of SEIMCHA. The first one focuses on a specific part of input image and the second one uses the visible correct area in the final image.

The former which is called H1 is based on H0 using a plain white image with a mark in the middle of top part of it (Figure 14). Note that the lines are imaginary and do not exist in the main image. Since the top part of an image is more important for identifying its upright, the mark is transferred to the top part of it.

Second heuristic, H2, uses the key image. After applying transformations, when the final key image is generated, a program calculates the percentage of black part of it as visible correct answer. If the correct answer area was less than 20% of color parts of whole image -not white margin parts- the final main image is marked as unusable in database. Figure 15 shows some examples of H2 output.

Consider that the image in Figure 15d is not unusable by itself. But since the visible key region is not enough to click, the user cannot response to the challenges presenting such sort of images. Foe fixing this problem, the answer area could be defined like a fuzzy variable which can be developed as a good point for future works.

[1] http://www.um.ac.ir.



(a) Input image (b) Key image

(c) Usable image (d) Unusable image

Figure 15. Example of applying H2

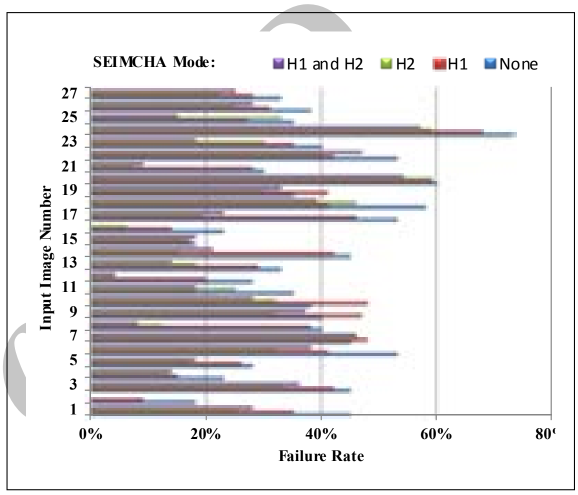


Figure 16. Difficulty rate of input images

4.4 Usability Analysis

SEIMCHA has a lot of aspects to be investigated as a variable affecting usability metrics. In this section, we report all analysis we performed on SEIMCHA feedback database.

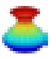
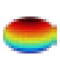
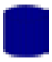
Input Images and 3D Objects analyzed failure and success rate of input images to find which sort of them are more appropriate for SEIMCHA. Figure 16 shows difficulty rate of these 30 images in 4 modes of SEIMCHA; without heuristics, with H1, with H2 and with 2 heuristics. There are four input images that their failure rate after applying H1 and H2 is still more than 50% (Figure 17). It is interesting to note that these images have multiple upright orientations. According to our previous works, users can recognize these images well [16] but they cannot identify the upright orientation of them.

Also we analyzed six 3D objects we applied in SEIMCHA. Table 4 shows usability of each 3D object. The sphere has the most failure rate before applying heuristics, but it is changed to the best shape after that. Other 3D objects have almost the same failure rate.

(a) (b) (c) (d)

Figure 17. Example of unusable images

Table 4. Usability of 3D objects (Failure Rate)

Number	1	2	3	4	5	6
3D object						
Without heuristics	40%	32%	**53%**	35%	39%	38%
With H1 and H2	32%	26%	**17%**	26%	26%	29%

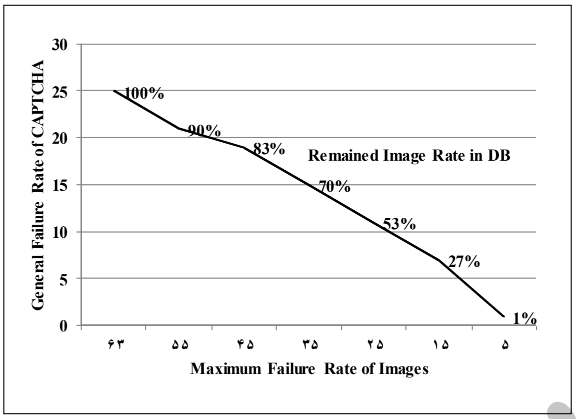


Figure 18. Effect of removing hard images on SEIMCHA failure rate and remained image rate

Success Rate and Response Time Table 5 shows SEIMCHA results for 4 modes. These results are before filtering input image database of difficult images for human.

As we discussed in the previous section, some images are not appropriate to be applied in SEIMCHA. We can remove hard images from database. Figure 18 shows the effect of removing hard images on general failure rate and remained images rate in database when SEIMCHA works with two heuristics.

As it is shown in Figure 18, we can remove the images with 25% failure rate and also more. General failure rate of SEIMCHA will decrease to 10% and it means that success rate increases to almost 90%. However in this case, we have to remove 55% of all input images and add some new images with less hardness rate.

Table 5. Failure rate of SEIMCHA before filtering input images

SEIMCHA mode	Without heuristics	H1	H2	H1 and H2
Failure rate	39%	34%	26%	25%
Success rate	61%	66%	74%	75%

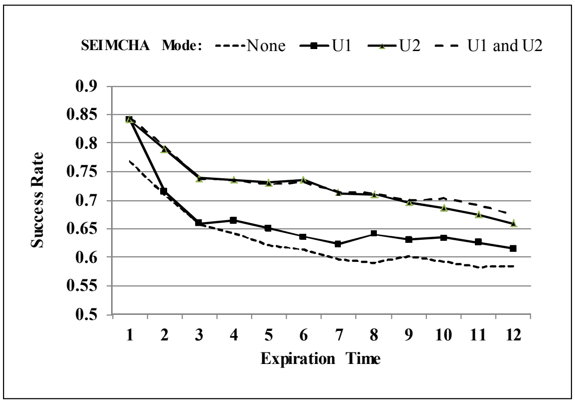


Figure 19. Relationship between success rate and expiration time

On the other hand, since simple response mechanism of SEIMCHA—single clicking—response time is too short. Table 6 shows response time average in 4 modes of SEIMCHA.

Average response time is 4.03s for all challenges and 3.81 s for success challenges. We defined an expiration time and used the time to find whether challenges with more response time improve success rate or not. Figure 19 shows success rate depend on the expiration time.

It can be concluded from Figure 19 that users usually are successful in challenges in shorter time and challenges with long response time are difficult for users and will be failed by them.

4.5 Security Analysis

We discussed 3 types of attacks before. The probability of random guessing—as the first attack—is reported in Table 7 for SEIMCHA. A final image contains some white margins between 30% and 80%, and in average about 50%. An attacker could perform some preprocessing to click on a colorful point of image. As discussed before, we adjust the minimum margin of correct answer to 20% of whole image (colorful parts). Random guessing success rate could be improved by decreasing the minimum margin of correct answer area.

This probability is for one image per test. A SEIM-

Table 6. Response time

SEIMCHA mode	Without heuristics	H1	H2	H1 and H2
All challenges	4.18 s	4.11 s	4.02 s	4.03 s
Success challenges	3.87 s	3.87 s	3.79 s	3.81 s

Table 7. Random guess attack probability

SEIMCHA mode	Without heuristic	H1	H2	H1 and H2
Percentage of answer area without any preprocessing	12.32%	13.5%	17.4%	17.5%
Percentage of answer area with preprocessing	24.62%	27.11%	34.79%	35.05%

CHA system which displayed 8 images would achieve a guess success rate of less than 0.7×10^{-6}%.

The second attack is direct matching. As we calculated in Tagging image CAPTCHA, an attacker should perform 16.8×10^{12} operations on it. When heuristics are applied in SEIMCHA, about 20% of images will be removed. So, the needed operations are 13.44×10^{12} which is a considerable time and is a great deal for the attacker.

And finally, the third kind of attacks are machine learning based ones which use some learning methods to act like a human to answer the challenge in the CAPTCHA. Fortunately, using geometric transformation functions distorts topology of point features and the shapes in the image which are two common ways to learning an image [17, 18]. In addition, adding upright orientation concept makes SEIMCHA more difficult for machine since it should identify top part of an image more than recognizing the content of it in the form of a label. However, it is important to notice that we didn't design any special machine learning system to pass SEIMCHA and we just discussed the strength of the proposed approaches in theory. Trying to design such attack systems would be interesting as another point for future works.

5 Comparison

As it mentioned in Section 2, there are some CAPTCHA systems using other transformations. 2 more similar works are Collage [8] and 2D CAPTCHAs from 3D models [9]. The former only uses rotations, but the latter changed the color, light and some other distortions on images. In addition, we introduced Assira as one of the most famous image based CAPTCHA systems which ask user to recognize cats and dogs [7]. Also What's Up [5] and Sketcha [3] are two main CAPTCHAs asking user to identify upright orientation of images. What's Up displayed input images with random rotations and Sketcha shows line drawing images rendered from 3D models. In this section, we aim to discuss usability metrics and security metrics for these CAPTCHAs and the proposed CAPTCHAs. Then, we present some interesting further experiments to compare these works.

5.1 Usability Metrics and Security Metrics

Unfortunately, there are no response time and success rate for Collage and 2D CAPTCHAs from 3D models [9, 14]. Assira has 83.4% success rate and 15 seconds for selecting 6 images out of 12 in one round. And if the challenge repeats 3 times these numbers go up to 99.96% and 45 seconds respectively [7]. Again there is no response time for What's Up but it is 35 seconds for 10 images in Sketcha. Success rate is 84% for 3 images in What's UP and 88% in Sketcha [3, 5]. Finally in this work, users are successful is just over 91% and in about 6 seconds in Tagging CAPTCHA and about 90% and in 4 seconds for SEIMCHA.

Also random guessing attack in Assira is 0.39% for 8 and 0.024% for 12 images. This is about 16% for Collage while it is about 3.3% for 2D CAPTCHAs from 3D models and the proposed Tagging CAPTCHA for 1 image. It is 25% for 1 image in Sketcha and it decreases to 0.001% for 8 images. The probability of random guessing is 4.44% for 1 image in What's UP and again it reduces to 0.009% for 3 images. And finally, it is 17.5% for the proposed SEIMCHA system based on the statistics in the log database.

One of the most important things makes a CAPTCHA system interesting to users is its response mechanism. Assira and Collage are interesting since their GUI provide users a single click mechanism. But 2D CAPTCHAs from 3D models and the proposed Tagging CAPTCHA are weak in this property because they ask user to search in a list for the proper label. User's response **type** of What's Up is image rotation using slider or mouse movement or up-down control. Sketcha requires the user to rotate each image in a set of drawing until everyone is upright, by clicking to turn them 90 degrees at a time. These tasks need more time than a single click on the image which is provided in SEIMCHA. Table 8 summarizes these comparisons.

5.2 Further Experiments

It is possible to automatically identify the images by using reverse indexing image search engines like TinEye.com. If we want to categorize it as an attack, it is a kind of direct matching. TinEye finds exact and altered copies of the images that you submit, including those that have been cropped, color adjusted, resized, heavily edited or slightly rotated [19]. Indeed, TinEye produces a digital signature of finger print for submitted image and compare it with all finger prints of saved images. We examined the strength of TinEye to find rotated images. In this experiment, a rotated image in the range of [-20, 20] was submitted to tineye.com

Table 8. Usability and Security Metrics Comparison

CAPTCHA	Response time	Success rate	Random guessing	Response mechanism
Assira [7]	15s / 1 round 45s / 3 rounds	83.4% / 1 round 99.96% / 3 rounds	0.39% / 8 images 0.024% / 12 images	Single click / 1 image Multiple click / 1 round
Collage [8]	Not reported	Not reported	16% / 6 images	Multiple click / 1 object
2D from 3D [9]	Not reported	Not reported	3.3% / 1 image	Selecting label from list
Tagging CAPTCHA	6s / 1 image	91%	3.3% / 1 image	Selecting label from list
What's Up [5]	Not reported	84% / 3 images	4.44% / 1 image 0.009% / 3 images	Slider / Moving mouse / Up-down control
Sketcha [3]	35s / 10 images	88%	25% / 1 image 0.001% / 8 images	Multiple click / 1 image
SEIMCHA	4s / 1 image	90%	17.5% / 1 image	Single click / 1 image

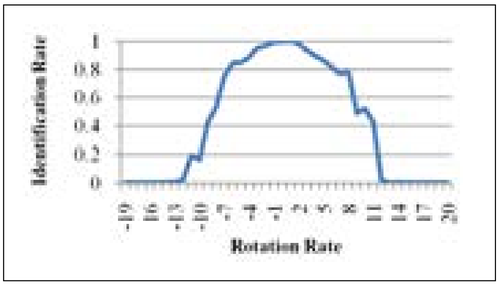


Figure 20. Identification rate by tineye.com based on rotation rate

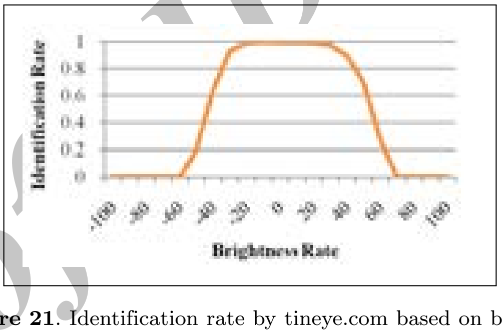


Figure 21. Identification rate by tineye.com based on brightness rate

step by step. TineEye can find rotations between -15 to 15 degrees which can be seen in Figure 20.

However it is easy for a program to rotate an image in different degrees, and then submit to TinEye and find the main image.

Again the above experiment was repeated for brightness. We changed the image light from -100% to 100% from dark to light. As it can be resulted from Figure 21, tineye.com can recognize the distorted image in a high rate. It is noticeable that images out of [-60%, 60%] are not recognizable by human. Then tineye.com can find them better than human in some cases.

The experiment has been done with the geometric transformations proposed in this work. Fortunately, tineye.com couldn't find any version of the input image which is a promising result for this work.

Today, designers consider such attacks to evaluate a CAPTCHA system [2]. It can be concluded from above experiments that Collage, 2D CAPTCHAs from 3D models, Assira and What's Up are weak to tineye.com. whereas, Sketcha and SEIMCHA are robust to this search engine.

6 Conclusion and Future Work

A common approach to improve the security of an image based CAPTCHA is to display a transformed version of an image to user instead of the main image saved in database. Geometric transformations were presented as a new successful solution in this paper since however many variables were fixed for our experiments, the final search space is still too large to traverse by an attacker in a direct matching attack. Usability of the Tagging image CAPTCHA proposed based on geometric transformations is better than the similar works and the transformations increase security too. As it can be seen in table 8 all metrics except the probability of random guessing is improved in Tagging image CAPTCHA.

Furthermore, we presented a new semantic image based CAPTCHA named SEIMCH which is a none-tagging CAPTCHA using upright orientation and geometric transformations. Applying these two, provides a more usable and secure practical CAPTCHA which eliminates problems of the proposed Tagging image CAPTCHA. SEIMCHA has a simple response mechanism -single clicking- which makes it faster than similar works. Finally, by selecting an appropriate set of input images, SEIMCHA has an excellent response time and success rate which is about 4 second per each image and about 90%, respectively. This gives the confidence to extend the proposed approaches.

As future works, we suggest below:

- Using more mathematical functions for transforming images and exploiting the best set of them.
- Using a fuzzy method to identify upright orientation instead of a 0 and 1 mechanism.
- Designing a Multiple SEIMCHA which shows several images in the GUI instead of one image to reduce the probability of random guessing attack.
- Applying a mechanism to update the database of input images continuously to improve security and also to replace images with more usable images.
- Designing an 'almost right' response mechanism instead of complete right answer. It would improve security more than before [7]. Imagine SEIMCHA displays 3 images to users. A complete right answer means 3 correct identifications and an almost right answer means 2 correct answers out of 3 images.
- Designing a particular attack on SEIMCHA that uses several types of attacks.
- And finally, studying SEIMCHA in the context of a working website involving vast number of users.

References

[1] 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 (SOUPS)*, page 14. ACM, 2009.

[2] L.V. Ahn, M. Blum, N.J. Hopper, and J. Langford. Captcha: Using hard ai problems for security. In *Proceedings of the 22nd international conference on Theory and applications of cryptographic techniques*, pages 294–311. Springer-Verlag, 2003.

[3] S.A. Ross, J.A. Halderman, and A. Finkelstein. Sketcha: a captcha based on line drawings of 3d models. In *Proceedings of the 19th international conference on World wide web*, pages 821–830. ACM, 2010.

[4] AA Chandavale, AM Sapkal, and RM Jalnekar. A Framework to analyze the security of Text based CAPTCHA. *International Journal of Computer Applications IJCA*, 1(27):127–132, 2010.

[5] R. Gossweiler, M. Kamvar, and S. Baluja. What's up captcha?: a captcha based on image orientation. In *Proceedings of the 18th international conference on World wide web*, pages 841–850. ACM, 2009.

[6] M.H. Shirali-Shahreza and M. Shirali-Shahreza. Persian/arabic baffletext captcha. *Journal of universal computer science*, 12(12):1783–1796, 2006.

[7] J. Elson, J.R. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. *CCS*, 7:366–374, 2007.

[8] R. Soni and D. Tiwari. Improved captcha method. *International Journal of Computer Applications IJCA*, 1(25):107–109, 2010.

[9] M.E. Hoque, D.J. Russomanno, and M. Yeasin. 2d captchas from 3d models. In *SoutheastCon, 2006. Proceedings of the IEEE*, pages 165–170. IEEE, 2005.

[10] The CAPTCHA Project. http://www.captcha.net/captchas/pix/.

[11] T. Pavlidis. Why meaningful automatic tagging of images is very hard. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1432–1435. IEEE, 2009.

[12] L. Zhang, M. Li, and H.J. Zhang. Boosting image orientation detection with indoor vs. outdoor classification. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 95–99. IEEE, 2002.

[13] S. Lyu. Automatic image orientation determination with natural image statistics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 491–494. ACM, 2005.

[14] M. Shirali-Shahreza and S. Shirali-Shahreza. Advanced collage captcha. In *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, pages 1234–1235. IEEE, 2008.

[15] Vlad Atanasiu. ROIRotate Function; A Function to Fill Corners of Rotated Image. Available from http://www.mathworks.com/matlabcentral/fileexchange/1825, Updated 2008.

[16] M. Mehrnejad, A. Ghaemi, A. Harati, and E. Toreini. A new image based CAPTCHA based on geometric transformations. In *8thInternational ISC Conference on Information Security and Cryptology*, FUM, Iran, 2011.

[17] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[18] S. Belongie and J. Malik. Matching with shape contexts. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pages 20–26. IEEE, 2000.

[19] TinEye. http://www.tineye.com/faq.

Maryam Mehrnejad was born in 1986 in Sabzevar, Iran. She received her BS and MS in Computer Engineering from Ferdowsi University of Mashhad (FUM) in 2009 and 2011, respectively. She was a member of Security Information and Communication Lab in FUM and also a member of FUM CERT Lab during her studies. Her main research interests are Security Engineering, HCI-Sec (Human and Computer Interaction and Security) and Applied Soft Computing.

Abbas Ghaemi Bafghi was born in April 1973 in Bojnord, Iran. He received his BS degree in Applied Mathematics in Computer from Ferdowsi University of Mashhad, Iran in 1995, and MS degree in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran in 1997. He received his PhD degree in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran in 2004. He is member of Computer Society of Iran (CSI) and Iranian Society of Cryptology (ISC). He is an assistant professor in Department of Computer Engineering, Ferdowsi University of Mashhad, Iran. His research interests are in cryptology and security and he has published more than 50 conference and journal papers.

Ahad Harati was born in 1978. He received his BS in Computer Engineering from Amirkabir University of Technology (2000) and his MS in Artificial Intelligence and Robotics from University of Tehran (2002). In 2003, he joined Autonomous System Laboratory at Swiss Federal Institute of Technology in Lausanne (EPFL) and two years later along with other colleagues moved to Zurich. He got his PhD in Robotics in 2008 from ETHZ (Swiss Federal Institute of Technology in Zurich). Later he moved back to Mashhad and joined Ferdowsi University of Mashhad, where he is currently an Assistant Professor. His main research interests include Range Data Processing and Multiresolution Analysis, Image Processing and Vision, Simultaneous Localization and Mapping, Human Machine Interaction, and Multiagent Learning.

Ehsan Toreini was born in September 1984 in Ghazvin. He is MS graduate of Islamic Azad University, Mashhad Branch in 2010 and BS graduate of Ferdowsi University of Mashhad in 2007. He is now a lecturer in Islamic Azad University, Mashhad Branch and member of Young Researchers' Club. His main fields of study are Data Mining, Machine Learning and Computational Intelligence.