Advanced Character Collage CAPTCHA

Goran Martinovic, Zdravko Krpic

Faculty of Electrical Engineering, Josip Juraj Strossmayer University of Osijek,
Cara Hadrijana bb, 31000 Osijek, Croatia
goran.martinovic@etfos.hr; zdravko.krpic@etfos.hr

Abstract: Text based CAPTCHA systems are widely used as a security mechanism for web access control. Considering their broad use, many attacks are challenging them every day. Most of the attacks aimed at CAPTCHAs are based on the latest computer vision techniques, AI methods and OCRs, so it is imperative to enhance these methods even more. There are number of proposals for CAPTCHA security, but it is hard to achieve a good balance between CAPTCHA practicality and its security. Advanced Character Collage CAPTCHA is a highly random novel method which uses the strengths of unbroken CAPTCHAs along with the weaknesses of present ones, and relies on imperfection of computer vision techniques. The proposed CAPTCHA is generated through a series of unique creation steps, each of them implementing carefully analyzed features in order to increase human recognition rate, and at the same time, to reduce computer recognition rate. The degree of recognition within the proposed method is evaluated using several tests, while its readability by humans is tested through two surveys.

Keywords: CAPTCHA; recognition rate; security; web access control

1 Introduction

Security is a major concern on web exposed systems holding valuable data or something that can be compromised. There are many types of attacks that can be carried out on these systems. A variety of bots, spiders, DOS attacks, domain hijacking, cache poisoning, worms and spam pose a serious threat to online systems and can cause major losses. Therefore, it is imperative that these systems have the most reliable security systems. Besides encryption, secure connections and protocols, there is one portion of authorization system where the computer has to decide: "Human user or computer bot?" If the user is human, an access is granted, possibly to very important data, money or goods. CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a test which distinguishes whether a user is human or computer bot. There are many different types of CAPTCHAs, most of them including a small image from which the user has to decipher the letters and type them in a small form box in order to identify himself as a human to be granted an admission to a certain part of

the web site. CAPTCHA security is mostly used in web sites which include an e-mail account creation, web rating systems, polls, search entries, forum posts, downloads and many other in order to prevent malicious users from spamming, distributing copyrighted and stolen material, inducing inflation or deflation of rankings and polls, and similar unwanted actions. The fact that CAPTCHAs presently protect many systems makes them a desirable target for everyday attacks by using various machine vision and AI techniques. There are number of specialized attacks which aim at CAPTCHA security, such as OCR and non-OCR based attacks, statistical based methods, structural analysis, wavelet fractal feature extraction, neural network "divide and conquer", and other various AI-based procedures. Even a new threat, called 3rd party attacks, has emerged, which uses cheap human labor to manually solve CAPTCHAs in order to create thousands of various accounts, polls, spam, etc. Furthermore, 3rd party attacks are used to create databases of CAPTCHA-solution pairs for finite-state CAPTCHAs (combinations of which can be exhausted in a feasible amount of time) as an input for brute-force attacks. CAPTCHA security systems include different visual and non-visual CAPTCHAs which are presented to the user, and then he has to identify or compare certain images, retype the presented distorted letters or words, or type the letters heard in a sound CAPTCHA. Even a combination of visual and non-visual CAPTCHA is possible, enabling use by people with disabilities, although CAPTCHAs based on sensory abilities cannot be used on sensory-impaired human beings, as stated in [1].

The rest of the paper is organized as follows: Section 2 enumerates some of the most important related works that have led to many ideas proposed in this paper. Section 3 covers the proposed method in detail, including generation and implementation of the proposed method. Furthermore, in Section 4, readability features of the proposed method are evaluated through a couple of surveys, while the security features are analyzed in Section 5. Section 6 shows plans for future research, upgrades and plans based on this paper, and Section 7 concludes the paper.

2 Related Work

Since this paper analyzes only image-based CAPTCHAs, which are the part of visual CAPTCHA systems, only a fraction of the vast related work from this area will be mentioned.

Authors in [2] use the term Collage CAPTCHA for a three-step process of authorization, in which the user must choose the correct image and the name of the object on the image, and only then he is granted an access to the third step, which is entering the image name into a text box. Collage CAPTCHA is considerably secure, but the major flaw of this system is its usability, considering the length of the process, and the fact that error chance by the user is multiplied.

Also, the CAPTCHA alone is relatively easy to solve. An interesting concept is proposed in [3], in which the discernment between people and bots is done by the means of recognizing strangeness in a machine translation. The differentiation of this method is excellent, but the limited number of sentences and language dependability, as well as exposure to 3rd party solvers, bound this method to limited use. A 3D CAPTCHA [4] is a promising technique, in which the authors propose different implementations of 3D letters to deceive bots, with 80% overall hit-rate by humans and a relatively complicated generation (DirectX). One of the hardest methods for bots to decipher is proposed in [5], which is based on animation with moving letters presented to the human. The method protection is very good but the advancing OCR techniques and slow and complicated implementation limit its practicality. In [6] the authors propose kernels to break different common types of CAPTCHA, accentuating major flaws, such as susceptibility to line removal algorithms, letter pattern matching, dot removal, binarization, etc., all of which are more or less absent from the method proposed in this work – Advanced Character Collage CAPTCHA. They managed to solve EZ-Gimpy with 88% success rate.

Other related papers include various novel approaches, such as [7, 8], and methods of breaking visual CAPTCHAs [9, 10].

3 Proposed Algorithms

Various authors have proposed different CAPTCHA classifications, but the most common one divides CAPTCHA systems in to visual and non-visual. Most of the non-visual CAPTCHAs are based on sound, making them less secure than visual ones due to the high-quality voice recognition and noise removal programs, as mentioned in [7, 11]. Visual CAPTCHAs, on the other hand, can be divided into OCR and non-OCR based ones, as proposed in [2, 4, 5, 12]. Non-OCR CAPTCHAs are mostly image-based, a concept which many authors, e.g. [2, 7, 12], consider to be the future of CAPTCHA protection, or at least an important part of it. The same authors claim that these CAPTCHAs do not cause dissatisfaction to its users, as most of the OCR-based ones do, but they are the most susceptible to 3rd party attacks because of databases with a limited number of images. Even Asirra[1], the most famous example of an image based CAPTCHA, with the largest image database, was broken by the authors in [13] using machine learning techniques. The authors in [1] made a good point when they said that, "There is no way to prove that a program cannot pass a test which a human can

[1] Asirra is an image based CAPTCHA which uses one of the largest lost pets database in the world (http://www.petfinder.com) to generate an image query for a human to solve. A human has to distinguish between cats and dogs. More at http://research.microsoft.com/en-us/um/redmond/projects/asirra/.

pass, since here is a program – the human brain – which passes the test". So, the goal is not to make a computer-unsolvable CAPTCHA (which is impossible), but to create a CAPTCHA system which is difficult to solve for a computer and easy for a human. It is enough to make novel CAPTCHAs better than other CAPTCHAs to divert attacks from the system which it protects. In our proposed work, the goal was to make a simple CAPTCHA that is easy to generate and easy to implement on a variety of platforms, and at the same time, that provides significant resistance to computer vision attacks. However, in order to maintain high recognition rate by humans, and at the same time, to deny computer bots deciphering CAPTCHAs, the best approach is to try to implement as many strengths of the existing strong CAPTCHAs and to avoid as many of their flaws as possible. Therefore, the best method to use is to learn from previous experience in CAPTCHA systems, as well as from machine vision and state-of-the-art AI.

The property of CAPTCHA which enables recognition by humans (RBH) is the distinction between characters, background and the clutter. Moreover, the same property is used by computer vision to decipher CAPTCHA, but in a different way. Human perception is associative, and therefore this fact should be more exploited. This knowledge gives an important but often overlooked postulate: the characters do not need to be entirely visible to facilitate RBH and at the same time deny recognition by a computer vision (RBC) due to the lack of the character integrity. In the following subsections, Advanced Character Collage CAPTCHA creation steps will be analyzed in detail, resulting in the complete CAPTCHA generation algorithm presented at the end of the section.

3.1 Strengths and Weaknesses

The major weaknesses of present OCR-based CAPTCHAs, as pinpointed in [14], can be: constant font, aligned glyphs, constant glyph position, no deformation, constant colors, no perturbation, constant background, non-textured background, weak color variation, etc. So the proposed CAPTCHA should avoid all these flaws as much as possible. The method proposed in [15] has undergone some major changes in order to fulfill security demands as much as possible, while retaining high RBH and easiness of implementation. Most aspects of the proposed method were analyzed and improved, and for every aspect there follows a description. The major change of the proposed CAPTCHA is the use of an edge detection filter, which facilitates two major improvements: resistance to assorted color segmentation attacks and usability by color blind people. Another characteristic which is omitted from our proposed method is that of using a finite set of CAPTCHA images, as they can be easily classified by the 3rd party attacks. A different improvement of the method from [15] is proposed in [16], retaining the color based CAPTCHA.

3.2 Background

The proposed CAPTCHAs were made on a 640x190 pixel white background canvas. Any color can be used for the background, but lighter colors increase RBH. The background is composed of basic geometric shapes (rectangles, circles and semicircles) in order to increase curve similarity with the characters which are going to be placed on the clutter. These shapes are painted with various semitransparent pale colors with reduced contrast, they are randomly sized, their placement is random, and they overlap. Semi transparency ensures better clutter, especially if edge detection is applied to it afterwards. Shape size and a color palette are limited by a certain threshold. Moreover, shapes can be rendered randomly, or a database of these shapes can be used, from which shapes are randomly chosen and copied at various locations on a canvas. We propose random generation of shapes, thus avoiding the need for their external storage. If an external storage is used, it can also be utilized for a CAPTCHA buffer, a concept which is described in subsection 5.4. The number of generated shapes is also bound to a certain threshold based on the canvas and shape size, because there should be enough shapes to saturate the background, but not too many, in order to avoid oversaturation and thus making characters more distinguishable to computer bots, and less visible to humans.

3.3 Character Composition

Allowed characters are random clear type font letters (uppercase and lowercase) and digits 0-9. Bold or very thick fonts should be avoided because they cause readability issues. After the background has been generated, it is split into r vertical regions, $R_{i,vert}$, where r is the number of characters in one CAPTCHA, $4 \leq r \leq 6$. In our experiment, font face and size were constant. An outlined character-shaped mask is placed on each region $R_{i,vert}$, and the masked region of a character is then copied to region $R_{i,vert}$ of another character, and vice versa. That way, characters are composed of the same texture as the background clutter. Furthermore, the character-shaped mask is meshed into regions based on the previously designed texture beforehand, with the intention of avoiding pixel continuity. The mesh texture lines should be sufficiently thick to separate characters into pieces, but not so thick as to reduce RBH. Our research has shown that optimal meshed texture line is approximately 6 pixels thick, and the example of the used mesh texture can be seen in Fig. 1.

Optionally, before placing meshed character-shaped masks on the background, r regions with greatest color difference $R_{i,color}$ could be found on the background. These regions can serve as placeholders for character masks before copying, increasing readability on both the color and grayscale versions of the Advanced Character Collage CAPTCHA.

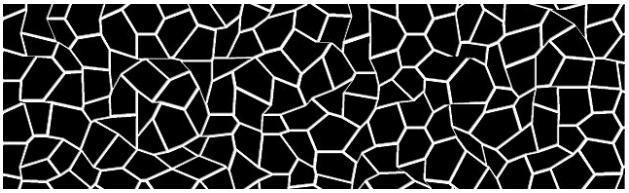


Figure 1
Mesh texture used in the experiment

3.4 Character Placement

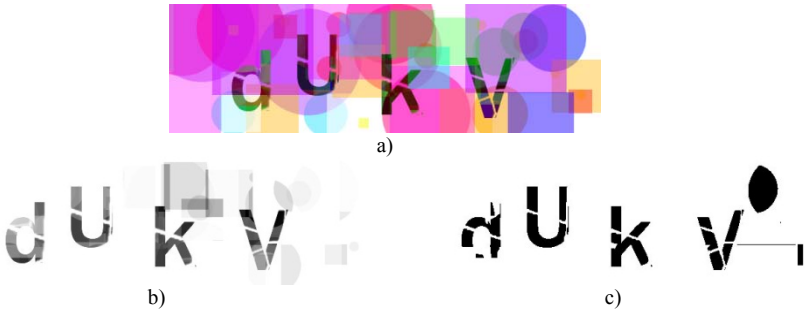
A common method to avoid one of the major weaknesses of the existing CAPTCHAs, which is constant glyph alignment and rotation, is to apply mild warp to a glyph-shaped mask, along with slight random rotation (up to 30º in an arbitrary direction), which is different for every glyph, although some authors propose up to 45º [4]. In our work, rotation and warp were neglected, because the background was composed only of straight lines and circles; warped lines would be too prominent after line removal preprocessing methods and would compromise security of the CAPTCHA. However, glyph deformation analysis will be a part of our future work. Letter placement in a region is random, considering that the whole glyph is visible, i.e. is not outside the canvas, by the means of using random offset values from the centre of the region.

3.5 Edge Detection

The major change to the proposed method based on the work done in [15] was to apply an edge detection filter to the image, which also converts image to grayscale. The main reason for such a change was insufficient resistance to color segmentation attacks, which could easily separate glyphs from the background, as seen in Fig. 2. Fig. 2a shows the first implementation of the Advanced Character Collage CATPCHA; there is an exclusion operation applied between glyph layer and the background layer, resulting in a high RBH. However, if the color channel mixing is applied, the glyphs can easily be isolated by computer vision, as shown in Fig. 2b. Another major weakness is brightness and contrast tuning, the implementation of which can lead to an even better glyph segmentation by computer, Fig. 2c.

Edge detection, in addition to providing greater resistance to attacks, allows color blind people to solve the CAPTCHA, thus spreading the pool of potential users.

The illustration of the Advanced Character Collage CAPTCHA creation steps can be seen in Fig. 3. Fig. 3a shows a white background canvas that is saturated with random shapes, shown in Fig. 3b.



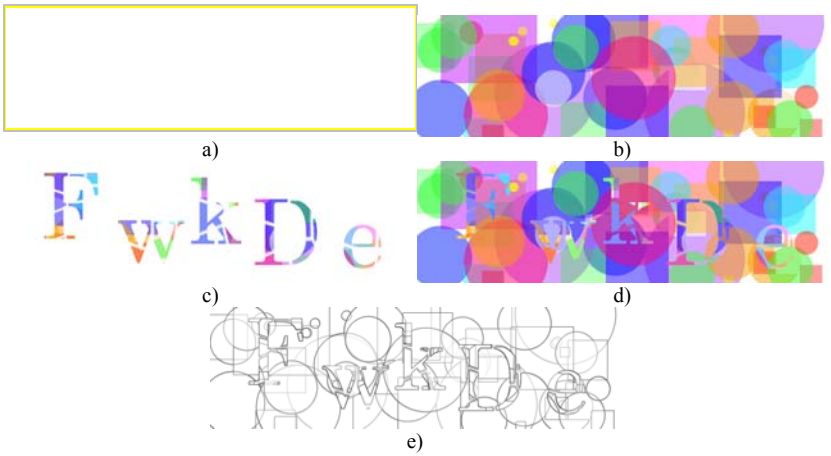
a) b) c)

Figure 2

Character Collage CAPTCHA from [15]: a) original image, b) after using color channel mixer, c) after contrast and brightness tuning

The glyph shaped regions are placed onto saturated background and copied, Fig. 3c, put back to the corresponding background regions $R_{i,vert}$, Fig. 3d, and finally by applying the edge detection result in the CAPTCHA image, shown in Fig. 3e.

With everything taken into consideration, a complete Advanced Character Collage CAPTCHA algorithm can be proposed, that is illustrated in Fig. 4.



a) b) c) d) e)

Figure 3

Advanced Character Collage CAPTCHA creation steps: a) white canvas, b) background clutter, c) meshed glyph shaped regions, d) glyph shaped regions put onto the background clutter, e) resulting CAPTCHA

1. *Open canvas.*
2. ***For*** *i=0 to* $\mathbf{m}$, $\mathbf{m} \in [min\_saturation, max\_saturation]$
   - 2.1. *Generate two numbers* $\mathbf{x}$ *and* $\mathbf{y}$ *such that* $\mathbf{x} \in [0, image\_height]$, $\mathbf{y} \in [0, image\_width]$.
   - 2.2. *Create randomly sized shape and paint it with randomly generated color.*
   - 2.3. *Place the centre of the shape at coordinates* $(\mathbf{x}, \mathbf{y})$.
3. *Generate random number* $\mathbf{r}$ *such that,* $4 \leq \mathbf{r} \leq 6$.
4. *Split the canvas into* $\mathbf{r}$ *vertical regions* $\mathbf{R_{i,vert}}$.
5. *Calculate the central coordinates* $(\mathbf{x_i}, \mathbf{y_i})$ *of every region* $\mathbf{R_{i,vert}}$.
6. ***For*** *i=0 to* $\mathbf{r}$
   - 6.1. *Choose random character* $\mathbf{c_i}$.
   - 6.2. *Create character shaped mask.*
   - 6.3. *Load meshed texture from a file.*
   - 6.4. *Apply random slight warp to the loaded texture.*
   - 6.5. *Permeate the character shaped mask with the resulting mesh texture.*
   - 6.6. *Create offset* $\mathbf{p}$ *and* $\mathbf{q}$.
   - 6.7. *Place the centre of the mask into coordinates* $(\mathbf{x_i} + \mathbf{p}, \mathbf{y_i} + \mathbf{q})$.
   - 6.8. *Copy masked area.*
   - 6.9. *Select random region* $\mathbf{R_{j,vert}}$, $j \neq i$.
   - 6.10. ***If*** *selected regions flag* $\mathbf{s_j} \neq 0$
     - 6.10.1. *Repeat selection*
   - 6.11. ***Else***
     - 6.11.1. *Paste copied masked area to selected region, with its centre aligned to the offset* $(\mathbf{x_j} + \mathbf{p}, \mathbf{y_j} + \mathbf{q})$.
     - 6.11.2. *Set selected regions' flag* $\mathbf{s_j}$ *to 1.*
     - 6.11.3. *Put character* $\mathbf{c_i}$ *into CAPTCHA string at* j^{th} *position.*
7. *Apply edge detection filter on the whole image.*
8. *Save as image.*
9. *Save CAPTCHA string as an image pair for comparison.*

Figure 4
Advanced Character Collage CAPTCHA generation algorithm

4 Readability Survey

RBH is the most important feature of the CAPTCHA system. Consequently, humans have to test the CAPTCHA to facilitate its readability properties. Two surveys have been conducted to eliminate features which reduce human recognition rate. Both surveys were attended by random groups of people in a way that they have been given CAPTCHA tests to solve them.

4.1 First Survey

In the first survey, participants were not exclusively informed that the CAPTCHA was case-sensitive. A survey consisted of 24 CAPTCHAs, which all included letters a-z and A-Z, and were of a length between 4 and 5 characters ($4 \leq r \leq 5$). Digits were not used. A sample of a given CAPTCHA test is given in Fig. 5.

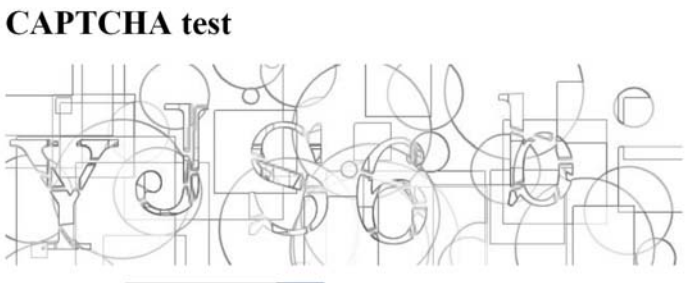


Figure 5
Example of a survey test

110 anonymous random people did the test, so the demographic data is not available/not known. The results of the survey are presented in the Table 1. The overall CAPTCHA hit-rate was 62.7%, which is a good result considering the lack of fine tuning. 65.5% of all errors were caused by inability to determine the glyph case, while the remainder relates to false glyph recognition. There were no obscure glyphs.

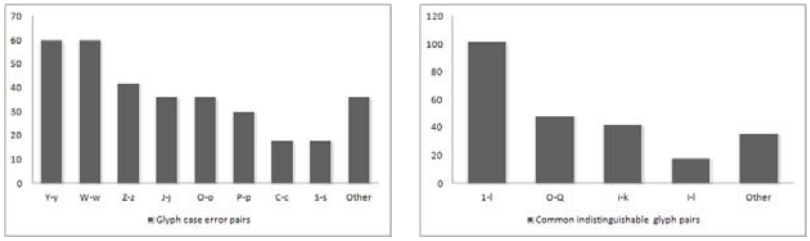
Table 1
Results of a first survey

Overall performance	**Hit rate**	[%]	62.7
	Average solving time	[s]	9.1
No. of unidentified glyphs per CAPTCHA	**1-glyph miss**	[%]	59.8
	2-glyph miss	[%]	24.4
	3-glyph miss	[%]	11.6
	4-glyph miss	[%]	4.3
Glyph error types	**Glyph case miss**	[%]	65.5
	Glyph miss	[%]	34.5
	Obscure glyphs	[%]	0.0

Most of the incorrectly recognized CAPTCHA tests had a single glyph miss (59.8%), and 24.4% of all faulty tests had two-glyph miss. Moreover, multiple glyph misses were mainly caused by the fact that users did not know that tests were case-sensitive, therefore resulting in a 11.6% share of three-glyph misses, and even a 4.3% share of four-glyph misses in the total human recognition error. It is estimated that approximately 13% of all solved CAPTCHAs were falsely recognized because of the above reason.

Other multi-glyph misses were caused by problematic glyph pairs, which can be divided into two groups. The first group consisted of glyphs whose uppercase and lowercase versions are difficult to discern. These glyph pairs are shown in Fig. 6a, with their respective shares in the overall human recognition error. The second group of glyph pairs consisted of similar letters, and their share in the total error is shown in Fig. 6b. The first survey has shown that the hit rate can be improved by

avoiding the mentioned glyphs pairs, which would then improve RBH, but it would also source a smaller CAPTCHA combination space. The negative effect of these improvements can be avoided by increasing the glyph pool, which will be considered in survey 2. Another way to improve RBH is to use fonts which offer greater dissimilarity between letters in the problematic pairs, such as console fonts or old style fonts. Additionally, the hit rate can be further improved by using letter mask region select technique proposed in subsection 3.3, based on $R_{i,\ color}$ regions.



a) b)

Figure 6
Error intensity for the problematic glyph pairs

Analysis of the average time for a human to solve the proposed CAPTCHA shows that it is not time consuming, with an average human solving time of only 9.1 seconds.

4.2 Second Survey

In the second survey, 104 participants were informed that the given CAPTCHA is case-sensitive. In addition to that, the elimination of some problematic glyph pairs was done; for example, uppercase glyphs "I" and "O" and lowercase glyphs "l" and "q" were removed from the glyph pool. A different font (Century) was chosen to accommodate greater difference between glyphs in other problematic glyph pairs. Additionally, to increase the glyph pool, and therefore to reduce the risk of brute force attacks, digits 1-9 were added to the glyph pool. "0" was left out intentionally because of the similarity with the letter "O". The CAPTCHA length was increased from $4 \leq r \leq 5$ to $4 \leq r \leq 6$. Table 2 shows the performance and error analysis of the second survey. The overall hit rate was improved to 89.9%, which is a very good result. The average solving time was increased by 0.4s, but if the increased number of glyphs is taken into consideration, this increase is negligible. It is worth mentioning that the inability to determine whether the letter is uppercase or lowercase is still a major cause of CAPTCHA recognition errors (40.9%), but this time other letter recognition errors are almost equally present (37.7%). There were a number of situations where one glyph was significantly less visible than the other, which caused 14.3% of all errors. Finally, there were several occasions (with 7.1% share of errors) where participants mistyped the number (0 and 9 most of the time) because of their keyboard location.

Table 2
Results of a second survey

Overall performance	**Hit rate**	[%]	89.9
	Average solving time	[s]	9.5
No. of unidentified glyphs per CAPTCHA	**1-glyph miss**	[%]	96.4
	2-glyph miss	[%]	3.6
	3-glyph miss	[%]	0.4
	4-glyph miss	[%]	0.0
Glyph error types	**Glyph case miss**	[%]	40.9
	Glyph miss	[%]	37.7
	Obscure glyphs	[%]	14.3
	Number glyph mistype	[%]	7.1

5 Security and Performance

Since there are no available tools for testing the CAPTCHA resistance to AI attacks, deciphering steps are analyzed from related and previous work. That way, the most common attack routines can be isolated and simulated in order to apply them to the proposed method. Unfortunately, the best CAPTCHA benchmark is real-world use, i.e., when it draws enough attention, as mentioned in Section 3. Most of the researchers [4, 6, 11] agree that the procedure for deciphering CAPTCHA can be divided into three main steps: Preprocessing, Segmentation and Classification, although some authors [10] include an additional step before the last – Feature extraction. The preprocessing part of the process converts the CAPTCHA to grayscale and removes any noise and background. After the image passes through the preprocessing step, segmentation is applied, which separates regions on the image which (should) contain glyphs. The optional next task is to extract unique features of the characters (number of holes, height of character, etc.) to further enhance the last step, which is character recognition. Finally, OCRs are applied for character recognition.

The CAPTCHA proposed in [15] was susceptible to certain image manipulations which did manage to successfully isolate glyphs in several cases. For example, these were mixing color channels, contour tracing algorithms, brightness and contrast adjustments and custom edge detection tools, which served as the guidelines for enhancements to the Advanced Character Collage CAPTCHA. These tests can be seen in [16]. In the next subsections resistance to RBC in every CAPTCHA deciphering step will be analyzed.

5.1 Preprocessing

Background removal tools are mainly based on distinction between characters and the clutter, such as line, color, discontinuity, dot and mesh removal, and color segmentation. Line removal is not feasible because the characters are mainly composed of the same lines as the clutter, so by removing them, glyph information is also removed. Moreover, dot removal fails for the same reason line removal tool does. Color segmentation, as mentioned in subsection 3.5, does not apply to the proposed CAPTCHA, and the discontinuity removal is dependent on line removal. Also, converting CAPTCHA to binary colors removes a lot of information from the glyphs, making them unusable for recognition. In addition, textured mesh is random and not regular, so applying universal mesh removal tools also does not lead to deciphering improvement. The texture mesh comprises hexagons with randomly sized edges to eliminate the possibility of an attacking party duplicating it and filling in the missing glyph pieces. Moreover, the texture mesh can be generated each time a new CAPTCHA is generated.

5.2 Segmentation

Segmentation is the most important step in deciphering CAPTCHA, because this is the step in which the human outperforms the machine. Therefore, segmentation should be as hard to perform as possible. In the proposed method, if the attacker manages to separate glyph regions from the clutter, the resulting image does not have enough information about the glyphs to successfully implement common OCR. If the glyphs are extracted, however, they still pose a challenge for an OCR or the pixel count methods because of the meshed nature. This challenge would be easier for an attacker if the mesh would have been a simple continuous mesh, but in the proposed CAPTCHA this is not the case, as noted in the previous subsection.

5.3 3rd Party Attacks

The proposed novel CAPTCHA is highly resistant to 3rd party attacks, by the means of exploiting finite CAPTCHA states to create a database that the malicious user can exploit to spam, or to create thousands of fake accounts, polls and so on. The proposed CAPTCHA is highly random, from the background to glyph generation and placement, which also discourages the use of machine learning techniques, neural networks and similar AI attacks. Using letters instead of complete words also helped in improving the resistance from 3rd party attacks because of significant increase in CAPTCHA combination space. Moreover, this feature also helped to eliminate the possibility of dictionary attacks. However, if an attack method is used such as the authors in [8] tried to overcome, the proposed CAPTCHA could not resist. Nevertheless, an additional improvement of the proposed CAPTCHA can be done through interaction similar to [8], or just by using allowed time windows for solving CAPTCHA.

5.4 CAPTCHA Buffer

With the aim of using the proposed CAPTCHA in high traffic web applications, another novel idea is proposed. When issuing concerns about system slowdowns because of a lot of CAPTCHA generation demands, although the generation algorithm is not computationally intensive, a CAPTCHA buffer can be used. A CAPTCHA buffer is a storage which contains a predefined number of pre-generated CAPTCHAs. The concept is based on the idea of generating a certain number of CAPTCHAs in advance, when the system is lightly used, not only by demand, and summoning them as necessary. In this way, the sporadic system slowdowns can be avoided in peak usage periods. This mechanism can be used in conjunction with the clutter shape storage, described in subsection 3.2.

6 Future Work

The future tasks for the proposed method development are simplification of generation and readability improvement. Moreover, some fine tuning characteristics of the proposed method should be evaluated, such as the influence on RBC and RBH of glyph rotation, font face, character shaped mask warping, CAPTCHA size, etc. The authors plan to differ the number of characters more intensively, such as $4 \leq r \leq 7$. Also, another survey is planned that will encompass more people and with many different CAPTCHAs in order to gain better perspective on the practicality of the future proposed CAPTCHA.

Conclusions

CAPTCHA protection is considered by some authors as a weak protection against bots and spammers, but its ubiquitous use and the many researches still pending on it tell the different story. Its main characteristics are simple implementation, high practicality, good acceptance by people and fair security. In order to avoid the necessity of complicated security systems for simple tasks, such as mail registrations, auctions, polls, ballots, forum posts, etc. CAPTCHA is seen as the best balance of all the needed characteristics for these tasks. However, as CAPTCHA security advances, so too do the attack methods, such as various OCR and non-OCR based attacks, 3rd party solving techniques, AI approaches and many others. All these considerations need to be taken into account when creating new CAPTCHA security.

The proposed novel method, Advanced Character Collage CAPTCHA, proved itself superior to the most common types of attacks, along with its simple implementation and good readability. Its basic strengths lay in the strengths of unbroken CAPTCHAs, as well as in the flaws of the machine vision technologies, and AI imperfections. Despite the fact that the proposed CAPTCHA will be probably rendered unusable over time as technology advances, it is still a full of

challenges for an attacker and carries some novel ideas for possible future CAPTCHA implementations.

Acknowledgement

This work was supported by research project grant No. 165-0362980-2002 from the Ministry of Science, Education and Sports of the Republic of Croatia.

References

[1] L. von Ahn, M. Blum, N. J. Hopper, J. Langford: CAPTCHA: Using Hard AI Problems for Security, Proc. of 22nd Int. Conf. on Theory and Applications of Cryptographic Techniques, Warsaw, Poland, 4-8 May 2003, pp. 294-311

[2] R. Soni, D. Tiwari: Improved CAPTCHA Method, International Journal of Computer Applications, Vol. 1, No. 25, pp. 92-94, 2010

[3] T. Yamamoto, J. D. Tygar, M. Nishigaki: CAPTCHA Using Strangeness in Machine Translation, Proc. of 24th IEEE Int. Conf. on Advanced Information Networking and Applications, Hamamatsu, Japan, 20-23 April 2010, pp. 430-437

[4] M. Imsamai, S. Philmoltares: 3D CAPTCHA: A Next Generation of the CAPTCHA, Proc. of 2010 Int. Conf. on Information Science and Applications, Seoul, South Korea, 21-23 April 2010, pp. 1-8

[5] J. Cui, L. Wang, J. Mei, D. Zhang, X. Wang, Y. Peng, W. Zhang: CAPTCHA Design Based on Moving Object Recognition Problem, Proc. of 3rd Int. Conf. on Information Sciences and Interaction Sciences, Wuhan, China, 23-25 June 2010, pp. 158-162

[6] A. A. Chandavale, A. M. Sapkal: Algorithm for Secured Online Authentication Using CAPTCHA, Proc. of 3rd Int. Conf. on Emerging Trends in Engineering and Technology, Goa, India, 19-21 November 2010, pp. 292-297

[7] H. Gao, D. Yao, H. Liu, X. Liu, L. Wang: A Novel Image Based CAPTCHA Using Jigsaw Puzzle, Proc. of 13th IEEE Int. Conf. on Computational Science and Engineering, Xi'an, China, 11-13 December 2010, pp. 351-356

[8] H. D. Truong, C. F. Turner, C. C. Zou: iCAPTCHA: The Next Generation of CAPTCHA Designed to Defend Against 3rd Party Human Attacks, Proc. of 23rd IEEE Int. Conf. on Communications, Kyoto, Japan, 5-9 June 2011, pp. 1-6

[9] J. Yan, A. S. E. Ahmad: Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms, Proc. of 23rd IEEE Computer Security Applications Conference, Miami Beach, FL, USA, 10-14 December 2007, pp. 279-291

[10] A. A. Chandavale, A. M. Sapkal, R. M. Jalnekar: Algorithm to Break Visual CAPTCHA, Proc. of 2nd Int. Conf. on Emerging Trends in Engineering and Technology, Nagpur, India, 16-18 December 2009, pp. 258-262

[11] E. Bursztein, S. Bethard: Decaptcha: Breaking 75% of eBay Audio CAPTCHAs, Proc. of 3rd USENIX Workshop on Offensive Technologies, Montreal, Canada, 10 Aug 2009, pp. 1-7

[12] M. Shirali-Shahreza, S. Shirali-Shahreza: Advanced Collage CAPTCHA, Proc. of 5th In. Conf. on Information Technology: New Generations, Tehran, Iran, 7-9 Apr 2008, pp. 1234-1235

[13] P. Golle: Machine Learning Attacks against the Asirra CAPTCHA, Proc. 15th ACM Conf. on Computer and Communications Security, New York, NY, USA, 17 Oct 2008, pp. 535-542

[14] S. Hocevar, 2004 [Online] Available: http://caca.zoy.org/wiki/PWNtcha. [Accessed: March 2011]

[15] G. Martinovic, A. Attard, Z. Krpic: Proposing a New Type of CAPTCHA: Character Collage, Proc. of the 34th Int. Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, 23-25 May 2011, pp. 1447-1451

[16] G. Martinovic, Z. Krpic: Testing the Reliability of Character Collage CAPTCHA Protection, Proc. of the 27th International Kandó Conference – Science in Practice, Óbuda University, Budapest, 17-18 November 2011