ISSN: 1803-3814

BUBBLE CAPTCHA - A START OF THE NEW DIRECTION OF TEXT CAPTCHA SCHEME DEVELOPMENT

Ondrej Bostik, Karel Horak, Jan Klecka, Daniel Davidek

Brno University of Technology
Department of Control and Instrumentation
Technicka 12, 61600 Brno
Czech Republic
bostik@feec.vutbr.cz

Abstract: *CAPTCHA, A Completely Automated Public Turing test to tell Computers and Humans Apart, is well-known system widely used in all sorts of internet services around the world designated to secure the web from an automatic malicious activity. For almost two decades almost every system utilize a simple approach to this problem containing a transcription of distorted letters from image to a text field. The ground idea is to use imperfection of Optical Character Recognition algorithms against the computers. The development of Optical Character recognition algorithms leads only to state, where the CAPTCHA schemes become more complex and human users have a great difficulty with the transcription.*

This paper aims to present a new way of development of CAPTCHA schemes based more a human perception. The goal of this work is to implement new Captcha scheme and assess human capability to read unusual fonts newer seen before.

Keywords: *CAPTCHA, reCaptcha, AniCap, HelloCaptcha, Bubble Captcha*

1 Introduction

With the development of web services the situations occurred, when people started to use automated systems to interact with web services. Their intention was clear: influence various online pools, add an unwanted and obtrusive post to public discussion forums, execute multiple search queries to online databases and so on. The need for a new technique which differentiates human user from the automated system was imminent.

CAPTCHA (Completely Automated Public Turing Test to tell Computers and Humans Apart) is defined as a general task that must be at the same time very easy for the human brain, but enormously difficult for a computer program both for the computing resources and for the algorithm complexity. The probability of successful passing the test by a human could be near 100%, but the computer algorithm should pass in only very small amount of attempt, closing the 0% success rate.

Although is the Captcha[1] titled as "Turing test", the correct designation should be "Reverse Turing test," because machines are in the role of arbiters, not the humans.

The goal of this work is to implement the new type of text-based Captcha scheme based on a generation of pseudo-random font consist of bubbles. The next part is to carry out a test on wide enough selection of population to determine the suitability of the Captcha scheme for common usage.

2 Current Captcha methods overview

Common practice to defending the web pages from attackers is to populate some kind of Captcha scheme based on the transcription of distorted characters from the image. This section summarizes state of the art solutions in text-based Captcha schemes.

2.1 Static text-based Captcha scheme

The majority of today's popular web portals utilize very simple Captcha schemes to limit access by automated programs. Every Captcha challenge consists of a purposely degraded image of several alphanumerical characters. The purpose of this type of Captcha scheme is to transcribe indistinct characters into text form alongside with inputted data. The initial idea was based on imperfections of Optical Character Recognition (OCR) algorithms at that time.

[1] Acronym will be written in lowercase to improve readability of the text

Development progress in current computer vision algorithms and primary the improvement in OCR algorithms lead to the state when it is a rather easy task to overcome static text-based Captcha with automated computer attacks. New task based on this principle should keep some recommendation, which could be divided into two main areas.

Every static text-based Captcha scheme must be primarily secured with some kind of anti-segmentation technique. If the scheme can be segmented to the individual characters, in is usually not a problem to recognize the characters successfully [5].

The current state of the art Captcha schemes relies on tilting the characters, waving the text across the image etc. They also rely on adding more lines, but which must be sufficiently wide and inappropriate angles so that they can not be distinguished by Hough's transformation. An important element is also to use different keyword lengths, so the program is unable to explicitly identify the search regions where individual characters should be located [5].

Using security measures like the complex background is not recommended. To ensure readability for human users, it is necessary to unify the colors of the text or to use a different method to make the characters more readable. Based on the findings of the currently applied principle, these codes can be easily broken down and used against the Captcha scheme [5].

The second stage is secure Captcha scheme at the level of individual characters. It is recommended to use characters of different sizes and different rotations. It is also important to use different character fonts. On the contrary, it is not recommended to use a large set of characters, because for example problems with the distinction of zero and letter O is common to the computer and human user. The use of random noise is also not recommended because the current algorithms handle the noise better as the human brain [5].

2.2 Drag and Drop text-based Captcha scheme

The perspective method introduced in [7] based on the principle of dragging objects to the right place. The challenge is user-friendly, intuitive and fast. It can be an interesting challenge for the machine as it must recognize not only the input image but also decode the characters in the response stack. And most importantly - they have to drag them to the correct place, which is a program-intensive operation. The applet can also analyze the movement of the mouse and other objects on the page and draw further conclusions.

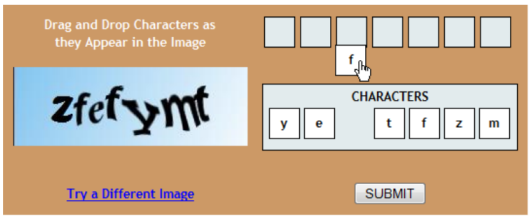


Figure 1: Drag and Drop Captcha (taken from [7])

Users have a great advantage in dealing with situations with uncertain characters, where "l" can not be distinguished from "1", "0" from "O", etc. because of the human instincts. On the other hand for users, the test take extra time, because instead of writing a response in a text box, characters need to be moved one by one [7].

2.3 Animated text-based Captcha scheme

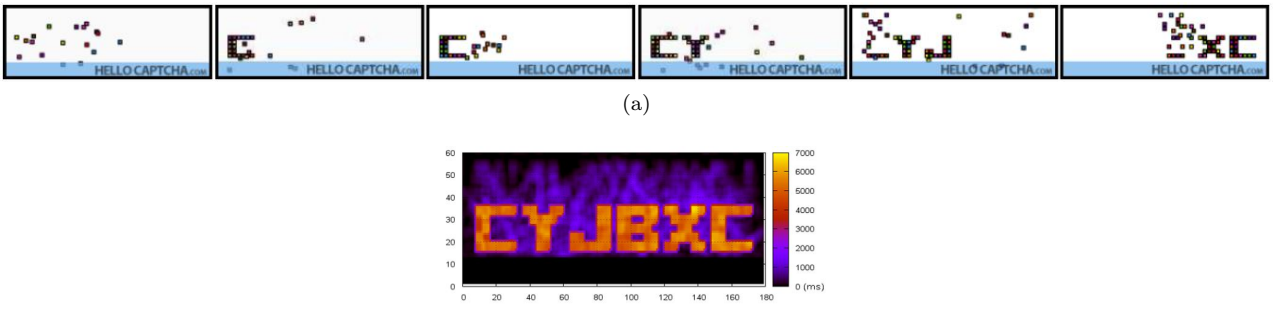
This is the type of challenge where the Captcha image or part of it moves and in this way makes it difficult to read the code automatically. Hello Captcha system[2] was chosen as a representative. It is a set of 84 different kinds of animated tests. All of them are based on animation such as flying letters, hiding or uncovering parts of the image, and many others.

A thorough safety analysis was conducted in the provided study[8]. As a result, it was found that despite the apparent complexity such schemes could also be broken. First, it is necessary to find out which variant from the set of 84 types the currently solved image is and then apply the appropriate strategy to determine its solution.

The type of challenge can be determined based on the number of frames, the different transition times between each slide, the number of background images, the predominant color of the images etc. Based on these factors the type of challenge can be determined with the accuracy of 80% to 100%.

Many Captcha code solutions are spread over several frames, as can be seen on figure 2a. However, the often solution is to fold the so-called "Pixel Delay Map" (figure 2b) from the whole sequence (or part of the

sequence). Pixel Delay Map is created by summing the brightness values at the corresponding position over the entire selected time. The resulting image is easy to threshold and can be used to determine the exact solution using standard OCR algorithms.

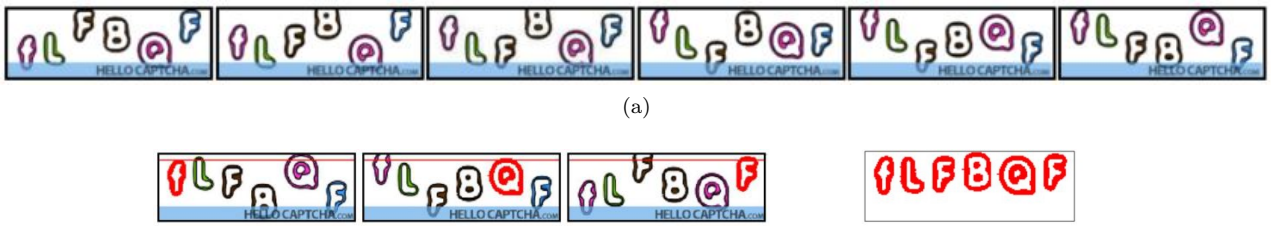


(a)

(b)

Figure 2: HelloCaptcha - animated Captcha of Flitter/Colorful type. (a) Several sample frames taken from animation (b) Pixel Delay Map (taken from [8])

For other schemes such as the one on the figure 3a based on letter movement in the vertical direction, an imaginary line can be created in the image (see fig. 3b). When any part of the image overlaps the line, it will trigger action, which stores the connected moving region which crosses the line. Once the animation cycle is complete, all these objects can be stacked in one image (see figure 3c) and submitted to OCR algorithms.



(a)

(b) (c)

Figure 3: HelloCaptcha - animated Captcha of Spring/Jumpers type (a) Several sample frames taken from animation (b) Selected frames featuring character crossing segmentation line (c) Composite image prepared for OCR (taken from [8])

For other diagrams, we can always select a defined region of the image that is always at the same coordinates. Then we could compose a complete image from sub-frames taken from exactly defined frames.

An automated system built to verify results has been able to find solutions with the success of 16 to 100% depending on the type of challenge, and therefore this system can be considered as cracked [8].

2.4 Motion text-based Captcha scheme

AniCap (see figure 4) as a representative of the 3D Captcha challenges is based on the human perception of space. The generated test is for the human eye ostensibly composed of several layers of letters. Thanks to animation based on motion parallax, most of the humans are able to distinguish the position of the letters in the space and select 4 letters in the foreground. The schema can not be broken using the usual procedures based on color analysis, edge search, differential image. Reconstruction of 3D space will be very difficult in this case because all individual images contain a great rate of distortion and it is not easy to search for the correspondences [6].

2.5 Static text-based reCaptcha

The reCaptcha system was originally designed by Professor Luis von Ahn and his colleagues in the period from 2007 to 2009. The system was subsequently sold to Google, which continues with managing and development.

The original intention of the system was to use human work associated with transcription of computer-generated images to something more productive than just anti-spam protection. According to [3], in 2008, more than 100 million Captcha codes were resolved every day. Assuming an average time of 10 seconds per Captcha image, humanity has spent more than 30 years of time with code transcription every day.

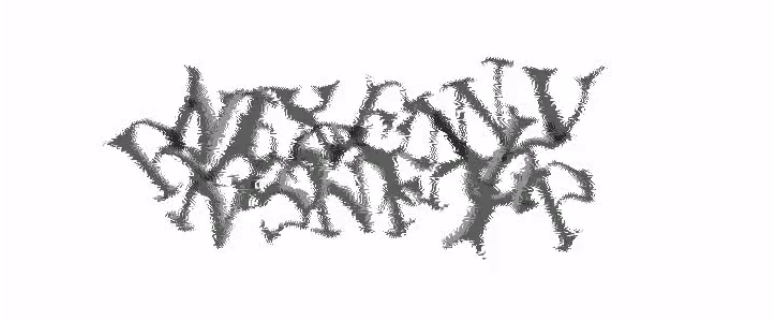


Figure 4: Sample of AniCap Captcha scheme - answer SYAK (taken from [6])

The reCaptcha system aims to take advantage of this mankind's mental effort. Originally, the system was used to help OCR algorithms to transcribe texts from the pre-digital era and convert this data to more readable and searchable form. Now, this system also enhances map data by recognizing building numbers, image contextual analysis, and other innovative Google systems.

The principle of the reCaptcha system is simple. Scanned documents are converted into digital text form by two independent OCR algorithms. The outputs are compared between themselves and the English dictionary. If the OCR outputs differ or the word is not found in the English dictionary, the word is marked as "suspicious" and stored for further processing. These words are submitted to the user of the reCaptcha system (see figure 5) for further attempts to recognize [3].

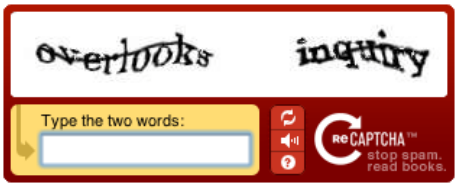


Figure 5: Sample of Google reCaptcha text scheme, (taken from [1])

In order to exclude the human transcription error, every result to transcribe the word is recorded from both people and OCR algorithms. The transcripts are then voted among themselves, the human input counts for one voice, the OCR algorithm weight is half. To validate the transcript, the weight of the transcription must reach at least a $2,5$ voice. Words that passed this algorithm are then added to the control set [3].

To distinguish a human visitor from the machine, each picture is composed of two words. For one word, the correct transcription of the world has been determining correctly. This word serves as control word for Captcha purposes. The second word is selected from the set of "suspicious" words and the aim of transcription is to find the right solution. However, the visitor does not know which word is controlling and which one is currently under the trials [3].

The authors of the original project state that this system is capable of converting text into digital form with the success rate of 99.1%, which is comparable to the method used in the transcription of texts by two human professionals. Standard OCR programs work with the success rate of 83.5% [3].

The main advantages of the system include the use of English words instead of random clusters of letters and numbers, which will significantly speed up transcription. Also, the system utilized human labor that would otherwise be totally unused. For security reasons, it can be highlighted that the reCaptcha system uses of words from a scanned document that previously did not be successfully recognized by classical OCR algorithms, which further enhances schema security.

2.6 Static image-based reCaptcha

Proceded development of the reCaptcha system resulted in the creation of a simpler version of the system (see figure 6). Instead of transcribing the text from the image, the goal of image-based reCaptcha is finding images with similar content as a reference image. This also improves Google's service for extracting semantic information from a general image. This version has found its main application primarily on mobile platforms due to its simplicity and touch screen friendly usage.

In 2016, the article [10] was published, containing a successful method for breaking the image version of the reCaptcha system. The proposed system is based on the acquisition of semantic information from the image.

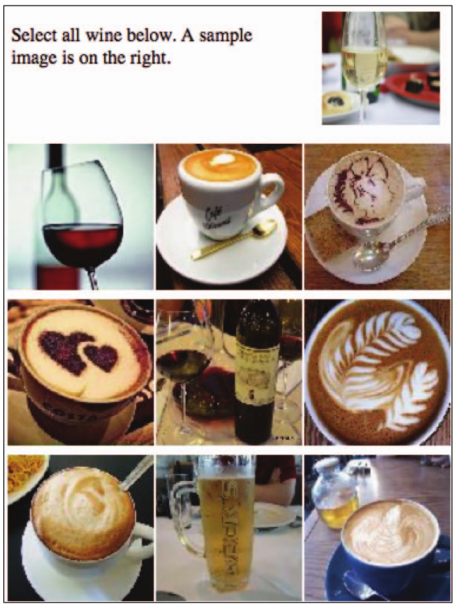


Figure 6: Sample of Google reCaptcha image scheme, (taken from [10])

One of the methods used is Google Revers Image Search (GRIS) to search for similar images based on an input image. It also utilizes online annotation services [10].

The big weakness of the system turned out to be a small gallery of used images. Manually created database of annotated images obtained from previously submitted Captcha codes has greatly increased the success of the system.

Another important finding was the fact that 74% of cases contain two correct answers; in the rest of the calls, you need to select three answers. Based on these findings, a designed system was able to attack more than 2200 frames with the success of 70, 78% [10].

2.7 No Captcha reCaptcha

This is a system that evaluates the behavior of a person on the web. The test of the suitable level of difficulty will be presented to the user based on users previous behavior. Ideally, if the system is almost certain that the user is a human, the user is prompted to check the "I'm not a robot" check box, as in 7. With a decreasing level of assurance, an image-based or text-based type of reCaptcha is provided to the user. The goal of the system is to ensure maximum user comfort.

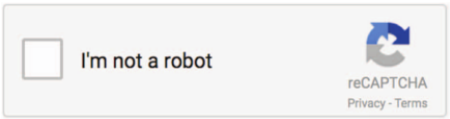


Figure 7: Sample of Google No Captcha reCaptcha scheme, (taken from [10])

Even during the background behavior analyses of a user, the cited research system [10] is foolish enough to make up to eight calls of this simple type per day. The attacker need only to create a virtual machine and use a web browser (such as Selenium) to mimic the human user's behavior (browsing the web, performing random searches and viewing their results, opening videos on youtube.com, etc.). The result of this effort is Google Tracking Cookie, which is a key element of Google's behavior analysis.

This article describes several key factors that lead to a decision on the complexity of the reCaptcha presented to the user. Although this version of the reCaptcha system is considered to be the best in the industry, it has been proven that it is possible to successfully break through [10].

3 Bubble Captcha implementation

Development of Bubble Captcha scheme starts as an idea after reading a research [5]. One of the key elements for strengthening the Captcha scheme is to use multiple fonts. To improve this recommendation, our idea was

to generate the font randomly.

Our next inspiration was well-known Rorschach test [9], the psychological test based on human perception of inkblots. Creation Captcha scheme, which utilizes some kind of Rorschach images or abstract art, could be an ultimate goal of Captcha development.

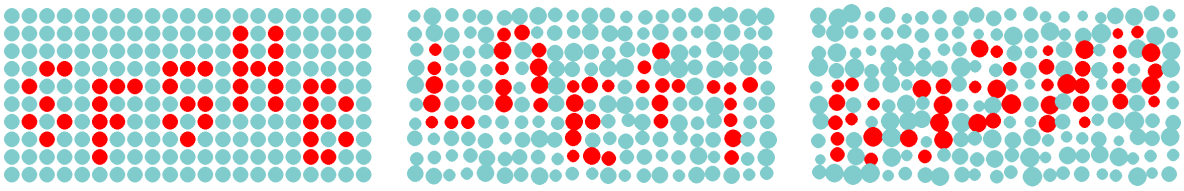
To demonstrate the way, the development could go, we design a simple Captcha scheme based on the distribution of bi-color ink stain or ink bubbles distributed in a firm grid through the Captcha scheme image.

For this purposes, we prepare a framework for testing various parameters of proposed system written as a PHP class for quick deployment of Captcha schemes on the web. The main input parameter of the class is the vector of the two-dimension binary array representing positions of red and green balls in the grid. During the generation of particular Captcha image, we randomly select 5 to 7 characters from used character set. The bubble representation of each letter is then placed into the resulting image. The empty positions in the grid are the filled in by green balls.

Other parameters of the PHP class are for example number of characters per image and the percentual rate of randomness in the picture.

Until this moment, the image is held in memory in form of position coordinates for each ball. The public method called *getImage()* can be used to generate the PNG picture via PHP commands. This method converts virtual bubbles stored in the array into the resulting image is a process of random selection of balls from memory. Each one is then placed in random coordinates around the intended position. The diameter of all the balls is also random.

Examples of proposed Captcha scheme is shown on figure 8. For uniformity and better comparison between variants, we select only images contains five letters. All source code can be downloaded from Bitbucket [4].



(a) (b) (c)

Figure 8: Buble Captcha variants overview (a) with no displacement - answer 6F5HB (b) with medium displacement rate - answer LNE4T (c) with big displacement rate - RJ3HN

We present three variants of Bubble Captcha. Differences between the three variants are the rate of randomness in bubble diameters and in bubble placements. The simplest variant (a) presented on figure 8a contains every bubble in precise centers of the grid with uniform diameters for each bubble.

Variant (b) and (c) presented on figure 8b and 8c respectively contains various rate of randomness in positioning and bubble diameters. All the parameters of all three variants are presented in table 1.

Table 1: Buble Captcha parameters for each variant

Variant [−]	Buble diameter [px]	Buble variation [%]	Grid dimension [px]	Grid variation [%]	Number of letters [−]
a	30	0 %	35	0 %	5 - 7
b	30	10 %	35	20 %	5 - 7
c	30	15 %	35	30 %	5 - 7

The greater rate of randomness resulted in rapid decreasing of human readability even for trained personal. For this reason, we do not implement more scheme variants with a greater rate of randomness.

4 Human subject experiments

Resulting PHP class presented in the previous section was utilized in experimental web portal hosted on our server. The web page was designed to show random Bubble Captcha image of selected variant. Users can provide a name and his transcription of shown Bubble Captcha challenge. Users can freely select the level of difficulty, but the accompanying text asks every participant to try each of the difficulty levels. Every entry is then stored in MySQL database alongside the right answer and provided name. The collected data is then used to show the overall statistics on the page.

A heterogeneous group of 24 users in total was selected from population. The test set contains booth man and woman, the age of participants was uniformly distributed between 18 and 50 years. Subjects have involved at different times of the day and come from the various background.

The users completed 645 Buble Captcha challenges total with a success rate of 68%. The results of the experiment for each of the variants are presented in table 2.

Table 2: Total statistics per difficulty level

Level of difficulty	Total attempts count	Succes attempts count	Failed attempts count	Success rate
Variant a	191	138	53	72%
Variant b	295	200	95	68%
Variant c	159	103	56	65%
Sum	645	441	204	68%

As we expected, the success rate decreased with increasing level of difficulty of challenge, but the difference is only 7%, which is not significant.

The next experiment aimed to determine the error rate at the level of individual character. The results are presented on figure 9.

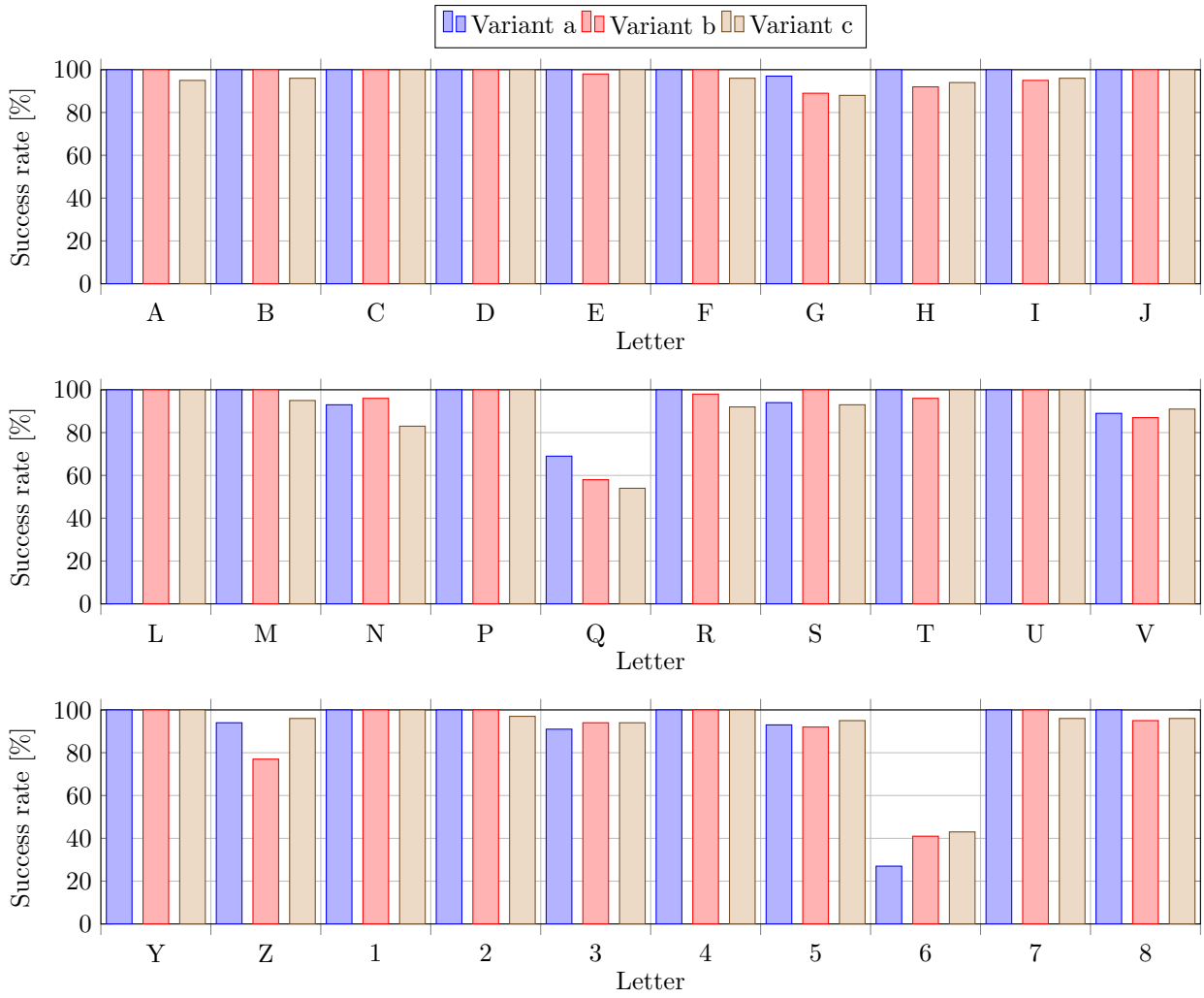


Figure 9: Success rate of all used characters for all three types of Bubble Captcha

The statistics represent fact, that almost all the characters used in Bubble Captcha scheme are readable for human subjects participating in the experiment. We recorded some problems with several of the characters.

The worst statistics during the testing gain the number 6. Only 37% of displayed character six was transcribed correctly. At the same time, users reported difficulty with this character, because they intuitively

cannot distinguish number six with number 5 and letter S. Several participants pointed to fact, that used font for number six is inappropriate. As displayed on figure 8a, numbers 5 and 6 are very similar for untrained eye.

In the same spirit, the second worst character, the letter Q, was considered to be letter O or number 0. Interestingly, we do not include letter O or number 0 in test set due to their similarity. It is a good practice not to include this character in any generic Captcha scheme. Human mind at normal circumstances differentiate between this characters based on context and ordinary Captcha scheme does not provide the context for the human mind.

The next three most problematic characters are the letter V commonly confused with the letter U, then the letter Z commonly confused with number 2 and letter N because of a strange font used for this letter.

5 Conclusion

This paper illustrates a new direction of Captcha scheme development. During the first experiment on a human subject, the success rate was 68%, which could look bad. But in the deeper exploration, we discovered, that this success rate was dragged down by the inappropriately chosen representation of some letters. We want to improve the current font and also prepare more variant for each letter. To further improve the success rate of human users, we will narrow the character set striping down the hardest letters.

We are aware, that this particular scheme can be successfully attacked by numerous algorithms like neural networks or k-NN classification algorithms. The goal of this paper is to assess the usability of the idea and Buble Captcha will be improved to harder the task of computer cracking of this scheme.

The next step in development will be implementing some advanced anti-segmentation techniques like using more colors, which will be selected dynamically to increase the difficulty of Bubble Captcha scheme. The next idea is to generate more object and stack them layer on layer to achieve more complex structure.

Acknowledgement: The completion of this paper was made possible by the grant No. FEKT-S-17-4234 - "Industry 4.0 in automation and cybernetics" financially supported by the Internal science fund of Brno University of Technology.

References

[1] The Official CAPTCHA Site (2010). URL http://www.captcha.net/

[2] CAPTCHA : HelloCaptcha.com (2016). URL http://www.hellocaptcha.com/

[3] von Ahn, L., Maurer, B., Mcmillen, C., Abraham, D., Blum, M.: reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science (80-.). **321**(12 September 2008), 1465–1468 (2008). DOI 10.1126/science.1160379. URL http://www.ncbi.nlm.nih.gov/pubmed/18703711

[4] Bostik, O.: Bubble Captcha Source Code (2017). URL https://bitbucket.org/ondrejbostik/bubble-captcha

[5] Bursztein, E., Martin, M., Mitchell, J.C.: Text-based CAPTCHA strengths and weaknesses. Proc. 18th ACM Conf. Comput. Commun. Secur. **2011**, 125–138 (2011). DOI 10.1145/2046707.2046724. URL http://doi.acm.org/10.1145/2046707.2046724http://cdn.ly.tl/publications/text-based-captcha-strengths-and-weaknesses.pdf%5Cnhttp://dl.acm.org/citation.cfm?id=2046724

[6] Chow, Y.W., Susilo, W.: AniCAP: An Animated 3D CAPTCHA Scheme Based on Motion Parallax, pp. 255–271. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). DOI 10.1007/978-3-642-25513-7_18. URL http://link.springer.com/10.1007/978-3-642-25513-7_18

[7] Desai, A., Patadia, P.: Drag and Drop: A Better Approach to CAPTCHA. In: 2009 Annu. IEEE India Conf., pp. 1–4 (2009). DOI 10.1109/INDCON.2009.5409359

[8] Nguyen, V.D., Chow, Y.W., Susilo, W.: Breaking an Animated CAPTCHA Scheme. In: J.Z. Feng Bao, Pierangela Samarati (ed.) Appl. Cryptogr. Netw. Secur., pp. 12–29. Springer, Berlin, Heidelberg, Singapore (2012). DOI 10.1007/978-3-642-31284-7_2. URL http://link.springer.com/10.1007/978-3-642-31284-7_2

[9] Rorschach, H.: Psychodiagnostics: A Diagnostic Test Based on Perception, 10th edn. Hogrefe Huber Pub (1998)

[10] Sivakorn, S., Polakis, I., Keromytis, A.D.: I am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs. In: 2016 IEEE Eur. Symp. Secur. Priv. (EuroS P), pp. 388–403 (2016). DOI 10.1109/EuroSP.2016.37