A Novel Animated CAPTCHA Technique based on Persistence of Vision

Shafiya Afzal Sheikh, M. Tariq Banday
Department of Electronics and Inst. Technology
University of Kashmir
Srinagar, India

***Abstract*—Image-based CAPTCHA challenges have been successfully used to distinguish between humans and bots for a long time. However, image-based CAPTCHA techniques are constantly broken by hackers, forcing web developers to implement more robust security features and new approaches in CAPTCHA images. Modern-day bots can use many techniques and technologies to break CAPTCHA images automatically. These techniques include OCR, Segmentation, erosion, threshold, flood fill, etc. This led to innovative CAPTCHA systems, including those based on drag and drop, image recognition, fingerprint, mathematical problems, etc. Animated image CAPTCHAs have also been designed to show moving characters and objects and require users to recognize the characters or objects in the animation. Unfortunately, these CAPTCHA systems have also been broken successfully. This research proposes a novel animated CAPTCHA technique based on the persistence of vision, which shows text characters in multiple layers in an animated image. The proposed CAPTCHA technique has been implemented in PHP using GD library functions and tested using various popular CAPTCHA breaking tools. Further, the proposed CAPTCHA challenge has also been tested against the frame separation based breaking technique. The security analysis and usability study have demonstrated user-friendliness, vast accessibility, and robustness.**

Keywords*—*CAPTCHA; OCR; animation; segmentation; botnet; HIP; CAPTCHA usability

I. Introduction

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is some kind of test or challenge which can be solved by a human user very quickly but cannot be solved by modern computer software [1]. These tests help distinguish between humans and computer programs. Unfortunately, hackers widely use automatic computer programs or bots to misuse Internet-based services causing harm to the services and service providers. Therefore, it is incumbent that these services be prevented from automated access and misuse by bots, and it should be done without affecting human users. CAPTCHA challenges help the Internet-based services distinguish between humans and bots, and based on the CAPTCHA test result, and they deny access to the bots. Nowadays, CAPTCHA tests are extensively used on the Internet and have effectively kept away automated bots and prevented misuse of online services only for human users [2].

CAPTCHAs are used for securing web-based services in many ways. They prevent bots from creating email accounts that can misuse online email services and send SPAM emails. CAPTCHAs are also used to prevent search engines and crawlers from accessing web pages and accessing or copying any type of content. Spammers use web crawlers to automatically fetch website content and harvest email addresses from website content. CAPTCHAs can help hide email addresses by default and reveal them only to human users. CAPTCHAs help prevent hacking attacks by defending against brute force and dictionary attacks. These attacks work by trying many login attempts at a fast rate. CAPTCHAs help determine if the client is a human, and only then the service allows further attempts to log in or access the prevented resource. Online services are intended to enable human users to keep trying logging in, even after a bunch of failed attempts, rather than disallow further login attempts, as in the case of bots. Asking a human user to solve a CAPTCHA instead of blocking them offers users a better user experience than being blocked from further login attempts. CAPTCHAs prevent bots from accessing and spamming discussion forums, comment sections of websites, online polling systems and social media applications. Gaming bots can be highly competitive against human users in playing computer games and thus need to be kept away from online gaming platforms. CAPTCHAs help e-commerce websites reject bots that obtain product information and pricing for price comparison. CAPTCHAs play an essential role in keeping these bots away from online services.

Hackers constantly keep trying to bypass CAPTCHAs by implementing CAPTCHA breaking techniques in their bots, making it necessary to design more secure CAPTCHA challenges. There are several types of CAPTCHA challenges currently on the Internet-based services viz. Text-based CAPTCHA, Image-based CAPTCHAs, Audio-based CAPTCHA, Video-based CAPTCHAs, Puzzle-based CAPTCHAs, Mouse based CAPTCHAs and Invisible CAPTCHA.

The most common type of CAPTCHA is Text-based, in which a set of characters is displayed on an image, and the user is required to recognize the text characters and type them in a text box. If the user input matches the characters displayed on the image, the CAPTCHA is accepted as passed. Some examples of Text-based CAPTCHAs are EZ gimpy, Gimpy, Register, Ticketmaster, Yahoo and its multiple Versions, Mailblocks, Google, MSN, Holiday inn priority CAPTCHA, Phpcaptcha.org, FreeCap, Megaupload, BotDetect, Cryptograph, LinkedIn, Authorize, Baidu, Blizzard, CAPTCHA.net, CNN, Digg, Megaupload, Slashdot,

Wikipedia, Hollowstyle, Tencent, Sina, CmPay, MicrosoftCAPTCHA, Baffle Text, TeaBag CAPTCHA 1.2, 3D CAPTCHA, Handwritten CAPTCHA, Synthetic handwritten CAPTCHA, MSN CAPTCHA, 3D CAPTCHA, STE3D-CAP, Sigma-Lognormal CAPTCHA, DevaCAPTCHA, Google CAPTCHA etc.

In Image-based CAPTCHAs, one or more images are displayed to the user, and they are asked to recognize objects on the images. This CAPTCHA type is very effective because computer programs are not smart enough to process and recognize non-text objects from an image. Examples of image-based CAPTCHA are BONGO, Anomalies Image CAPTCHA, Assira CAPTCHA, PIX CAPTCHA, Implicit CAPTCHA, Google Image CAPTCHA, Drawing CAPTCHA, Facebook CAPTCHA, Image Block Exchange, Face Recognition, Multilingual, KitttenAuth, MosaHIP, Image Flip CAPTCHA [3] etc.

Audio-based CAPTCHA is another CAPTCHA type, but they are not as common as the other CAPTCHAs. They allow users to play a sound and recognize words spoken in the audio. The audio usually has background noise to prevent it against voice recognition based breaking attacks. This CAPTCHA type helps blind or visually impaired computer users pass CAPTCHA tests. Examples of audio-based CAPTCHA are CAPTCHA for blind users, HIPUU, Google reCAPTCHA, Digg etc.

Video-based CAPTCHAs are yet another type of CAPTCHAs that are rarely used. They require a user to watch a short clip and then answer a question based on the information provided in the video. E.g., recognizing a human gesture, moving objects or text from the video. Video-based CAPTCHAs require more internet bandwidth and use attentiveness and time. Some examples of video CAPTCHA are 3D animation CAPTCHA, AniCAP, Motion CAPTCHA, New video CAPTCHA, NuCAPTCHA, HelloCAPTCHA, DotCHA, etc.

Another common type of CAPTCHA scheme is Puzzle-based, in which the user to require to solve a small, easy puzzle that a computer program cannot solve. It depends solely on human intelligence because computer programs are nowhere near good at solving random puzzles. A few examples of puzzle CAPTCHA are 3D animation CAPTCHA, AniCAP, Motion CAPTCHA, New video CAPTCHA, NuCAPTCHA, HelloCAPTCHA, DotCHA, etc.

Mouse based CAPTCHAs are effective and very easy to use. They require users to click a button or a checkbox to declare that they're not bots. The CAPTCHA system records the users' mouse movement patterns and analyses those patterns to determine whether or not the user is a bot. The best example of this type of CAPTCHA is Google reCAPTCHA v2. Other examples are Mouse CAPTCHA, unCAPTCHA, Drag and Touch CAPTCHA [4] etc.

Invisible CAPTCHAs are gaining popularity lately. They do not require users to do anything, making them the most user-friendly way to distinguish between humans and computer programs. They work by analyzing users' previous actions like recent website activity, session information and other parameters like browser or client information, IP address reputation etc. The best example of invisible CAPTCHA is Google reCAPTCHA v3 [5].

CAPTCHAs are not immune to attacks. Hackers constantly keep trying to break or bypass CAPTCHA systems to perform their activities efficiently. They make use of advanced techniques to find ways to break the CAPTCHAs. CAPTCHAs in all of the types mentioned earlier have been successfully broken.

Simple Text-based CAPTCHA challenges are very easy for a bot to break with the advent of image segmentation and OCR technology. To prevent text-based CAPTCHAs from being broken using OCR and related technologies, the text characters on the image are distorted and deformed in different ways to make automatic text recognition difficult while still keeping them recognizable by human users [6], [7].

Image-based CAPTCHA challenges have also been broken successfully using feature extraction (colour and texture), SVM classification techniques, face detection using kNN classification techniques, google reverse classification, HSV model, image collection, tag classification techniques. Audio-based CAPTCHAs have been broken using vertical segmentation, DFT recognition, Ada Boot, SVM, CNN techniques.

Video-based/Animation CAPTCHAs have also been broken using frame selection, pixel display timing, vertical segmentation, connected pixels, flood fill, k-means clustering techniques using SIFT and NN classifications.

Mouse-based CAPTCHA has been successfully broken using fake click implementation techniques, image annotation services and tag classifiers. In addition, invisible CAPTCHAs have also been broken using Reinforcement Learning techniques with a high success rate [5], [8], [9].

A. Contribution

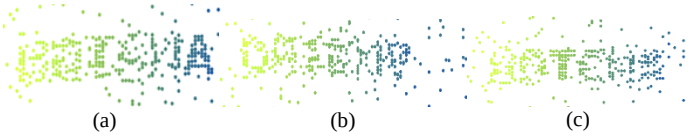
In this research, a CAPTCHA has been designed, which works on the concept of persistence of vision or retinal persistence. On the retina of an eye, the visual perception of an object does not end immediately and remains for a fraction of a second even after the light coming from it stops entering the eye. This research proposes to use this phenomenon of the human eye to display a set of images to the users at a fast frame rate, each frame showing partially visible alphanumeric characters, giving the user an illusion of completely visible numbers or characters. Furthermore, the proposed CAPTCHA displays two sets of partially visible characters, one after another, for a short duration. Individually, none of the images shows any of the characters thoroughly, making it difficult to extract the character information from individual frames of the animated CAPTCHA. Therefore, the CAPTCHA is very secure against frame separation, segmentation and OCR based CAPTCHA breaking techniques.

II. Literature Review

AniCap is an animated 3D text-based CAPTCHA challenge based on the concept of motion parallax. Each 3D character in AniCap has a random 3D transformation that rotations in all three dimensions. Unlike other approaches that add random

clutter to the CAPTCHA challenge to deter automated attacks, it uses overlapping text-on-text with no distinct colours or borders around the character. The two layers of text move around at different paces that give the user a feeling that the two layers of text are to varying distances from the users' perspective. This allows the user to recognize the text in front and back layers separately. AniCap is difficult to break using common CAPTCHA breaking techniques because the foreground and background colours are not distinct, and therefore edge detection is not possible easily [10].

DotCHA is a new and unique 3D animated CAPTCHA that displays text and uses human interaction to avoid the shortcomings of existing 2D and 3D CAPTCHAs. DotCHA requires users to use a mouse or finger gestures on a mobile device to rotate a random-looking extensive collection of small balls, constantly moving in a 3D circular direction, a 3D text model, to identify the correct letters. The 3D model formed by the collection of moving balls is a twisted form of 3D letters around a centre pivot axis, and it displays different letters at different angles of rotation. This is because the balls line up to form a shape of letters only at specific angles of rotation. Because the characters in DotCHA are made up of many small balls instead of solid colour text characters, it belongs to the scatter-type CAPTCHA category. Therefore, it can't be broken using segmentation techniques. DotCHA is also safe against machine learning-based attacks because the characters are recognized only at a specific degree of rotation [11]. However, our analysis of DotCHA was very user-unfriendly because it is tough to recognize the characters as they are not distinctly distinguishable due to a lack of solid colour and distinct borders or edges. Fig 1. shows a few screenshots of DotCHA.



(a) (b) (c)

Fig. 1. DotCHA Sample Images. (a) in which the Character 'A' is Visible. (b) in which the First Character is Supposed to be 'D' (c) the Letter 'T' is Barely Recognizable.

HelloCAPTCHA is an animated CAPTCHA web service that generates CAPTCHA animations on the HelloCAPTCHA server. It generates 2D animations which display animated text characters. The service generates at least 84 known different variants of animations. HelloCAPTCHA provides programming interfaces for PHP and JAVA web applications and plugins for Joomla, Drupal, WordPress. (hellocaptcha.com). HelloCaptcha Developers claim that their animated CAPTCHA schemes are user-friendly, secure against breaking attacks, especially because of the extra time required to break the CAPTCHAs over multiple frames instead of a single image in traditional static CAPTCHAs [12]. [13], [14] have successfully broken HelloCAPTCHA using various techniques that include Pixel Delay MAP (PDM), Calculating Line, Color Selection, Frame Selection etc. They report a success rate between 16% and 100%.

Motion CAPTCHA is a video-based CAPTCHA scheme in which a short video clip is displayed to the user. The video shows a human performing some action or gesture. The user must watch the clip, recognize the action or gesture performed in the video, and choose the correct description of the action from a list of actions provided to the user alongside the video. This CAPTCHA scheme requires users to be good at understanding the English language and takes time to watch the video. The video may also take time to download in case of less internet bandwidth. [15] highlights the weaknesses of the MotionCAPTCHA.

Gesture-based animated CAPTCHA is a unique type of animated CAPTCHA in which sign language is used to convey numbers utilizing a video clip. For example, in the video clip shown in Fig.2., hands are rendered, which offer a few numbers in sign language. This type of CAPTCHA is easy for users familiar with sign language, but it is initially difficult for those who don't know sign language, which is the majority of the people in the world. Furthermore, this CAPTCHA type can't be broken using conventional CAPTCHA breaking methods and tools. Instead, it will require an AI-based system to recognize the hand gestures and break such CAPTCHA challenges [16].

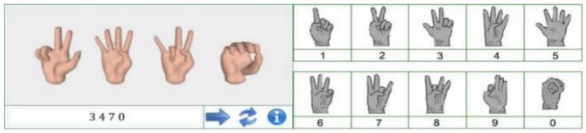


Fig. 2. Shows some Sample Hand Gestures Rendered using this Technique and Instructions for Understanding some Numbers in the Sign Language.

III. Proposed Solution

This research proposes creating an animated CAPTCHA challenge designed to mitigate the security issues that make animated CAPTCHAs insecure against breaking techniques. The proposed animated CAPTCHA is based on image retention by the retina of the eye for a short duration after the image is not visible anymore. Because the human is able to process the data in such a way that it has the ability to create meaning from a sequence of meaningless partially visible pieces of an object, seen in a sequence for a very brief period of time, it is possible to use this special ability to distinguish a human from a machine which doesn't have such an ability. This special ability of the Human eye can be used to create an illusion of fully visible text characters, even though they are shown piece by piece, sequentially at a fast speed.

The CAPTCHA system generates a set of images with random background colours, lines and patterns. A transparent image is then created on which a group of randomly generated alphanumeric characters is printed. The number of characters is chosen randomly for the two sets of frames. The alphanumeric characters are printed in a random colour and font size between 12 and 16. The text is printed starting at the random top and left pixel locations and is also rotated randomly between +30 and -30 degrees before being printed on the image.

Then, sequentially, parts of all the characters are erased from the image to create a frame with partially visible characters. This process is repeated several times to create a set of frames, and in every frame, different parts of the characters are trimmed off. These transparent frames are then pasted on the previously generated background images. Creating the

frames is repeated with a different set of alphanumeric characters to create another set of frames. Hence, two sets of frames are constructed with two different sets of alphanumeric characters on them. The two sets contain different frames every time a new CAPTCHA is generated, so the number of frames for each character set is not fixed. Finally, the two sets of frames are combined to form a gif animation, completing the process of generating the CAPTCHA challenge. The output is an animation that shows a group of animated characters for a few seconds and then replaces the characters with another set of characters for the next few seconds. The flowchart of the proposed CAPTCHA scheme is shown in Fig. 3.

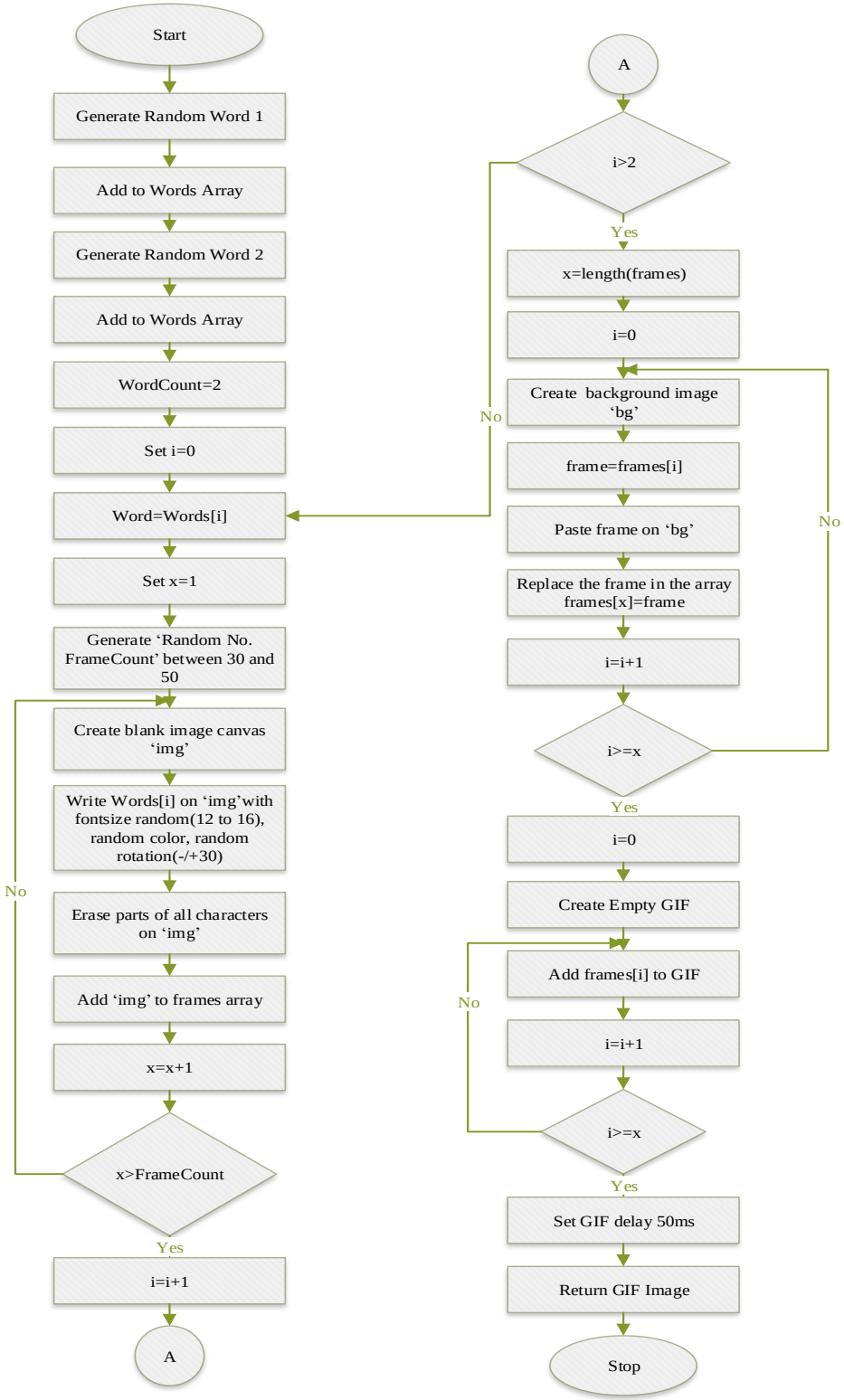


Fig. 3. The Flowchart shows the Implementation of the Proposed CAPTCHA Scheme.

The flowchart starts with generating two words or sets of random characters that will be displayed in the proposed CAPTCHA. The two words are stored in an array. Then, for each word, an integer variable FrameCount is created and its value is set randomly between 30 and 50. Then a blank/transparent image is created and the first word in the array is written on the image using a random font size between 12 and 16 using a randomly created colour and the text is printed on the image at an angle, the degree of which is chosen randomly between -30 and +30 degrees. Then parts of the image are cleared so that the text is not readable. After that, the image is pushed into an array named Frames. This process is repeated FrameCount times, thereby creating a random number of frames, between 30 and 50. This process is repeated for the second word in the array of Words. This results in the creation of two sets of frames for two randomly chosen sets of characters. Both sets contain between 30 and 50 frames. Then, for each frame in the Frames array, a random background image is created with some random noise and colours. One by one, the two sets of frames are pasted on top of the random backgrounds, resulting in a set of frames containing partially visible text characters on random background images. Finally, all the frames are joined together and converted into a GIF image with a frame delay of 50ms between the frames. The resulting GIF image is output and is the desired CAPTCHA scheme proposed in this research.

IV. Implementation

The proposed CAPTCHA has been implemented and tested in this research. The implementation was done in PHP language using GD library functions [17]. The GD library allows creating blank images using the imagecreatetruecolor() function on which the random text characters are printing using imagettftext() function, with random text colour and random font size, from a pre-determined size range at a random angle of rotation. The top and left pixel location for printing the text is also chosen randomly for every set of frames. The colour, font size and rotation angle and location are passed as parameters to the imagettftext() function. Then, using the imagecolortransparent() function, the system is instructed to render a pre-defined RGB colour (Ct) transparently. After that, parts of the text characters pasted on the image are removed and made transparent by adding rectangular shapes of Ct colour over the text. Fig. 4. shows some of the frames with partially visible text:

The frames thus created are pasted on randomly generated background images using the imagecopy() function. The frames are then combined and encoded as a gif image producing the final animated CAPTCHA.

Fig. 4. (a) shows a simple text printed on a blank image and (b) to (f) show a few frames with different parts of the text trimmed off. It can be noticed that the text is not visible or barely recognizable in these frames. Fig. 5. shows a few frames of the text over random background images, which also do not have fully visible text characters. It is impossible to show a screenshot of the CAPTCHA in motion, the way the human eye perceives it because a screenshot can only show a single frame.

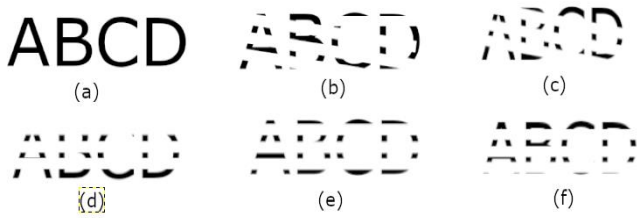


Fig. 4. A Sample Actual Text Frame and Frames with Partially Visible Text.

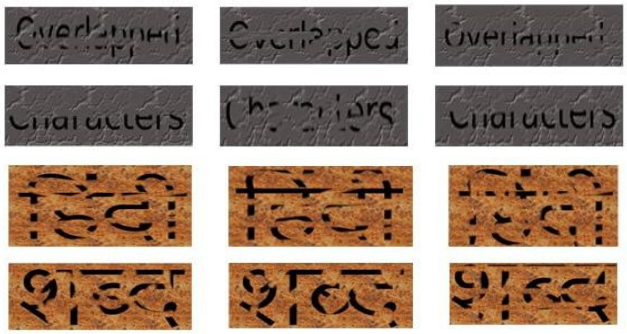


Fig. 5. Sample Frames from the Final CAPTCHA Animation.

The proposed CAPTCHA has been implemented and tested in English and Hindi languages and can be implemented in any other language. Furthermore, the CAPTCHA scheme was integrated and tested with a Multilingual CAPTCHA system [18], [19], [20], wherein the proposed CAPTCHA could create challenges in all the supported languages.

V. Security Features and Analysis

The animated CAPTCHA can be broken down into individual frames, and the individual frames can be subject to various captcha breaking algorithms, including segmentation, etc. The motion CAPTCHA is challenging to break using OCR or AI techniques as the recognition of characters from animation is difficult. Separate frames of the gif are useless for the OCR script. Background and noise randomization increase the problem for bots. The fast frame rate makes the characters seem intact, while in reality, the characters are NOT complete in ANY of the frames separately. The characters are incomplete, and therefore, moderate character distortion will suffice. Further, the presence of two sets of characters overlapped in the same place doubles the challenge of breaking the CAPTCHA because the gif image contains parts of two different characters at the same place, overlapped, in different frames. So, the challenge for any bot trying to break the CAPTCHA is to recognize two overlapped, partially visible characters from the same location in the various frame. If a bot tries to obtain multiple parts of a character from a given location, from multiple frames, it will have to figure out which parts belong to which character set. The following attacking techniques have been considered while designing the proposed CAPTCHA:

A. Segmentation

Segmentation techniques are used to break the text on a CAPTCHA image into various characters. The proposed CAPTCHA uses rotation and change of position of characters,

the background and foreground sets of lines—also, the entire set of characters' changes somewhere at the middle of the animation. The partially trimmed characters make the text on individual frames unrecognizable, even to the human eye. These features make the segmentation of characters extremely difficult, if not impossible.

B. Number of Characters

Common breaking techniques start by counting the number of characters on a CAPTCHA image and dividing the image into that many pieces. This makes it easy to extract a single character from a piece of the image. This research suggests using a different number of randomly chosen characters every time the CAPTCHA is requested. This makes it difficult to predict the number of characters on a CAPTCHA image, and thus breaking the image into pieces doesn't guarantee a single character on every part of the image. Furthermore, it is suggested to use a different number of characters for the two sets of characters proposed in this research. The varying font size on other framesets and CAPTCHA images further complicate the attack process.

C. Position

The starting position of the text on the proposed CAPTCHA varies every time, and over multiple frames, the characters don't stay at a single position but keep moving slightly and slightly, changing the rotation. This way, predicting the position of characters isn't possible in the proposed CAPTCHA, which mitigates a range of attacks.

D. Proper use of Color

Incorrect use of colours can adversely affect the security of any type of CAPTCHA. This research proposes multi-colour background objects and lines so that reducing the number of colours in frames doesn't reveal helpful information about the background and text colours.

E. Number of Frames

The proposed CAPTCHA has a different number of frames every time a CAPTCHA is generated. As the proposed CAPTCHA displays two sets of characters during the animation, it is possible to predict that a different set of characters becomes visible halfway, i.e. after 50% of the frames. Therefore, the proposed CAPTCHA distributes the frames among the two sets of characters unevenly, using a different number of frames per character set. Furthermore, the distribution varies per CAPTCHA generated; therefore, the two-character sets don't have the same number of frames. This makes it difficult for any attack to determine which frame the character set changes.

F. Frame Delay

The frames in the proposed CAPTCHA are delayed unevenly. The frame rate varies randomly between 1 and 10 milliseconds for every frame. This complicates the prediction of the rate of change of various pixel characters for any breaking attack.

G. PDM Attack

The attack based on Pixel Delay Map (PDM) work assumes that the characters required to be recognized by human users will be displayed for a longer duration of time than other moving parts of an animated CAPTCHA. The pixels that don't change colour for a fixed period are mapped to determine the location of the text characters. The proposed CAPTCHA doesn't have any moving background or foreground pixels. The entire background and partially visible sections of the foreground text are static and visible for the whole duration of the frame. Therefore, the PDM technique of breaking animated CAPTCHAs is useless on the proposed CAPTCHA scheme.

The security analysis of the proposed CAPTCHA was done using GSA CAPTCHA Breaker [21] and CAPTCHA Sniper [22], which are very popular and powerful CAPTCHA breaking tools. First, a set of 200 CAPTCHA animations was generated using the implementation of the proposed system, which was used for the security analysis. Then, one by one, the animations were broken down into individual frames or images. Each animation was broken down into 60 frames, resulting in 12000 frames. All the 12000 frames were analyzed individually using CAPTCHA Sniper and GSA CAPTCHA Breaker with a varied set of configurations and combinations of configurations. Because the proposed CAPTCHA scheme doesn't expose text character information in individual frames, these tools did not recognize a single character from 12000 frames.

The only way to break the proposed CAPTCHA is to collect individual text character information from more than a few consecutive frames. First, the selected frames must be processed to clear the background from the frames and leave the visible sections of the text untouched. Then the parts of the text characters must be extracted from individual frames and combined in a single image to form recognizable text characters. The proposed CAPTCHA mitigates this type of attack by slightly changing the location of the text by a few pixels in every frame. However, the change of location is performed randomly in the x and y axes. Therefore, if collected over multiple frames and combined, the parts of the text will result in a heap of small chunks of characters, completely unrecognizable, as shown in Fig. 6. The proposed CAPTCHA was tested against the breaking method discussed above. To simplify the attack, the step for adding the background image to the frames was skipped when generating the CAPTCHA, and the individual frames were exported to transparent backgrounded PNG images. The first step in breaking the CAPTCHA by removing the background objects and noise was not required. The frames thus contained partially visible text characters on a clear transparent background. The frames were then superimposed on each other to combine the partially visible parts of the text, expecting to get the full-text characters. The resulting image was then read using Tesseract OCR [23], [24], which didn't succeed in recognizing any characters.

Fig. 6. Sample Actual Character and Character Information Combined from 15 Frames.

VI. USABILITY

Unlike traditional static image-based CAPTCHA, the text characters in the individual frames in the proposed CAPTCHA don't need any transformation, deformation, or visual effects to make it difficult for bots that use various text extraction and recognition algorithms and OCR technologies to recognize the characters. It is an illusion created by the animation in the human brain that the text becomes visible and recognizable to the humans. At the same time, in reality, the frames are entirely useless for machines or bots. The user, therefore, doesn't have to work hard and try to recognize deformed text, as in the case of most types of CAPTCHAs.

An online usability study was carried out to test the usability and user-friendliness of the CAPTCHA images. Five hundred volunteers were asked in different social media groups to solve the proposed CAPTCHA challenge, with each user asked to solve five challenges. A web page was created, in which the user was asked about their preferred language, the options being English and Hindi. Based on the language chosen, the users were shown a second page with five CAPTCHA challenges in their chosen language, each challenge having a textbox for entering the answer. The responses were verified and the answers stored in the backend database. As shown in the Table 1, five (5) English language CAPTCHAs were attempted by 350 English-knowing users, which is a total of 1750 attempts. Out of the 1750 attempts, 1732 were correctly solved and 18 attempts were unsuccessful. For Hindi, 5 CAPTCHA challenges were solved by 150 users which sums up to a total of 750 attempts. Out of the 750 attempts, 741 were correctly solved and 9 were failed attempts. In combination of both languages, 2473 attempts were successful out of a total of 2500 attempts, giving 98.92% success rate.

TABLE I. USABILITY STUDY OF PROPOSED CAPTCHA

Language	**No. of Users**	**Challenges per User**	**Total Attempts**	**Solved**	**Failed**	**Success %age**
English	350	5	1750	1732	18	98.97
Hindi	150	5	750	741	9	98.8
Total	**500**	**5**	**2500**	**2473**	**27**	**98.92**

VII. CONCLUSION AND FUTURE SCOPE

Animated CAPTCHAs are an effective way to keep bots away and prevent online services from being misused. But they are vulnerable to breaking attacks and are broken easily by analyzing individual animation frames. This work discusses various available animated CAPTCHAs and techniques that have been used to break them. Animated CAPTCHAs, however, have the potential to do that can be done using static images. Therefore, a new animated CAPTCHA scheme has been proposed and implemented that makes use of the animation techniques and the natural behaviour of the retina of an eye to propose a very effective CAPTCHA scheme, which is immune to the traditional animated CAPTCHA breaking attacks and techniques. The CAPTCHA has been designed keeping the security loopholes in mind that weaken the animated CAPTCHAs. The proposed CAPTCHA has been tested against possible attacks programmatically using GSA CAPTCHA Breaker and CAPTCHA Snipper. The proposed CAPTCHA is user-friendly, easy to use, secure and innovative, and easy to implement. The traditional animated CAPTCHAs display the characters in only specific frames, thereby making users wait and spend quite a bit of time trying to recognize the characters. The proposed CAPTCHA text is visible to the user throughout the animation cycle. It does not require any special browser plugins to display the animation because it is created as a gif animation.

The phenomenon of image retention of the retina of Human eye is an extremely unique feature which has been used in this research to distinguish a human from a computer by implementing and successfully testing a novel CAPTCHA challenge based on its special capability. Therefore, we believe that this concept opens up possibilities for future research towards designing novel, highly secure and user friendly CAPTCHA challenges using this phenomenon.

ACKNOWLEDGEMENT

This work has been supported by Science and Engineering Research Board (SERB), Department of Science and Technology (DST), Government of India under its file no. EMR/2016/006987.

REFERENCES

[1] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. "CAPTCHA: Using Hard AI Problems for Security," InE. Biham, editor, EUROCRYPT, volume 2656 of Lecture Notes in Computer Science, pages 294–311. Springer,2003.

[2] K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. "Building Segmentation Based Human-Friendly Human Interaction Proofs (HIPs)," In H. S. Baird and D. P. Lopresti, editors, HIP, volume 3517 of Lecture Notes in Computer Science, pages 1–26. Springer, 2005.

[3] M.T. Banday, N.A. Shah, N.A, "Image Flip CAPTCHA," ISeCure, The ISC International Journal of Information Security, Iranian Society of Cryptology, Tehran, Iran, ISSN 2008-2045 and 2008-3076, 1(2), pp. 103-121, 2009.

[4] A.R.Shah, M. T. Banday, S.A.Sheikh. "Design of a Drag and Touch Multilingual Universal CAPTCHA Challenge", Department of Computer Science & Engineering, ABES Engineering College, Ghaziabad on 22-23 Feb 2019, Springer.

[5] M.Kumar, M. K. Jindal and Munish Kumar, "A Systematic Survey on CAPTCHA Recognition: Types, Creation and Breaking Techniques," Archives of Computational Methods in Engineering, Springer, June 2021

[6] O. Starostenko, C. Cruz-Perez, F. Uceda-Ponga, et al., "Breaking text-based CAPTCHAs with variable word and character orientation". Pattern Recognition, 2015, 48(4): 1101-1112.

[7] G. Mori, J. Malik. "Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA," International Conference on Computer Vision and Pattern Recognition Proceedings, Washington, 2003, 134-141.

[8] I.Akrout, A.Feriani, M.Akrout, "Hacking Google reCAPTCHA v3 using Reinforcement Learning," arxiv: cs.LG/1903.01003.

[9] S. Sivakorn, I. Polakis, A.D. Keromytis, "I am Robot: (Deep) Learning to Break Semantic Im-age CAPTCHAs," In 2016 IEEE Eur. Symp. Secure. Priv. (EuroS P), pp. 388–403 (2016). DOI10.1109/EuroSP.2016.37.

[10] W. Chow, W . Susilo W, "AniCAP: an animated 3D CAPTCHA Scheme based on motion parallax," In: Proceedings of 10th international conference on cryptology and network security, pp 255–271, 2011.

[11] S. Kim, S. Choi, " DotCHA: A 3D Text-based Scatter-Type CAPTCHA", Web Engineering; Bakaev, M., Frasincar, F., Ko, I.Y., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 238–252.

[12] http://www.hellocaptcha.com/; accessed on May 2021.

[13] V. Nguyen, Y. Chow, W. Susilo, "Breaking an animated CAPTCHA scheme", In International conference on applied cryptography and network security, 2012, pp 12–29.

[14] S. S.A. Shah, R. A.Shaikh, R. H. Arain, "Reading the Moving Text in Animated Text-BasedCAPTCHAs," In International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 9, No. 12, 2018.

[15] L.A. Leiva, F. Álvaro, µcaptcha: "Human Interaction Proofs Tailored to Touch-Capable Devices via Math Handwriting," In: International Journal of Human-Computer Interaction,2015, 31(7), pp.457–471.

[16] A. Shumilov, A. Philippovich, "Gesture-based animated CAPTCHA", In Information and Computer Security; Bingley Vol. 24, Iss. 3, pp. 242-254, 2016.

[17] S. Stobart, M. Vassileiou, "GD Library," In PHP and MySQL Manual. Springer Professional Computing. Springer, London, (2004).

[18] M.T. Banday, S.A. Sheikh, "Design of CAPTCHA Script for Indian Regional Websites," In Communications in Computer and Information Science Security in Computing and Communications, Springer Berlin Heidelberg, pp. 98–109, 2013.

[19] M.T.Banday, S.A. Sheikh. "Design of Secure Multilingual CAPTCHA Script", International Journal of Web Portals, IGI Global Vol. 7, No. 4, pp. 1-27, 2015.

[20] M.T.Banday, S.A. Sheikh. "Service Framework for Dynamic Multilingual CAPTCHA Challenges: IN-CAPTCHA" 2014 International Conference on Advances in Electronics, Computers and Communications (ICAECC-2014) 10-11 October 2014, Reva Institute of Technology and Management, Bangalore, India, published by IEEE, ISBN:9781479954971, pp. 1-6.

[21] GSA CAPTCHA Breaker, "https://www.gsa-online.de/product/captcha_breaker/", accessed on October 2021

[22] Captcha Sniper (CS), "https://www.captchasniper.com/", accessed on December 2021.

[23] Tesseract OCR, " https://github.com/tesseract-ocr/" accessed on Nov. 2021

[24] R. Smith, An Overview of the Tesseract OCR Engine, In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), IEEE Conference, 23-25 September 2007, Brazil.