A Multibiometrics-based CAPTCHA for Improved Online Security

Brian M. Powell
West Virginia University
brian.powell@mail.wvu.edu

Abhishek Kumar
IIIT-Delhi
abhishek10004@iiitd.ac.in

Jatin Thapar
IIIT-Delhi
jatin10033@iiitd.ac.in

Gaurav Goswami
IIIT-Delhi
gauravgs@iiitd.ac.in

Mayank Vatsa
IIIT-Delhi
mayank@iiitd.ac.in

Richa Singh
IIIT-Delhi
rsingh@iiitd.ac.in

Afzel Noore
West Virginia University
afzel.noore@mail.wvu.edu

Abstract

CAPTCHA (Completely Automated Public Turing Test to tell Computers and Humans Apart) have been a common tool for preventing unauthorized access to websites for over a decade, but increasingly sophisticated optical character recognition algorithms and attack strategies have rendered traditional CAPTCHAs insecure. In this paper, we propose a new CAPTCHA incorporating multiple biometric modalities. Users are asked to identify faces, eyes, and fingerprints in a complex composite image. With over 1,900 volunteers and 30,000+ attempts, the proposed approach achieves high human accuracy while being resistant to existing attacks on CAPTCHAs and to detection by state-of-the-art software.

1. Introduction

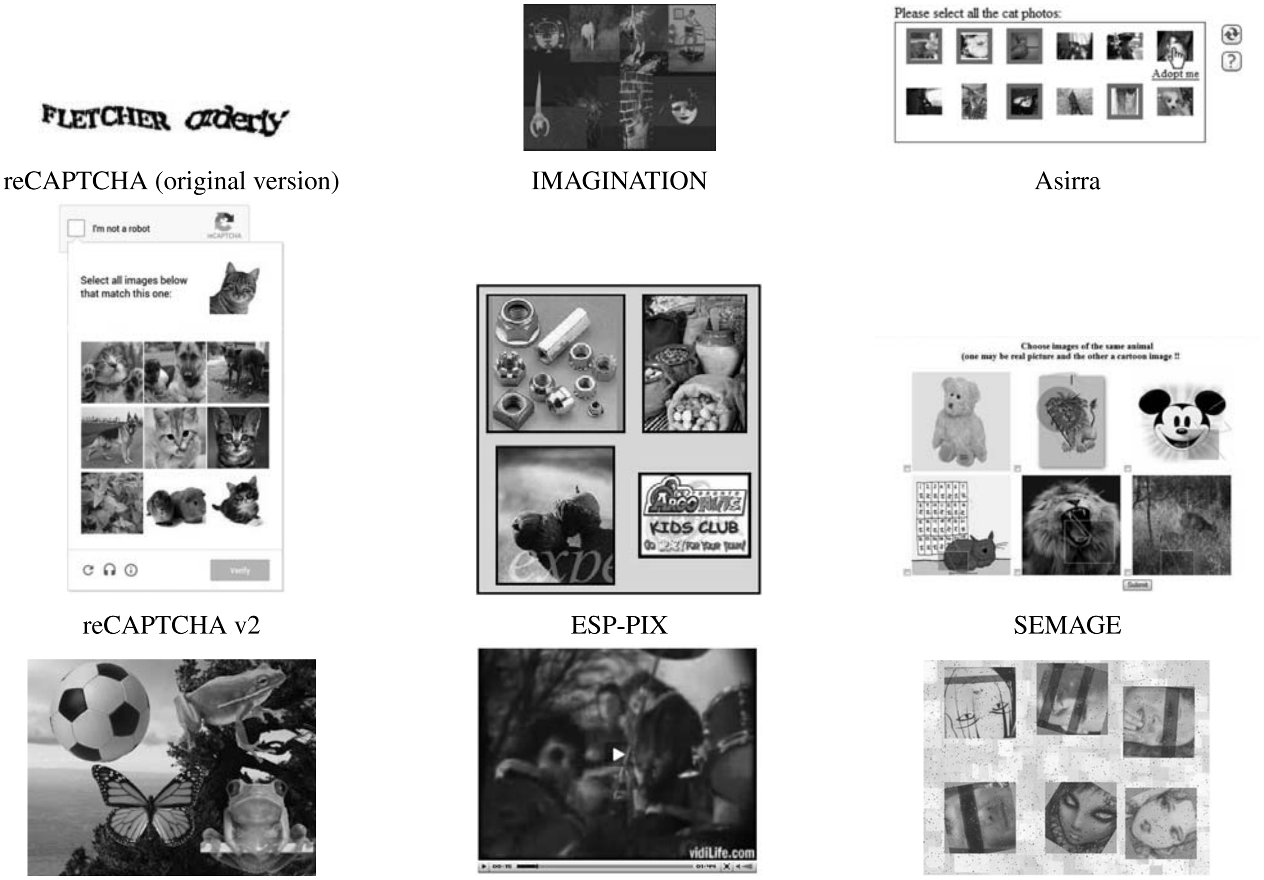
The Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is a type of challenge-response test used to distinguish between human users and automated scripts [18]. The tests have become quite common on websites, where they are used to restrict access to resources that should only be used by humans and not software bots. Common applications include preventing spam posts in commenting systems, ensuring that only human users are able to register for accounts, and preventing brute force login attacks on user accounts. They can also be used to mitigate Denial of Service (DoS) attacks [19]. Given the growing importance of cloud-based computing and the need to ensure these systems remain secure and highly available, CAPTCHAs are an important tool in protecting online resources.

Successful CAPTCHA implementations must present would-be users with a test that is difficult for automated software to solve while being easy for legitimate human users to complete. Existing CAPTCHA tests generally belong to one of the three categories: (1) text-based, (2) video and audio-based, and (3) image-based. Figure 1 shows some examples of common existing CAPTCHAs.

Text-based CAPTCHAs are the most frequently used type [16]. The original version of reCAPTCHA is one common implementation [24]. Text-based CAPTCHAs present users with distorted images of text that they must decipher and type-in. As optical character recognition (OCR) technology has improved, the intensity of distortions that must be applied to CAPTCHAs for them to remain unrecognizable by OCR has increased. This has reached a point where humans frequently have difficulty solving many text-based CAPTCHA tests [3]. Even when they are solvable, text-based CAPTCHAs can be difficult for mobile users to answer due to the difficulty of entering arbitrary text strings using on-screen keyboards [11].

The second major category of CAPTCHAs involve those which use video or audio. Video-based CAPTCHAs ask users to specify keywords that describe the content of a video clip [10]. While effective, the video clips require significant bandwidth and may be difficult to view on mobile devices. Audio-based CAPTCHAs generally ask users to enter words and letters that are spoken in a recording. These tests are frequently used as a fallback for other CAPTCHAs for users with visual impairments, but they are subject to defeat using novel analysis techniques [1, 2]. Due to these limitations, audio and video CAPTCHAs have seen limited real world use.

Image-based CAPTCHAs have become popular because


FLETCHER orderly
Please select all the cat photos:
Adopt me
I'm not a robot
reCAPTCHA
Select all images below that match this one:
Verify
KIDS CLUB
Choose images of the same animal
(one may be real picture and the other a cartoon image !!)
Submit
vidiLife.com

reCAPTCHA (original version) IMAGINATION Asirra

reCAPTCHA v2 ESP-PIX SEMAGE

Scene Tagging CAPTCHA Video CAPTCHA FaceDCAPTCHA

Figure 1. Examples of existing CAPTCHAs.

of the security and usability shortcomings of the other types of CAPTCHAs. Most implementations rely on categorization tasks. Asirra and reCAPTCHA v2, Google's replacement for the original text-based reCAPTCHA, present users with a small set of images and a keyword [4, 6]. Users are asked to select the images matching the keyword to solve the CAPTCHA. Other image-based CAPTCHAs, such as ESP-PIX, take the opposite approach by asking users to specify a descriptor to categorize a set of presented images [23]. A third format, used by SEMAGE, asks users to select matching images within a set [21]. All of these implementations have significant security weaknesses because the small number of images and/or keywords involved make them subject to brute force and image classification attacks [6, 8].

A different form of image-based CAPTCHA relies upon users identifying items embedded within a composite image. The Scene Tagging and IMAGINATION CAPTCHAs require users to understand the items embedded in the composite image and their relative placement. Scene Tagging CAPTCHA asks users to identify which objects are placed next to other objects or the quantity of a specific type of embedded object [13]. IMAGINATION requires users to find the center of an embedded image and then to categorize the item [5]. These tests are vulnerable to attacks based on segmentation and object recognition [13, 25].

FaceDCAPTCHA also requires users to identify images embedded in a composite background, but it additionally incorporates a biometrics modality into its test. Users are presented with a composite image containing face images from human photographs, cartoons, and sketches. To solve the CAPTCHA, users must identify the images which are of actual human faces [9]. While FaceDCAPTCHA's use of a single biometric modality (faces) renders it vulnerable to attacks based on segmentation and deep learning [7], it does suggest an avenue for further research which we pursue in this paper. By incorporating multiple biometric modalities, it is possible to design a CAPTCHA which is significantly more difficult to attack. In the proposed approach, users are presented with composite images that contain faces, fingerprints, and eyes. They are also given a selection task, rendered as text in the CAPTCHA image, identifying the

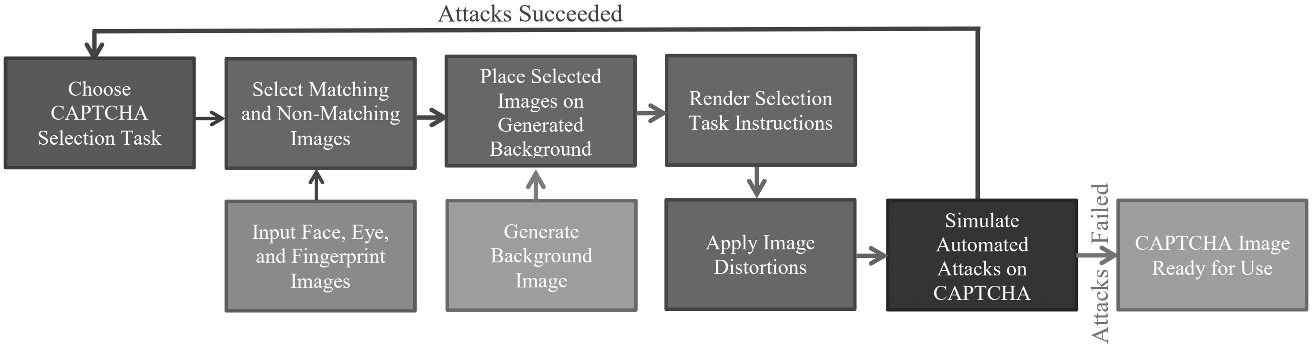

Attacks Succeeded
Choose CAPTCHA Selection Task
Select Matching and Non-Matching Images
Place Selected Images on Generated Background
Render Selection Task Instructions
Input Face, Eye, and Fingerprint Images
Generate Background Image
Apply Image Distortions
Simulate Automated Attacks on CAPTCHA
Attacks Failed
CAPTCHA Image Ready for Use

Figure 2. Steps involved in generating the proposed multibiometic CAPTCHA.

types of embedded images they must select to solve the CAPTCHA. This addition of a changeable task adds a new layer of difficulty for would-be attackers. They must be able to perform both natural language processing and computer vision tasks in order to successfully attack the CAPTCHA. While these tasks are challenging for software, testing involving over 1,900 volunteers has shown they can be readily completed by humans.

2. Proposed Multibiometrics CAPTCHA

Despite extensive research into the problems of face, fingerprint, and eye detection, these remain challenging tasks for computers to perform. Humans, however, are quite good at these functions. The proposed Multibiometrics (MB) CAPTCHA takes advantage of this relative difference in abilities by incorporating multimodal biometrics tests that require users to detect faces, fingerprints, and eyes in complex composite images. To solve the CAPTCHA, users must correctly click or tap on all embedded images which fulfill the selection task specified by instructions in the rendered CAPTCHA image. If the user correctly selects all matching embedded images without any mis-selections, they pass the CAPTCHA test and are assumed to be human.

2.1. CAPTCHA Generation Process

The proposed CAPTCHA generation process can be represented as,

$$C = F(I_{face}, I_{eye}, I_{fingerprint}, \phi, d) \tag{1}$$

where function F represents the series of image processing operations required to generate CAPTCHA C. C contains embedded images selected from a combination of sets I_{face}, I_{eye}, and $I_{fingerprint}$ depending upon a selection task chosen from ϕ. d represents a difficulty level from 1 to 5 used to determine distortions and image characteristics for the rendered CAPTCHA. CAPTCHA with higher difficulty levels are intended to be more challenging to complete.

As shown in Figure 2, there are several steps involved in generation of the proposed CAPTCHA. They are detailed below.

2.1.1 Background Generation

The generation of new CAPTCHA tests begins with the creation of a 800×400 pixel image containing a randomly shaded grayscale background. Between 900 and 1,500 geometric shapes, either circles, rectangles, or crosses, are semi-transparently overlaid in random locations. Each individual shape is of a randomly selected size and grayscale shade. This complex pattern is intended to create false targets for object detection algorithms that may be used to attack the CAPTCHA. During the testing conducted as part of this research, automated algorithms detected many false faces, fingerprints, and eyes in the background that did not actually exist. Since mis-selections are treated as incorrect attempts, these false images reduce the likelihood that an attack on the CAPTCHA is successful.

After all of the geometric shapes are placed, dilation is repeatedly performed on the entire background. This operation sets each pixel to the maximum (lightest) value of its adjoining pixels. It has the effect of creating a ragged, irregular border instead of crisp lines and reduces the success rate of edge detection and segmentation-based attacks.

The dilation operation can be represented as,

$$I \oplus S = \bigcup_{p \in I} S_p \tag{2}$$

where I is the image being dilated, S is a 3×3 structuring element, and S_p is the value of the structuring element centered at pixel location p.

Figure 3 shows an example of a rendered background after dilation is performed.

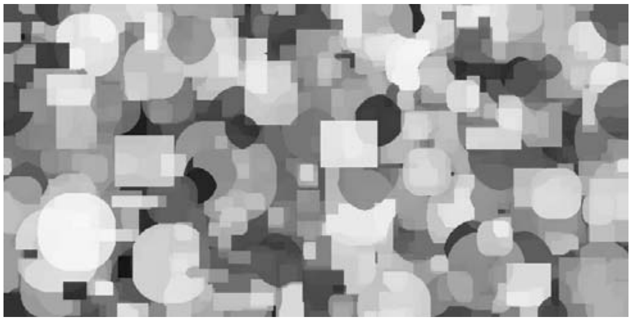


Figure 3. Example of a rendered background after dilation.

2.1.2 Selection Task Assignment

To solve the proposed MB-CAPTCHA, users are required to select the embedded images that meet the requirements of a specified selection task. This step determines the selection task that will be used for a particular CAPTCHA image.

The process begins by randomly choosing to use either 1 or 2 selection tasks for the CAPTCHA. Next, the generation algorithm randomly picks specific tasks from these three options:

1. Eye selection, where users must locate all of the images of eyes embedded in the CAPTCHA.
2. Fingerprint selection, where users must locate all of the images of fingerprints embedded in the CAPTCHA.
3. Face selection, where users are asked to select all embedded face photographs matching a specified gender (male or female) or race (black or white) attribute. The attributes are selected at random.

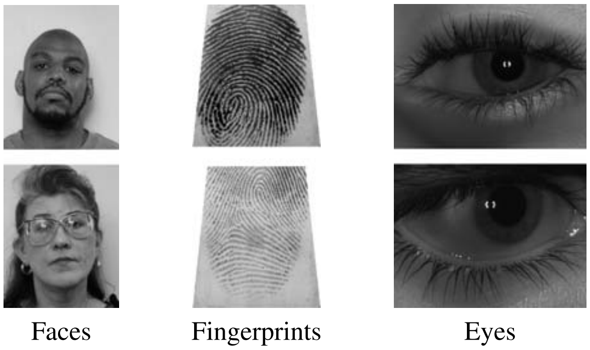
If two selection tasks are to be used for the CAPTCHA, the generation algorithm ensures two different tasks are selected.

2.1.3 Image Selection

After the selection tasks are chosen, the generation algorithm constructs two sets I_{match} and $I_{notmatch}$ based on images which match and do not match, respectively, the CAPTCHA's selection task requirements. The source eye, fingerprint, and face images are organized into the sets I_{match} and $I_{notmatch}$ as appropriate. Figure 4 shows examples of source images used in the generation process.

A total of 4 to 8 matching and non-matching images are selected to be embedded such that,

$$n_{total} = \Big\{ n_{match} + n_{notmatch} \, \Big| \, n_{match} \geq 2, n_{notmatch} \geq 2, 4 \geq n_{total} \geq 8 \Big\} \tag{3}$$



Faces Fingerprints Eyes

Figure 4. Examples of source images to be embedded.

n_{match}, $n_{notmatch}$, and n_{total} represent the number of matching, non-matching, and total embedded images, respectively. At least two matching images are required to ensure that a single guess cannot solve the CAPTCHA. Multiple non-matching images serve as false targets to reduce the likelihood that an automated algorithm will correctly solve the CAPTCHA.

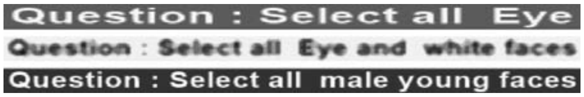
When face images are used, the images contain a background surrounding the face when difficulty level $d \leq 3$. There is no background surrounding the face when $d > 3$.

2.1.4 Image Placement

Each of the images to be embedded in the CAPTCHA is scaled to between 80 × 80 and 120 × 120 pixels in size to ensure there is room to embed the required number of images within the CAPTCHA. The images are also rotated a randomly selected amount between $[-60, 60)$ degrees. The rotation decreases the likelihood that detection algorithms will correctly identify the images once they are embedded in the background.

The image placement algorithm randomly selects locations for the embedded images. In choosing locations for placement, the algorithm ensures that the embedded images do not overlap each other or the edge of the CAPTCHA. It also maintains a 35-pixel margin at the top of the CAPTCHA for the selection task instructions to be placed.

Each image is embedded using alpha compositing (partial transparency) so that it blends into the background. This blending interferes with object detection since the embedded images may have different contrast, coloring, and edges than the algorithms are trained to detect.



Question : Select all Eye

Question : Select all Eye and white faces

Question : Select all male young faces

Figure 5. Instructions to be rendered in the CAPTCHA.

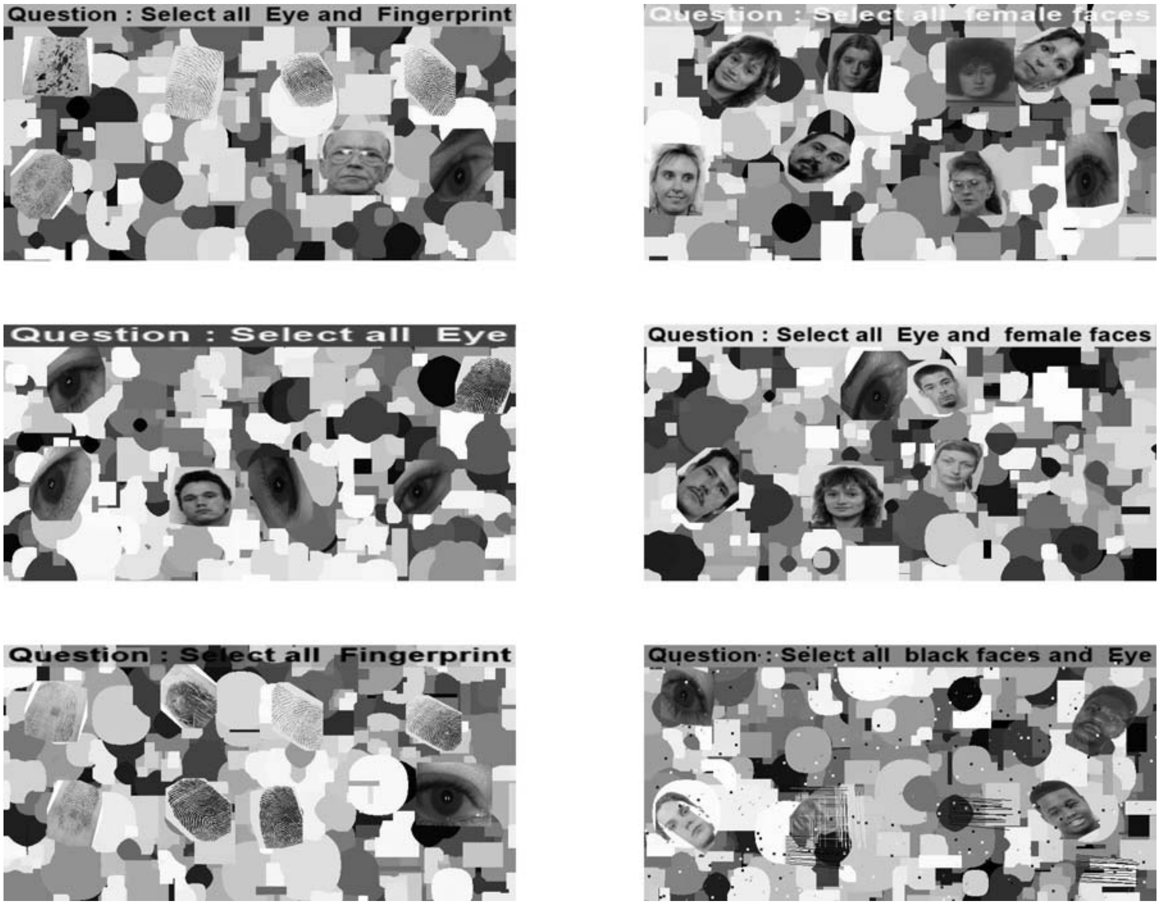

Question : Select all Eye and Fingerprint
Question : Select all female faces
Question : Select all Eye
Question : Select all Eye and female faces
Question : Select all Fingerprint
Question : Select all black faces and Eye

Figure 6. Samples of the proposed CAPTCHA. Users must select the images specified in the instructions to correctly solve the CAPTCHA.

2.1.5 Instructions Rendering

Once all the images have been placed, instructions containing the selection task to be used in completing the CAPTCHA are rendered. The instructions are displayed using the Arial font on a randomly shaded grayscale background with a randomly selected alpha channel (transparency) value. The color of the text varies upon the shade of the background: white text is used for darker backgrounds and black text is used for lighter backgrounds. For cases where difficulty level $d = \{1, 3, 5\}$, the text is additionally blurred to decrease the likelihood of optical character recognition (OCR) being able to successfully identify the selection task needed to complete the CAPTCHA.

The rendered text image is stretched to fit an area of 800×35 pixels, as shown in the examples in Figure 5. It is then embedded at the top of the CAPTCHA using alpha compositing. In cases where the alpha value is low, the grayscale shading appears transparent and the underlying complex background pattern shows through the text. This greatly reduces the success rate of OCR attacks in correctly identifying the selection task required for the CAPTCHA.

2.1.6 Image Distortion

Depending upon the difficulty level d used in generating the CAPTCHA, additional visual effects may be applied to the rendered CAPTCHA to distort the image. When $d = \{3, 5\}$, lines in various shades of gray may be drawn in random locations throughout the CAPTCHA in a striped or crosshatch pattern. Many small randomly shaded grayscale squares are also placed on the image when $d = \{3, 5\}$. These additional distortions are intended to interfere with the ability of detection algorithms to locate and segment the images embedded in the CAPTCHA or to perform OCR on the instructions.

2.1.7 Attack Simulation

After each CAPTCHA image is rendered, it undergoes a series of tests designed to ensure that it is not susceptible to attack by automated algorithms. The first test involves using best-of-breed commercial OCR in an attempt to detect the selection task specified by the CAPTCHA's instructions. If the selection task can be correctly identified, further testing is required to ensure the CAPTCHA's embedded images cannot be automatically detected:

1. If the selection task requires that faces be identified, the Face++ web service is used to locate embedded faces [14]. This tool determines the likely race and gender for each face it locates. The detected values are compared against the actual embedded faces to see if the required faces were correctly found.

2. If the selection task requires that eyes be identified, a version of the Viola-Jones algorithm trained for eye detection is used to locate them [22]. The detected locations are compared against the actual eye locations.

3. If the selection task requires that fingerprints be identified, the SourceAFIS fingerprint recognition software is used to construct fingerprint templates based on the image [20]. The location of the templates' features are compared against the actual locations of the embedded fingerprints to see if they match.

Face++, Viola-Jones, and SourceAFIS are used as they can be performed in near real time. Fast performance at this stage is critical since these tests are run in-the-loop during the CAPTCHA generation process, and as such, occur frequently.

In the event these tests were able to correctly determine the selection task and identify the images required to solve the CAPTCHA, the CAPTCHA image is discarded and the generation process begins anew. This ensures that publicly used CAPTCHA test images are resistant to automated attacks and viable as a security tool. Figure 6 shows examples of CAPTCHAs which have passed the simulated attacks and are ready for use.

3. Experimental Results and Analysis

The proposed CAPTCHA has been evaluated by over 1,900 human users. This section provides the details of the source image databases, participants, and protocol used in evaluating the proposed approach along with results and analysis.

3.1. Image Databases

The proposed approach uses three source databases to provide the face, fingerprint, and eye images used in generating the CAPTCHAs. The University of North Carolina Wilmington Craniofacial Morphology Database is used for face images [15]. Eye images are from the IIITD Multispectral Periocular Database [17]. Fingerprint images are from the FVC2004 database [12].

Table 1. Proposed CAPTCHA Success Rates by Criteria

Selection Task	**Human Success Rate**	**Attack Success Rate**
Eyes	90.5%	0.0%
Fingerprints	89.4%	0.0%
Faces of Specified Race, Eyes	87.9%	0.0%
Fingerprints, Eyes	83.7%	0.0%
Faces of Specified Race	80.0%	0.0%
Faces of Specified Gender and Race	72.7%	0.0%
Faces of Specified Gender	72.1%	0.0%
Faces of Specified Gender, Fingerprints	67.1%	0.0%
Faces of Specified Gender, Eyes	63.8%	0.0%
Faces of Specified Race, Fingerprints	62.8%	0.0%
Overall	**83.6%**	**0.0%**

3.2. Participants and Testing Protocol

The proposed MB-CAPTCHA was evaluated using 1,905 volunteers, all at least 18 years of age, using a large set of rendered CAPTCHA images. Volunteers attempted to access a website protected using the proposed CAPTCHA in an uncontrolled environment on a device of their choosing. Participants were free to use desktops, laptops, and mobile devices to complete the CAPTCHA. Only one CAPTCHA image was present on-screen at a time. Users who correctly solved the CAPTCHA were able to access the protected website, while those who did not were asked to try again until they were successful.

3.3. Analysis

Volunteers recorded a total of 30,664 attempts at solving the proposed CAPTCHA. The overall human success rate across all selection tasks was 83.6%, but Table 1 shows there was significant variation in success rate depending upon the exact selection task used. Humans were best able to solve CAPTCHAs where they were just asked to locate embedded items, such as eyes and fingerprints, rather than tasks where they had to both locate and categorize items, such as when selecting faces of specified genders or races. We found that users could identify the location of faces but sometimes failed to properly categorize them. Part of this difficulty may arise from ambiguity in the embedded images. For example, more than half of the users failed to identify the circled face in Figure 7 as being a man.

The CAPTCHAs' difficulty level and associated distor-

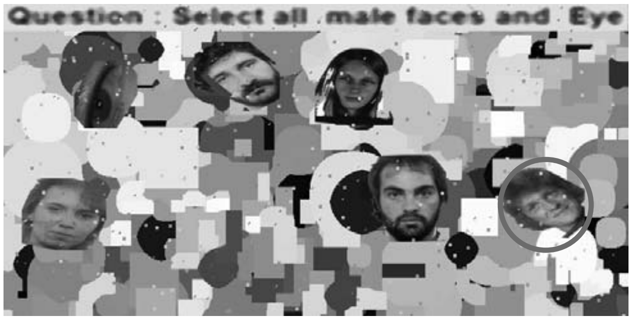


Figure 7. Many users failed to accurately identify the circled face.

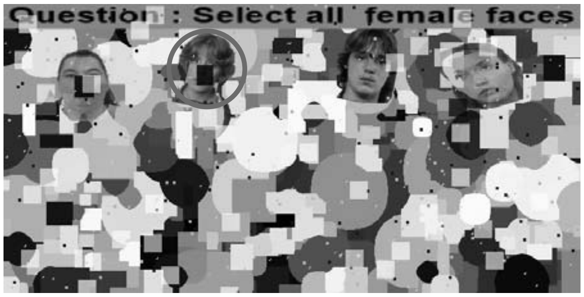


Figure 8. Many users failed to select the circled face due to the rectangle caused uncertainty about the gender.

tions seem to have minimal impact on humans' ability to successfully solve them. One situation where distortions do appear to hinder humans is when crosshatched lines or rectangles are placed on top of embedded images, as sometimes happens in CAPTCHAs with difficulty level $d = \{3, 5\}$. These distortions can make it difficult to detect the embedded object, or in the case of faces, to correctly identify their attributes. Figure 8 shows an example where a rectangle placed over an embedded image caused many users to fail to select the circled face. Otherwise, the impact of the difficulty levels and distortions was slight.

Since the proposed CAPTCHA's generation process is designed to remove images which are susceptible to object detection-based automated attacks, one of the most likely remaining avenues of attack involves brute force random guessing. We computed the probability of a brute force attack being successful. Each CAPTCHA contains 2-6 images which must be selected, each on average about 100×100 pixels in size, from an area 800×400 pixels in size. The chance of a single random guess at correctly answering the CAPTCHA is approximately,

$$\left(\frac{1}{6}\right)\prod_{i=0}^{6}\frac{(100)(100)i}{(800)(400)} = 0.000011\% \qquad (4)$$

As CAPTCHA test images are selected at random for each attempt, it is unlikely that a would-be attacker would see the same CAPTCHA test again for sometime after a failed attempt. When combined with the 1-in-10,000,000 likelihood of correctly solving an individual attempt, it becomes extremely time-consuming to use a brute force approach to defeat the proposed CAPTCHA.

4. Conclusion

This paper presents a novel approach for creating image-based CAPTCHAs to secure websites and online services. The proposed approach's multiple biometric modalities provide tests that are straightforward for humans to solve but remain difficult for automated attackers to successfully complete. The CAPTCHA's high human success rates (83.6% versus 70%-80% of common CAPTCHAs such as reCAPTCHA v1, MSN, and IMAGINATION [3, 5]), 0% effective automated attack success rate, and ease-of-use across a wide variety of devices are significant advantages over existing CAPTCHAs presently in use.

References

[1] J. P. Bigham and A. C. Cavender. Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In *Proc. 27th Int. Conf. on Human Factors in Comput. Systems*, pages 1829–1838, Boston, Massachusetts, 2009. ACM.

[2] E. Bursztein and S. Bethard. Decaptcha: Breaking 75% of eBay Audio CAPTCHAs. In *Proc. 3rd USENIX Workshop on Offensive Technologies*, Montreal, Canada, Aug. 2009.

[3] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky. How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation. In *Proc. 2010 IEEE Symp. on Security and Privacy*, volume 0, pages 399–413, Los Alamitos, California, 2010. IEEE Computer Society.

[4] A. Chowdry. How Google Is Making reCAPTCHA Simpler. *Forbes*, Dec. 2014.

[5] R. Datta, J. Li, and J. Z. Wang. Exploiting the Human-Machine Gap in Image Recognition for Designing CAPTCHAs. *IEEE Trans. Inf. Forensics Security*, 4(3):504–518, 2009.

[6] J. Elson, J. Douceur, J. Howell, and J. Saul. Asirra: a CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In *Proc. 14th ACM Conf. on Comput. and Commun. Security*, pages 366–374, Alexandria, Virginia, Oct. 2007. ACM.

[7] H. Gao, L. Lei, X. Zhou, J. Li, and X. Liu. The Robustness of Face-Based CAPTCHAs. In *Proc. 2015 IEEE Int. Conf. on Comput. and Inform. Technology*, pages 2248–2255, Liverpool, England, Oct. 2015. IEEE.

[8] P. Golle. Machine Learning Attacks Against the Asirra CAPTCHA. In *Proc. 15th ACM Conf. on Comput. and Commun. Security*, pages 535–542, New York, New York, Oct. 2008. ACM.

[9] G. Goswami, B. M. Powell, M. Vatsa, R. Singh, and A. Noore. FaceDCAPTCHA: Face Detection based Color Image CAPTCHA. *Future Generation Comput. Syst.*, 31:59–68, Feb. 2014.

[10] K. A. Kluever and R. Zanibbi. Balancing Usability and Security in a Video CAPTCHA. In *Proc. 5th Symp. on Usable Privacy and Security*, pages 1–11, Mountain View, California, July 2009. ACM.

[11] M. Kolsch and M. Turk. Keyboards without keyboards: A survey of virtual keyboards. In *Proc. Workshop on Sensing and Input for Media-centric Systems*, Santa Barbara, California, June 2002.

[12] D. Maltoni, M. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, London, England, 2nd edition, 2009.

[13] P. Matthews and C. C. Zou. Scene tagging: image-based CAPTCHA using image composition and object relationships. In *Proc. 5th ACM Symp. on Inform., Comput. and Commun. Security*, pages 345–350, Beijing, China, 2010. ACM.

[14] Megvii, Inc. Face++: Leading Face Recognition on Cloud, 2015.

[15] K. Ricanek and T. Tesafaye. MORPH: a longitudinal image database of normal adult age-progression. In *Proc. 7th Int. Conf. on Automatic Face and Gesture Recognition*, pages 341–345, Southampton, England, Apr. 2006. IEEE.

[16] S. K. Saha, A. K. Nag, and D. Dasgupta. Human-Cognition-Based CAPTCHAs. *IT Professional*, 17(5):42–48, Sept. 2015.

[17] A. Sharma, S. Verma, M. Vatsa, and R. Singh. On Cross Spectral Periocular Recognition. In *Proc. 2014 IEEE Int. Conf. on Image Processing*, Paris, France, Oct. 2014. IEEE.

[18] S. Shirali-Shahreza and M. H. Shirali-Shahreza. Bibliography of works done on CAPTCHA. In *Proc. 3rd Int. Conf. on Intelligent System and Knowledge Eng.*, volume 1, pages 205–210, Xiamen, China, 2008.

[19] A. Stavrou, D. L. Cook, W. G. Morein, A. D. Keromytis, V. Misra, and D. Rubenstein. WebSOS: an overlay-based system for protecting web servers from denial of service attacks. *Computer Networks*, 48(5):781–807, Aug. 2005.

[20] R. Vazan. SourceAFIS, 2015.

[21] S. Vikram, Y. Fan, and G. Gu. SEMAGE. In *Proc. 27th Annu. Comput. Security Applicat. Conf.*, page 237, Orlando, FL, Dec. 2011. ACM Press.

[22] P. Viola and M. Jones. Robust real-time object detection. *Int. J. Comput. Vision*, 57(2):137–154, 2002.

[23] L. von Ahn, M. Blum, and N. Hopper. ESP-PIX, 2004.

[24] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468, Sept. 2008.

[25] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition CAPTCHAs. In *Proc. 14th ACM Conference on Computer and Communications Security*, pages 187–200, Chicago, Illinois, Oct. 2010. ACM.