EURASIP Journal on
Information Security

RESEARCH

Open Access

Human-artificial intelligence approaches for secure analysis in CAPTCHA codes

Nghia Dinh[1*] and Lidia Ogiela[2]

Abstract

CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) has long been used to keep automated bots from misusing web services by leveraging human-artificial intelligence (HAI) interactions to distinguish whether the user is a human or a computer program. Various CAPTCHA schemes have been proposed over the years, principally to increase usability and security against emerging bots and hackers performing malicious operations. However, automated attacks have effectively cracked all common conventional schemes, and the majority of present CAPTCHA methods are also vulnerable to human-assisted relay attacks. Invisible reCAPTCHA and some approaches have not yet been cracked. However, with the introduction of fourth-generation bots accurately mimicking human behavior, a secure CAPTCHA would be hardly designed without additional special devices. Almost all cognitive-based CAPTCHAs with sensor support have not yet been compromised by automated attacks. However, they are still compromised to human-assisted relay attacks due to having a limited number of challenges and can be only solved using trusted devices. Obviously, cognitive-based CAPTCHA schemes have an advantage over other schemes in the race against security attacks. In this study, as a strong starting point for creating future secure and usable CAPTCHA schemes, we have offered an overview analysis of HAI between computer users and computers under the security aspects of open problems, difficulties, and opportunities of current CAPTCHA schemes.

Keywords: Human-artificial intelligence, CAPTCHA codes, Secure analysis

1 Introduction

CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) or HIP (*Human Interactive Proof*) is an automatic security mechanism to distinguish whether the user is a human or a computer program. It creates and scores tests that can be solved by humans but are beyond the capabilities of present computer programs. It has evolved into the most generally utilized standard security measure for preventing automated computer program attacks. With the growth of Web services, Denial of Service (DoS) attacks by malicious automated programs have become a severe issue, and the Turing test has become a crucial approach for distinguishing people from dangerous automated programs. A human judge is authorized to pose a series of questions to two players, one of which was a computer and the other a human, and tell them apart in the original Turing test. CAPTCHA, like the Turing test, distinguishes humans from computers, but the judge is now a machine. In general, CAPTCHA is a cryptographic protocol [1] whose underlying hardness assumption is based on an AI problem. CAPTCHA implies a win-win situation: either the captcha is not broken and there is a way to differentiate humans from computers, or the captcha is broken, and a hard AI problem is solved. CAPTCHA is usually a simple visual test or puzzle that a human can complete without much difficulty, but an automated program cannot understand. The test usually consists of letters, numbers, or their combination with overlapping and intersection. The CAPTCHA images may be distorted or shown against

*Correspondence: trong.nghia.dinh.st@vsb.cz

[1] VSB Technical University of Ostrava, 17. listopadu 15/2172, 708-33 Ostrava-Poruba, Czech Republic
Full list of author information is available at the end of the article

a complicated background to make them hard to be read by Optical Character Recognition (OCR) software. CAPTCHA has a wide variety of applications on the web and other applications such as Worms and Spam, Online Polls, Free Email Services, Preventing Dictionary Attacks and also plays a significant role in limiting usage rate.

HAI (*Human Artificial Intelligence*) researches the interactions between humans and computers, as well as the major phenomena that surround them. It denotes the usability characteristics that are firmly linked to the user interface and human factors. Hence, it is deeply involved with computer science, artificial intelligence, and cognitive psychology. The main concept in HAI is usability. From this perspective, puzzles like CAPTCHA, which humans can easily solve but computers find difficult, are an example of HAI. In this study, we provided an overview analysis of HAI under the security aspects of open concerns, difficulties, and opportunities of current CAPTCHA schemes. The remainder of this paper is organized as follows: Section II provides the taxonomy of CAPTCHA attacks. Section III describes CAPTCHA problem analysis. As a result, suggestions and recommendations are provided to build a good CAPTCHA in Section IV. Finally, Section V concludes the paper.

1.1 CAPTCHA evolution

The first person, Moni Naor [2], suggested theoretical approaches for distinguishing computers from humans. In 1997, the AltaVista web search engine was credited with being the first to use a CAPTCHA on the Internet [3]. Text-based CAPTCHAs were the leading technique in the early 2000s. A set of attacks were developed using image processing, pattern recognition, and machine learning (ML) algorithms to break popular text-based schemes [4]. Furthermore, anti-recognition and anti-segmentation algorithms were employed in an attempt to improve the security of existing text-based CAPTCHAs. In 2014, Google revealed that developments in AI technology could resolve distorted text variants with 99.8% [5]. Since 2004, computer vision (CV) problems, including image classification and recognition, were regarded as more difficult AI challenges than text recognition. Following that, many image-based CAPTCHA schemes with drag and drop, image selection, or sliding appeared in order to distinguish humans from computers. However, advanced CV and ML solutions aided in the defeat of the most important image-based CAPTCHA schemes between the years 2013 and 2018. Several image-based CAPTCHA schemes, such as reCAPTCHA V2 scheme, were attacked by ML [6]. Furthermore, approaches such as distortion, background noise mixing, and the use of adversarial instances were proposed as countermeasures against deep learning models. Adversarial examples by Szegedy et al. [7] and others have been suggested to enhance its security against ML-based attacks [8–10]. However, Na et al. [11] suggested a CAPTCHA solver that uses incremental learning on a limited dataset to defeat adversarial CAPTCHAs. To deal with visually impaired users, researchers proposed audio-based CAPTCHAs in addition to text-based and image-based CAPTCHAs. However, language barriers and poor usability limit the effectiveness of these schemes. Furthermore, supervised learning and automated speech recognition (ASR) [12] show how these schemes might be exploited. Researchers began developing behavioral-based CAPTCHA schemes in the 2010s to create difficulties based on behavioral features. The first behavioral-based CAPTCHA was launched by Geetest in 2012, while Google released No CAPTCHA reCAPTCHA in 2014 and invisible CAPTCHA in 2015 and 2017. Bot attacks mimicking the user's behavioral pattern have been demonstrated to be vulnerable to these schemes [6]. Because of the serious privacy concerns, Cloudflare recently decided to discontinue the use of reCAPTCHA [13]. Finally, recent research directions use sensor data to create challenges that are difficult for automated bots to replicate. However, we must wait a sufficient amount of time before we can fully evaluate sensor-based CAPTCHAs.

1.2 CAPTCHA codes

CAPTCHA schemes vary and are constantly improved as a result of advancements in advanced technology, AI, and hacking techniques. Main CAPTCHA codes, shown in Fig. 1, are currently classified as cognitive/behavioral-based, video-based, audio-based, image-based, text-based, and others.

1.2.1 Text-based CAPTCHAs

These CAPTCHAs became increasingly applied over the years. In these methods, the text is warped and shown to a user as an image and the user must enter this text accurately before passing this test. The AI hardness assumption is that humans can easily read the warped text, but bots using optical character recognition (OCR) techniques find it difficult. The different renderings of the challenge's text can be classified into three subcategories: 2D, 3D, and animation. In Table 1, we list a detailed taxonomy of the typical text-based CAPTCHAs.

2D text-based CAPTCHAs Andrei Broder with his team at the DEC Systems Research Center invented the 2D text-based CAPTCHA scheme in 1997. A similar method was used by the AltaVista website to prevent bots from influencing the rank of sites on the search engine

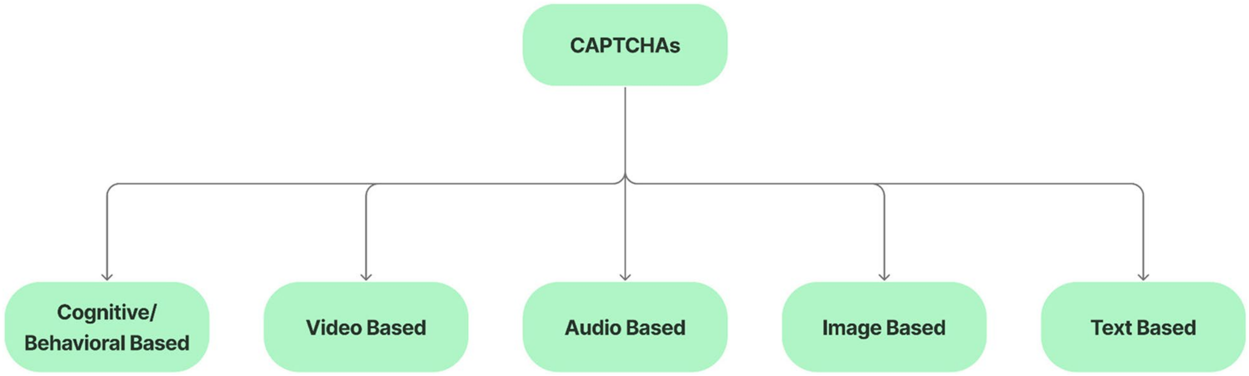
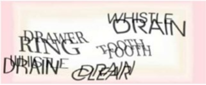
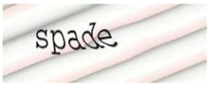
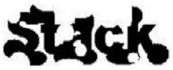
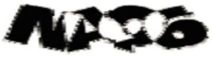
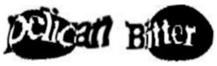
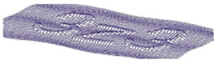
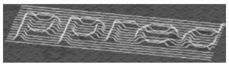
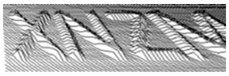
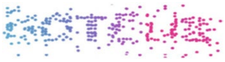
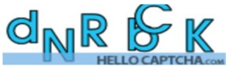
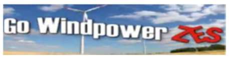
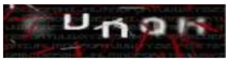


Fig. 1 Main schemes of CAPTCHA

Table 1 Typical text-based CAPTCHAs

CAPTCHA	Illustration	Challenge	Features	Type
Gimpy [14]		Recognize three words from a list of seven from a dictionary at random	Multiple strings, overlap, distortion, rotation, noise, background	2D text-based
EZ-Gimpy [14]		Recognize a single English word from a distorted image	One string, distortion, gradients, non-linear deformation, noise, background	
BaffleText [15]		Recognize a pronounced string of characters using difference masking	One string, distortion, difference masking	
Megaupload CAPTCHA		Recognize four characters who are overlapping and have negative intersection areas	Fixed length, distortion, overlap, negative intersection	
ReCAPTCHA V1 [16]		Recognize distorted text from old books that has been scanned	Two strings, distortion, noise, background, non-linear deformation	
Teabag3D [17]		Recognize a character sequence that appears on a grid in 3D space	Grids, protrusion, distortion, rotation, background and character blending	3D text-based
Super CAPTCHA [18]		Recognize a string of 3D characters	Lines, protrusion, distortion, rotation, background, character blending	
3DCAPTCHA [19]		Recognize a string of 3D characters	Texture, protrusion, distortion, background, character blending	
DotCHA [20]		To identify each letter, drag and rotate the model, then type the answer	Twisted form, anti segmentation by small spheres, rotation, human interaction	
HelloCAPTCHA [21]		Recognize a group of six characters in an animated GIF image	Multiple characters jumping, overlap, background, uppercase and lowercase mixing	Animated text-based
NuCaptcha [22]		Enter the final three red moving characters	Multiple characters jumping, rotation, coloring, adhesion, static and moving characters mixing, background	
Dracon CAPTCHA [23]		Recognize five characters that fade and blur at different points throughout the animation frames	Fade and blur effect, changing character locations, background, noise, coloring, interference lines	

[24]. Von Ahn and Blum created Gimpy CAPTCHA and EZ-Gimpy [14] in collaboration with Yahoo in 2000 to prevent bots from creating malicious advertisements and free accounts. Gimpy CAPTCHA requires you to correctly type at least three of seven random words in a dictionary. EZ-Gimpy is a condensed version of Gimpy only showing one word randomly in a dictionary. Generated word images use a variety of fonts, gradients, noise, and other effects to make them difficult for bots to recognize. Monica Chew and Henry Baird suggested BaffleText [15] in 2003, a text-based CAPTCHA using pronounceable pseudo-random words with masking algorithms to prevent recognition by OCR software. Megaupload.com created a segmentation-resistant CAPTCHA scheme in 2010. This method employs overlapping characters as well as the "Gestalt Perception" principle. According to the Gestalt perception principle, people can mentally reconstruct individual characters, whereas computers still struggle with this task. The first version of ReCAPTCHA [16] was designed to protect websites from computer attacks. If a user types correctly the known words from old books' two distorted words, they will pass the challenge. Chow et al. [25] proposed the concept of text-based clickable CAPTCHA. Their approach requests constructing a grid of clickable CAPTCHAs from multiple textual CAPTCHA challenges. The user must select the grid elements that correspond to the challenge requirement. Instead of using machine-printed text, the authors of [26, 27] proposed Handwritten CAPTCHAs to prevent recognition by OCR software.

3D text-based CAPTCHAs These CAPTCHA schemes take advantage of sequences of 3D character recognition by humans, but bots cannot, making them superior to 2D text-based CAPTCHAs. OCR Research Team [17] developed Teabag3D, a highly secure CAPTCHA. This CAPTCHA is composed of an image mixing textual characters with a 3D pattern. Super CAPTCHA [18] and 3DCAPTCHA [19] are text-based CAPTCHA schemes using the same assumptions as Teabag3D. Since 2013, Super CAPTCHA has been available as a WordPress.org plug-in. Imsamai and Phimoltares [28] developed the 3D CAPTCHA scheme, which involves showing 3D alphanumeric sequences and mixing many effects such as overlapping, rotation, noise, font variation, scaling, and other effects, to fool recognition of automated bots. Suzi et al. [20] recently suggested DotCHA, a 3D text-based CAPTCHA. 3D letters are made of small spheres in each challenge. Each letter is readable at a different twisted rotation angle around a horizontal axis. As a result, 3D text models need to be rotated several times to identify their letters.

Animated text-based CAPTCHAs These CAPTCHAs add a time dimension to text-based schemes. In detail, the textual content is animated in a short clip for each challenge, making the extraction more difficult for automated bots. In 2006, Fischer and Herfet [29] proposed one of the first animated CAPTCHA proposals. The concept of this CAPTCHA is to project text onto an animated deforming surface. Naumann et al. [30] developed an animated CAPTCHA with the idea of the human ocular system perception in 2009. Only when the letters move, users can tell the difference between the text and the background. With the same concept, Cui et al. [31] introduced an animated CAPTCHA that only recognizes correct characters on moving. The "zero-knowledge per frame" principle is applied to ensure no information leaks in each frame. In 2010, Creo Group released the animated HelloCAPTCHA [21]. For each challenge, a sequence of six characters is presented in a GIF image with some effects: random positions, various orientations, and others. The information is aligned to spread over multiple frames to prevent recognition over a single frame. The challenge in NuCaptcha [22] begins with a video of moving white font text, followed by three red characters in a dynamic background. To pass the challenge, the red characters must be typed correctly by the user. In Dracon CAPTCHAs [23], five characters are displayed in fixed locations that have been randomly changed with effects of fade, blur, and noise.

1.2.2 Image-based CAPTCHAs

Due to the recent failure of almost text-based CAPTCHAs, there is growing worry about their protection strength and accessibility. Lately, more designs are focusing on image-based instead of character recognition with the assumption of the general vision challenges being harder than text recognition. Table 2 contains a detailed categorization of the most commonly used image-based CAPTCHAs.

Interactive-based CAPTCHAs These CAPTCHAs are based on the user's interaction, such as swiping gestures or mouse movement, to reveal hidden points in an image. Conti et al. [32] suggested CAPTCHaStar in which the ability of humans to recognize shapes in a cluttered environment is used. The CAPTCHaStar challenge is made up of white pixels called stars that are randomly mixed together. The position of these stars changes depending on where the cursor is. Users must drag the cursor so that the stars form an understandable shape before clicking the left mouse button to pass the CAPTCHA test. Okada et al. [33] created Noise CAPTCHA with the same concept. This CAPTCHA is made up of two

Table 2 Typical image-based CAPTCHAs

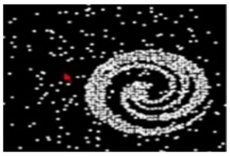
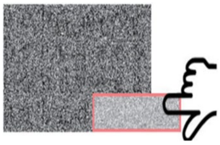
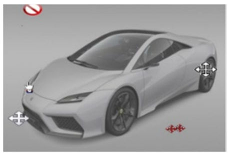
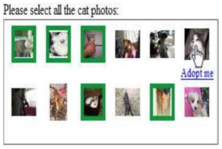
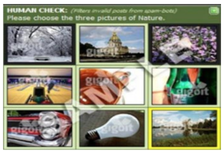
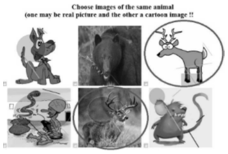
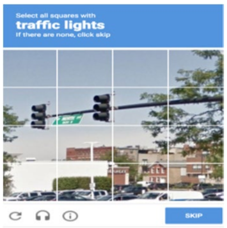
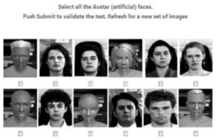
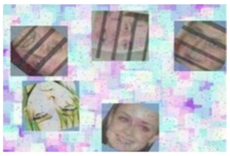
CAPTCHA	Illustration	Challenge	Feature	Type
CAPTCHaStar [32]		Move the cursor until a recognizable shape is formed	White pixels, noise, background, shape changing in term of moving cursor's location	Interactive-based
Noise CAPTCHA [33]		Move a small noisy image on top of a large noisy image until a hidden message or object appears	Noise, background, shape changing in term of moving cursor's location	
Cursor CAPTCHA [34]		Overlap the cursor on the target object in a randomly generated image	Background, noise, random location of target	
Asirra [35]	 Please select all the cat photos: Adopt me	Choose a cat from a collection of 12 images of cats and dogs	Grids, categorization of cats and dogs, location API integration (hence, poisoned database attacks)	Selection-based
HumanAuth CAPTCHA [36]	 HUMAN CHECK: *(Filters invalid posts from spam-bots)* Please choose the three pictures of Nature.	Choose images that have natural content	Limited image database, grids, masking images with logo	
SEMAGE CAPTCHA [37]	 Choose images of the same animal (one may be real picture and the other a cartoon image !!	Choose images that are semantically related from a set of images	Grids, limited image database, semantic linking	
No captcha reCAPTCHA [38]	 Select all squares with **traffic lights** If there are none, click skip SKIP	Choose all images that contain a specific object	Grids, object recognition, user activity tracking	
Avatar CAPTCHA [39]	 Select all the Avatar (artificial) faces. Push Submit to validate the test. Refresh for a new set of images	Choose an avatar face from a set of 12 images that include both human and avatar faces	Limited image database, grids, grayscale	
FaceDCAPTCHA [40]		Choose two images of the same person's face	Limited image database, noise, background, random image positions, rotation	

Table 2 (continued)

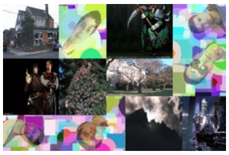
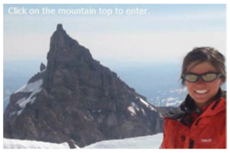
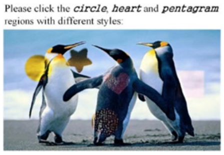
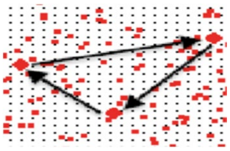
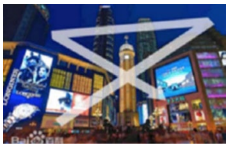
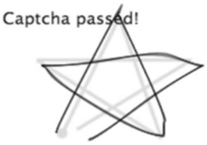
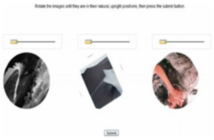
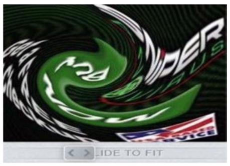
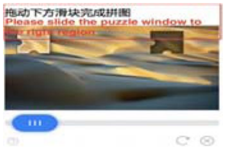
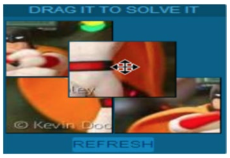
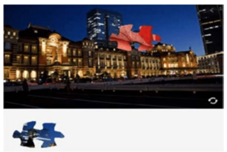
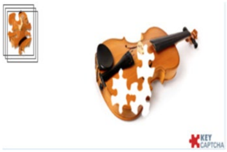
CAPTCHA	Illustration	Challenge	Feature	Type
FR-CAPTCHA [41]		Choose distorted real human faces from among nonhuman face images	Limited image database, noise, background, random image positions, rotation, distortion	
Implicit CAPTCHA [42]	 Click on the mountain top to enter.	Click on a specific area of an image	Limited image database, human craft, single target	Click-based
SACaptcha [43]	 Please click the *circle*, *heart* and *pentagram* regions with different styles:	Click on some of the image's regions that contain a specific shape mentioned in the challenge description	Limited image database, human craft, multi targets	
Drawing CAPTCHA [43]		Connect specific dots to one another	Noise, texture background, drawing	Draw-based
VAPTCHA [44]		Draw a similar trajectory to the reference trajectory	Noise, background, drawing patterns	
MotionCAPTCHA [45]	 Captcha passed!	Draw the shape shown in the box	Noise, background, drawing patterns	
WHAT's Up CAPTCHA [46]	 Rotate the images until they are in their natural, upright positions, then press the submit button. Submit	Slide the slider to the right to reorient at least three randomly rotated images	Three circle cells, limited image database, rotation	Slide-based
Minteye's Slide-to-Fit CAPTCHA [47]	 SLIDE TO FIT	Slide the slider until an undistorted image appears	Distortion, rotation, background, noise	
Tencent CAPTCHA	 拖动下方滑块完成拼图 Please slide the puzzle window to the right region	Move the slider such that two puzzle pieces match	Background, two puzzle pieces	

Table 2 (continued)

CAPTCHA	Illustration	Challenge	Feature	Type
Garb CAPTCHA [48]		To reconstruct the original image, drag and drop the puzzle pieces into their proper positions	Multi layers, background, noise, multi puzzle pieces, random positions	Drag and drop based
Capy CAPTCHA [49]		Drag a puzzle piece to finish a jigsaw puzzle	Multi layers, background, noise, one puzzle piece	
KeyCAPTCHA [50]		Drag three puzzle pieces to put the image together	Multi layers, background, noise, three puzzle pieces	

different-sized and noisy images, as well as a hidden object or message in one of the images. Users must drag the small noisy image to identify the hidden object in the large image before clicking the "submit" button to pass the CAPTCHA challenge. Cursor CAPTCHA, proposed by Thomas et al. [34], displays five cursors randomly in a generated image. To pass the challenge, users must overlap the mouse pointer onto a specific cursor.

Selection-based CAPTCHAs These CAPTCHAs require users to choose candidate images from a set of images. Only text or text with a sample image can be used to describe this task. Asirra [35] is a typical CAPTCHA of this scheme, in which users are asked to select all cats from a set of 12 images of dogs and cats. In HumanAuth CAPTCHA [36], users are required to pick up all images that contain natural content among natural content images (such as a tree or a river) with artificial content images (such as a car or a watch). SEMAGE (SEmantically MAtching imaGEs) CAPTCHA [37] differs from Asirra and HumanAuth CAPTCHA in that it requires users to select semantic images from an image set. As a result, the user must first recognize each image content and then identify the semantic relationship among them. Google released the "No captcha reCAPTCHA" [38] in 2014. Analyzing the browser environment (such as cookies and browser history), the system determines whether it is encountering a bot or not. The page will display only a checkbox or a selection-based CAPTCHA based on the risk level. The selection-based CAPTCHA challenge renders nine candidate images and a sample image describing the image's required content. In order to pass the challenge, the user must choose images that are similar to the sample. Facebook's image of CAPTCHA is similar to reCAPTCHA in its approach. To complete the challenge, users must choose images matching the hint description from a set of twelve images with varying content. Avatar CAPTCHA [39] asks users to select avatar faces from a set of 12 grayscale images that include both human and avatar faces. FR-CAPTCHA [41] and FaceDCAPTCHA [40] are two more face image CAPTCHAs. FR-CAPTCHA requires users to pick up the same person's two face images in a complex background. On the other hand, in FaceDCAPTCHA, users are required to choose between visually warped human face images and non-human face images.

Click-based CAPTCHAs These schemes display text and an image addressing where the user should click in order to pass the challenge. The main limitation of this type is that the challenge needs human intervention in order to generate a new instance. Implicit CAPTCHA [42] is a common example which requires users to click on an identical location of an image. Tang et al. [51] pioneered the use of SACaptcha in which the CAPCHA's some regions linking an explained specific shape must be clicked by users to pass the challenge.

Draw-based CAPTCHAs In 2006, Shirali-Shahreza, the first person, developed Drawing CAPTCHA [43], a drawing-based CAPTCHA. Diamond-shaped dots are connected by a user's drawing lines. The most difficult aspect is that users must identify these dots against a noisy background. VAPTCHA (Variation Analysis-Based

Public Turing Test to Tell Computers and Humans Apart) [44] consists of an image with a randomly generated trajectory in a challenge. To complete the challenge, users must draw a matching trajectory against this trajectory. In MotionCAPTCHA [45], similarly, users are also asked to draw a similar shape to the one rendered in the challenge box.

Slide-based CAPTCHAs In these CAPTCHAs, in order to solve a challenge, users must use a slider, such as dragging an image fragment to a correct location, rotating an image orientation or selecting a correct image form. WHAT's Up CAPTCHA [46] displays three rotated images randomly, and users must rotate the images to their correct position. Minteye's Slide-to-Fit CAPTCHA [47] displays a swirled image, and users must move the provided slider until they see the undistorted image version. Tencent CAPTCHA requires users to move the slider to match two puzzle pieces.

Drag and drop-based CAPTCHAs In these CAPTCHAs, users are required to align image pieces to form a complete image by dragging and dropping them. Garb CAPTCHA [48] displays four randomly shuffled pieces of an image. Users are required to reorder these image pieces to get the complete image to pass the CAPTCHA test. Hamid Ali et al. [52] pioneered the use of a puzzle-based CAPTCHA. Four image pieces of an image are required to be dragged and dropped into an empty four-cell grid to complete the challenge. Gao et al. [53] suggested a Jigsaw puzzle-based image-based CAPTCHA. In this CAPTCHA, an image is divided into many pieces (i.e., 9, 16, or 25) with only two wrongly positioned pieces. Users are required to swap the incorrect pieces to solve the challenge. Capy CAPTCHA [49] requires users to move a puzzle piece into a missing place in a challenge. This missing place is filled with a random image fraction. KeyCAPTCHA [50] displays three puzzle pieces and an incomplete image. Users are required to assemble these pieces to match the reference image. Once the cursor stays in the frame, the reference image will disappear. To pass the CAPTCHA challenge, users must move these pieces into the correct places.

1.2.3 Audio-based CAPTCHAs

For people with visual impairments, a suggested alternative to visual CAPTCHA schemes was audio-based CAPTCHA schemes. They must type what they have heard to pass the test. At Carnegie Mellon University, the researchers introduced audio reCAPTCHA, acquired by Google later. To solve the challenge, users are required to identify eight digits spoken in human noise and only accept one incorrect digit in these digits. The eBay Audio CAPTCHA is made up of six digits in various spoken noisy voices. Microsoft CAPTCHAs are made up of ten digits in different spoken voices mixing the noise of some conversations. Yahoo CAPTCHA requires users to enter seven digits after three child-spoken beeps with background noise. The 2013 version of Audio reCAPTCHA requires users to recognize all of the digits divided into three clusters in the challenge. Three or four overlapping digits are found in each cluster. The new version of reCAPTCHA in 2017 included ten spoken digits and background noise. In Table 3, we list the most popular audio-based CAPTCHAs.

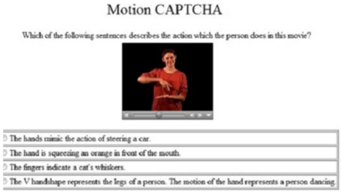
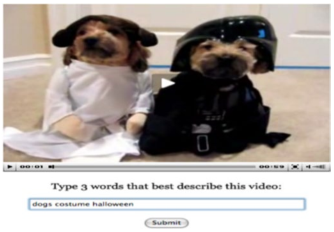
1.2.4 Video-based CAPTCHAs

In the challenge, a short video is created, reflecting a certain content, users are required to understand and describe it by text. In Table 4, we list some typical video-based CAPTCHAs. Kluever et al. [54] suggested a CAPTCHA in which with a short video, users are required to watch and then type three words to describe it. Shirali-Shahreza et al. proposed Motion CAPTCHA [55] which requires users to describe the motion of the person in their watching video by choosing one of the sentences.

Table 3 Typical audio-based CAPTCHAs

CAPTCHA	Challenge	Feature
Google Audio reCAPTCHA	- Version 2008: Recognize eight spoken digits against a background of human voices speaking backwards at varying volumes - Version 2013: Identify all of the digits in the challenge that are divided into three clusters, each of which contains three or four overlapping digits - Version 2017: Recognize ten spoken digits in the presence of background noise	Background, noise, cluster, overlap
eBay Audio CAPTCHA	Recognize six digits spoken in various voices with background noise	Six digits, background, noise
Yahoo Audio CAPTCHA	Recognize seven digits that appear after three beeps made by a child against a background of other children's voices	Seven digits, background, noise
Microsoft Audio CAPTCHA	Recognize ten digits spoken in different voices over a regular background noise of several concurrent conversations	Ten digits, different voices, background

Table 4 Typical video-based CAPTCHAs

CAPTCHA	Illustration	Challenge	Feature
Motion CAPTCHA [55]	 Motion CAPTCHA Which of the following sentences describes the action which the person does in this movie? ○ The hands mimic the action of steering a car. ○ The hand is squeezing an orange in front of the mouth. ○ The fingers indicate a cat's whiskers. ○ The V handshape represents the legs of a person. The motion of the hand represents a person dancing.	Choose the sentence that best describes the person's movement in the video	Semantic, limited video database, human craft
Kluever el al [54]	 Type 3 words that best describe this video: dogs costume halloween Submit	After watching a video, provide three words that best describe it	Semantic, limited video database, human craft

1.2.5 Cognitive-based CAPTCHAs

CAPTCHA methods based on cognitive abilities that provide increased security have largely replaced traditional Captcha methods. Cognitive abilities are brain-based skills that are the result of a distinct combination of neurobiological and psychological techniques. Knowledge, concentration, memory, judgment and assessment, reasoning and computation, problem-solving, and decision making are all aspects of human cognition and behavior. To distinguish between humans and bots, these CAPTCHA methods use biometric (something you are), physical (something you have), and knowledge-based (something you know) factors with or without the support of sensors like gyroscope or accelerometer [56, 57]. In Table 5, we list the most common cognitive-based CAPTCHAs. In 2020, Acien et al. [58] suggested BeCAPTCHA-Mouse that distinguishes humans from bots by analyzing mouse trajectories during the challenge. Gametrics [59] differentiates between humans and bots by collecting and analyzing the user's mouse movements during the operations of drag and drop to solve a dynamic cognitive game. GEETest and Netease [6], like Tencent CAPTCHA, require users to complete a sliding image-based CAPTCHA by moving the slider until two puzzle pieces are matched. If users complete the challenge and their sliding behavior is not suspicious, they are considered to have passed the challenge. Siripitakchai et al. [60] proposed EYE-CAPTCHA in which users are required to solve a math-based CAPTCHA by moving their eyes. To complete the challenge, the user must identify the correct answer and use his eyes to move the answer to the center of the screen. In 2014, Google launched "No CAPTCHA reCAPTCHA" (reCAPTCHA V2). All that is required is to check the "I'm not a robot" box. However, user behaviors (such as click, mouse moving, and other behaviors) along with other information (browser, cookies, history etc.) are collected and analyzed in the background. If users are suspected of being bots, they need to complete a second image-based reCAPTCHA. In 2017, Invisible reCAPTCHA, an upgraded version of reCAPTCHA V2 was released. The evaluation process is initiated in the background by triggering a JavaScript API call or by users clicking on an existing button. Invisible reCAPTCHA, like the "No CAPTCHA reCAPTCHA" approach, requires a second image-based reCAPTCHA challenge if users are suspected of being bots. In 2015, Guerar et al. [61], the first person, introduced the physical CAPTCHA for mobile devices, called CAPPCHA (Completely Automated Public Physical test to tell Computers and Humans Apart). Users must tilt the device to a specific degree, which is difficult for bots to do. Hupperich et al. [62] introduced Sensor CAPTCHA in 2016, in which users are required to perform a complex gesture (such as fishing, hammering, drinking) with their mobile devices. The authors of [63] proposed Pedometric CAPTCHA, in which humans are required to walk at least five steps. When the user walks, an acceleration is generated in the mobile device, making it difficult for bots. Mantri et al. [64] suggested a CAPTCHA scheme in which users must meet the requirement of moving the device in accordance with a specific guide showing on the device. Frank et al. [65] instructed users to perform a detectable gesture and recognized by the gyroscope (such as rotating, tilting, or drawing), on moving the device. Guerar et al. [66] developed Invisible CAPPCHA, which is similar to CAPPCHA in that the challenge is invisible to users. Reading sensors detect user taps as opposed to touchscreen

Table 5 Typical cognitive-based CAPTCHAs

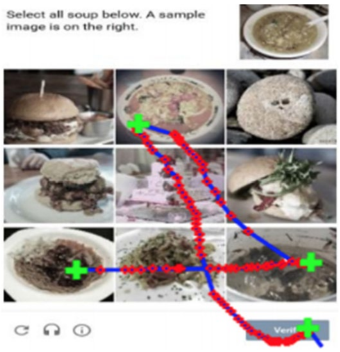
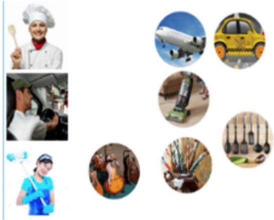
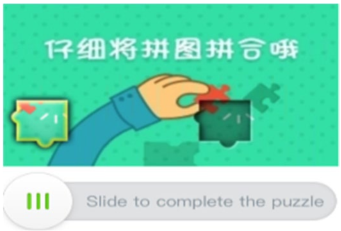
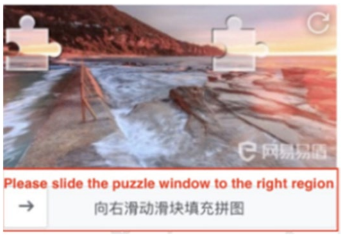
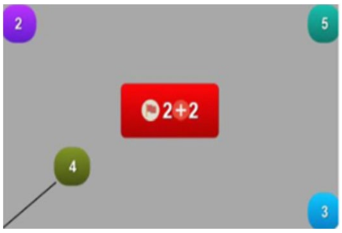
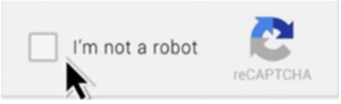
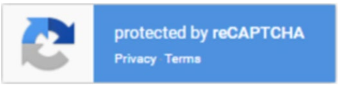
CAPTCHA	Illustration	Challenge	Feature
BeCAPTCHA-Mouse [58]	 Select all soup below. A sample image is on the right. 	Select all images in term of description	Grids, user mouse tracking, image selection, semantic
Gametrics [59]		Drag a subset of the moving objects to their corresponding static targets	Limited image database, human craft, semantic, image drag
GEETest	 仔细将拼图拼合哦 Slide to complete the puzzle	Slide the slider until two puzzle pieces match	Multi layers, background, noise, one puzzle piece and one missing puzzle piece, image slide
Netease [6]	 Please slide the puzzle window to the right region 向右滑动滑块填充拼图 	Slide the slider until two puzzle pieces match	Multi layers, background, noise, one puzzle piece and one missing puzzle piece, image slide
EYE-CAPTCHA [60]	 	The user finds the answer to a simple math operation displayed on the screen and moves it to the center with his eyes	Moving objects by eye, colorful, background, noise, math-based
No CAPTCHA reCAPTCHA	 □ I'm not a robot reCAPTCHA	Click on I'm not a robot Checkbox	User tracking
Invisible reCAPTCHA	 protected by reCAPTCHA Privacy · Terms	There is no visible challenge; it is triggered by a Javascript API or by the user clicking on an existing button on the website	User tracking

Table 5 (continued)

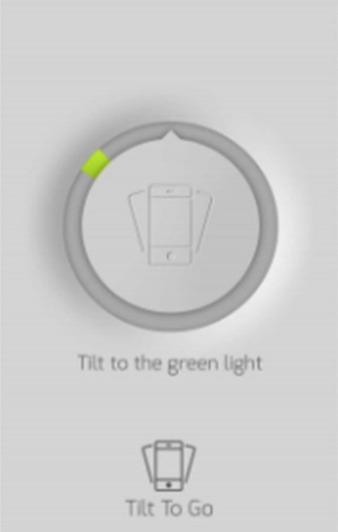
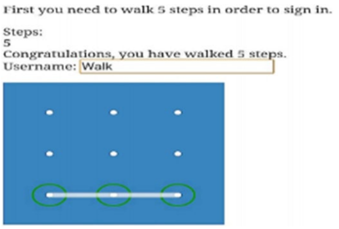
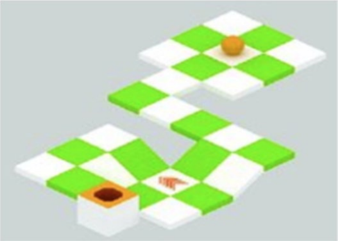
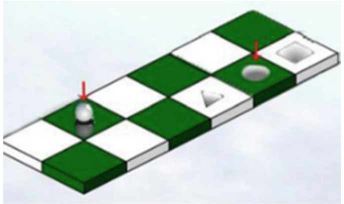
CAPTCHA	Illustration	Challenge	Feature
CAPPCHA [61]	 Tilt to the green light Tilt To Go	Tilt the device in a specific direction	Mobile devices, tilting devices
Sensor CAPTCHA [62]		Do gestures like hammering, fishing, and turning the body while holding the mobile device	Mobile devices, simulating gestures with devices
Pedometric CAPTCHA [63]	 First you need to walk 5 steps in order to sign in. Steps: 5 Congratulations, you have walked 5 steps. Username: Walk	Take at least five steps	Mobile devices, simulating gestures with devices
Invisible CAPPCHA [66]		No action is required	User tracking
AccCAPTCHA [67]		Play a simple rolling ball game or another popular game	Game, mobile devices
GISCHA [68]		Play a simple game in which you move a ball to a hole with a specific shape	Game, mobile devices

Table 5 (continued)

CAPTCHA	Illustration	Challenge	Feature
SenCAPTCHA [70]		Determine the animal's eye position, then tilt the device to move the ball there	Mobile devices, tilting devices
BrightPass [71]		In terms of screen brightness, enter a correct PIN digit or a deceptive lie digit	Screen brightness cognition

events, which bots can easily mimic. Furthermore, this CAPTCHA protects the user's privacy by not sending sensitive data to the server. AccCAPTCHA [67] requires a user to play the rolling ball game. To complete the game, the user must control the ball using the device's motion sensors. GISCHA, a mobile device game-based CAPTCHA, was proposed by Yang et al. [68]. To pass the challenge, a user must move the ball to the correct hole. Ababtain et al. [69] suggested the CAPTCHA which requires users to pass a simple game using sensors. They proposed five games, each with several static and one moving object. Users must move the moving object to hit the correct target static objects in order to pass the challenge. SenCAPTCHA was proposed by Feng et al. [70] for locating an animal facial key point. Users are shown a small red ball and an animal image. Then, they must control the red ball into the animal's eye center by tilting their devices. The authors [71] proposed BrightPass, a mobile authentication CAPTCHA to protect PIN/password. Their proposed mechanism uses screen brightness, which automated bots cannot detect, to determine when users should enter a correct digit or a deceptive digit. In the form of physical CAPTCHA, the authors [72, 73] proposed a PIN-based authentication CAPTCHA used for smartwatches. This mechanism is based on the same concept as CAPPCHA [74]. To enter the password, the bezel must be physically rotated to a specific degree. Similarly, the authors [75] use the digital crown rotation in smartwatches to protect the PIN code.

1.2.6 Other types

Stefan Popoveniuc [76] proposed the SpeakUP authentication method for remote unsupervised voting in 2010. Voice biometrics is enhanced with text-based CAPTCHA. Voters must read out loud a voted candidate's characteristics, rendered by 2D text-based CAPTCHA. Furthermore, voters' voice biometric characteristics are identified through a challenge. The author also suggested recording the voter's video of solving challenges. For protecting systems of facial authentication, Uzun et al. [77] suggested rtCaptcha, a Real-Time CAPTCHA. Users must record their out loud pronunciation of the presented 2D text CAPTCHA.

2 CAPTCHA attack analysis

CAPTCHA has developed into the most popular utilized standard security measure for preventing automated computer program attacks. In recent years, many attack methods, developed by hackers or researchers, have effectively cracked all common conventional schemes. Some methods, including Invisible reCAPTCHA, have not yet been broken. However, with the introduction of fourth-generation bots accurately mimicking human behavior, a secure CAPTCHA would be hardly designed without additional special devices. Specially, almost all cognitive-based CAPTCHAs with sensor support have not yet been vulnerable to automated attacks. However, they are still compromised to human-assisted relay attacks due to having a limited number of challenges and can be only solved using trusted devices. Table 6 lists various recent CAPTCHA attack techniques, with DNN/CNN and ML attack techniques dominating the list.

2.1 Attack against text-based CAPTCHA

Text-based CAPTCHAs were the first CAPTCHA scheme and still remain the most popular. Mori and Malik [78] introduced an attack method of shape matching in 2003 to pass Gimpy and EZ-Gimpy CAPTCHAs with an accuracy of 33% and 92%, respectively. The proposed method [93] used a correlation algorithm and

Table 6 Comparison of some recent CAPTCHA attacks

CAPTCHA	Attack method	Success rate	Type
Gimpy, EZ-Gimpy	Shape context matching [78]	33%, 92%	Text-based
Megaupload CAPTCHA	Segmentation [79]	78%	
ReCAPTCHA	Neural networks [80]	99.8%	
Teabag3D, 3DCAPTCHA, Super CAPTCHA	Pixel extraction [19]	31%, 58%, 27%	
HelloCAPTCHA	PDM (Pixel Delay Map)/CL (Catching Line) [81]	16% - 100%	
NuCaptcha	Box shape analysis & SIFT algorithm [82]	90%	
Asirra	SVM (support vector machine) [83]	82.7%	Image-based
HumanAuth	Side-channel attack [84]	92%	
Google image-based CAPTCHA	Deep learning/CNN [85]	70.78%	
Facebook image-based CAPTCHA	Deep learning/CNN [85]	83.5%	
reCAPTCHA V2	Deep learning/CNN [6]	79–88%	
Facebook image CAPTCHA	Deep learning/CNN [6]	86%	
China Railway CAPTCHA	Deep learning/CNN [6]	90%	
Avatar CAPTCHA	CNN [19]	99%	
FR-CAPTCHA	SVM [86]	23%	
FaceDCAPTCHA	SVM [86]	48%	
Minteye CAPTCHA	Sobel operators [87]	100%	
Tencent CAPTCHA	Deep learning/CNN [6]	100%	
Capy CAPTCHA, KeyCAPTCHA, Garb CAPTCHA	JPEG image continuity measurement [88]	65.1%, 20%, 98.1%	
CAPTCHaStar	Max concentration [89]	96%	
Audio reCAPTCHA	SVM [64]	45–58%	Audio-based
eBay audio CAPTCHAs	DFT (Discrete Fourier Transform) and supervised learning algorithm [65]	75%	
Microsoft and Yahoo audio	Non-continuous speech [66]	49%, 45%	
Audio reCAPTCHA	HMMs (Hidden Markov Models) [90], free online speech-to-text services, and minimal phonetic mapping [91]	52%, 85.15%	
GeeTest, Netease CAPTCHA	Sigmoid function [6]	96%, 98%	Cognitive-based
No CAPTCHA reCAPTCHA	Reinforcement learning [92]	96–97%	

a direct distortion estimation algorithm to successfully break EZ-Gimpy with a success rate of 99%. Chellapilla et al. [94, 95] created a highly secure CAPTCHA of anti-segmentation in 2005 after passing various text-based CAPTCHAs with machine learning. In 2008, several anti-segmentation CAPTCHAs, used by Google, Microsoft, and Yahoo, were demonstrated to be able to be cracked by El Ahmad and Yan [96, 97]. Later, other researchers attempted to pass these CAPTCHAs with higher success rates [98, 99]. El Ahmad and Yan [79] also broke Megaupload CAPTCHA with 78% of success. Google researchers [80] used neural networks to break the hardest category of ReCAPTCHA in 2014, with an accuracy of 99.8%. The authors [19] suggested 3D CAPTCHA attack methods without OCR software. In several 3D-based CAPTCHAs, such as 3DCAPTCHA, Teabag 3D, and Super CAPTCHA, they extracted pixels from the characters for automated challenge recognition. Using such a technique, the authors were able to break 3DCAPTCHA, Teabag 3D, and Super CAPTCHA with success rates of 58%, 31%, and 27%, respectively. Furthermore, the same authors [100] were able to pass Teabag 3D by using the 3D textual objects' side surface information. In the animated-based CAPTCHAs, Nguyen et al. [81] demonstrated how to easily extract information across multiple animated frames by using CL (Catching Line) or PDM (Pixel Delay Map). These methods successfully defeated animated CAPTCHAs such as KillBot Professional, iCAPTCHA, Dracon CAPTCHA, and Atlantis. Due to their vulnerability to segmentation attacks, the same methods were used in [81] to defeat HelloCAPTCHA variants with a success rate ranging from 16 to 100%. NuCaptcha is a segmentation-resistant animated CAPTCHA that works by overlapping and cramming together to counter PDM or CL attack methods. Elie Bursztein [82] separated objects in each frame with a success rate of 90% using an interest points (SIFT algorithm) density evaluation and bounding box shape analysis.

2.2 Attack against image-based CAPTCHA

Golle [83] was successful in breaking the Asirra scheme. To accomplish this, SVM (support vector machine) was used to classify cats and dogs with a success rate of 82.7%. Hernandez-Castro et al. [84] suggested a side-channel attack breaking HumanAuth with an accuracy rate of 92%. Facebook image-based CAPTCHA and Google image-based CAPTCHA were bypassed by Sivakorn et al. [85] with success rates of 83.5% and 70.78%, respectively. The authors [6] achieved success rates of 79 and 88% with the new and old variations of reCAPTCHA V2. They also defeated China Railway CAPTCHA and Facebook image CAPTCHA with success rates of 90% and 86%, respectively. Besides, these authors broke different image-based CAPTCHA schemes, including the Tencent CAPTCHA with a success rate of 100%. Convolutional Neural Networks (CNN) [19] was applied to successfully break Avatar CAPTCHA, with a success rate of 99%. Both FaceDCAPTCHA and FR-CAPTCHA were defeated by Gao et al. [86] with success rates of 48% and 23%, respectively. Minteye CAPTCHA was defeated in [87] by utilizing the length of the image's edges and Sobel operators. The attack method chooses the image with the smallest sum of edges based on the fact that a swirled image takes the longer edges. Hernandez-Castro et al. [88] suggested a low-cost attack using JPEG to measure image continuity. Using this side-channel attack, they successfully broke Capy CAPTCHA, Garb CAPTCHA, and KeyCAPTCHA with success rates of 65.1%, 98.1%, and 20%, respectively. Gougeon and Lacharme [89] were recently able to defeat CAPTCHAaStar with a success rate of 96%. They also demonstrated that the parameter tuning does not prevent this CAPTCHA from their attack on pixel concentration (stars) during image formation.

2.3 Attack against Audio-based CAPTCHA

Tam et al. [101] experimented with an SVM-based approach to defeat audio reCAPTCHA with a success rate of 45% for the exact matching solution and a success rate of 58% for a "one mistake" passing condition. Decaptcha by Burzstein and Bethard [102] demonstrated a success rate of 75% in bypassing eBay's audio CAPTCHAs. Their method analyzes the wave file using a Discrete Fourier Transform (DFT) and then clusters the energy spikes. Then, to recognize speech patterns, a supervised learning algorithm is employed to train audio data. The authors [103] introduced a CAPTCHA breaker with a non-continuous speech that broke Yahoo and Microsoft audio CAPTCHAs with success rates of 45% and 49%, respectively. The classification stage in this solver was supervised, whereas the automated segmentation stage was unsupervised. Amazon Mechanical Turk was used to label them, and the scraped CAPTCHAs were classified using the regularized least-squares classification (RLSC) algorithm. Due to the presence of semantic vocal noise, their system could only solve reCAPTCHA with a success rate of 1.5%. Sano et al. [90] suggested a CAPTCHA breaker for continuous speech to defeat anti-segmentation CAPTCHAs that overlap target voices. For speech recognition, Hidden Markov Models (HMMs) were employed and tested on the 2013 version of audio reCAPTCHA with a success rate of 52%. Bock et al. [91] presented unCaptcha that can bypass the 2017 version of audio reCAPTCHA with a success rate of 85.15% by utilizing free online services of speech-to-text and performing a minimal phonetic mapping for accuracy improvement.

2.4 Attack against cognitive-based CAPTCHA

Using four simulation functions (Softmax, Sigmoid, Tanh, and ReLu) to mimic human behaviors, Zhao et al. [6] successfully bypassed sliding-based CAPTCHA such as GeeTest and Netease CAPTCHA with success rates of 96 and 98%, respectively. By creating a tracking cookie for automated bots, Sivakorn et al. [85] were able to fool Google's risk analysis system. As a result, after 9 days of automated bots browsing various Google services, the solver can check the box of "I'm not a robot." Besides, the authors suggested a simple attack with a success rate of 70.78% for defeating the second reCAPTCHA V2 challenge. To break No CAPTCHA reCAPTCHA, the authors [92] applied the "divide and conquer" strategy. They were successful 97.4% of the time on a 100 × 100 grid and 96.7% of the time on a 1000 × 1000 screen resolution.

2.5 Attack against Other CAPTCHAs

Kluever et al. [54] developed a tag frequency-based approach to attack their proposed video-based CAPTCHA with a success rate of 13%. Hernandez-Castro et al. [104] were successful in breaking QRBGS CAPTCHA by the side-channel attack with a success rate of 44.54%. Mohamed et al. [105] demonstrated that dictionary-based attacks are able to defeat DCG CAPTCHAs. Moreover, developers [106, 107] proposed a solver that automatically bypasses SweetCAPTCHA, various slider CAPTCHAs (Taobao scheme) by developing a simple JavaScript code and puppeteer.

2.6 Other attacks

2.6.1 Side-channel attack

Side-channel attacks are processes that attempt to solve an issue that is considerably easier than the original. The intended solution is built around a difficult challenge (AI-hard problem), whereas the actual solution is built around any design or implementation issues to avoid the more difficult approach. These attacks rely on randomness deviations, missing uniform randomness, to identify a link between the challenges and their responses. In this case, the challenge provides (unintentionally, "leaked" or "side-channel") knowledge on the answer. ASIRRA's side-channel attacks are briefly described in this section [108]. ASIRRA is made up of over 25.000 photos, half of which are classed as cats or dogs. These photographs were processed by a classifier that, without utilizing any image recognition techniques, was able to discriminate between cat and dog pictures with about an accuracy of 60%. HumanAuth's authors opted to mix a PNG image with a random JPG image picked from the library to prevent easy image library indexing. Choosing a new watermark that has a greater impact on the original image would come at the expense of human usability.

2.6.2 Feature-based attack

In 2009, Philippe Golle [109] introduced the effective attacks on ASIRRA based on analyzing the CAPTCHA's features, such as font, shape, texture, and color. By employing image processing, this approach divides the photographs into a cell grid of texture and color (grayscale), which is then fed into support-vector machine (SVM) classifiers with a classification success of 83%.

2.6.3 Database-based attack

If a CAPTCHA is based on a public knowledge database (i.e., labeled photos), there are numerous potential attacks against that database:

- Database indexing attacks: the database can be downloaded (at least partially) to obtain the information needed to solve the CAPTCHA.
- Database poisoning attacks: with an open and unprotected CAPTCHA database, our information can be uploaded to help us solve the CAPTCHA with this information.

2.6.4 Human solving attack

CAPTCHAs are intended to be completed by humans, but there exist markets for labor services solving CAPTCHAs [110] (usually in cheap labor regions) and relay attacks, which transmit CAPTCHA challenges to humans who benefit from solving them [111].

3 CAPTCHA problem analysis

3.1 Attack threats

With the evolution of automated attacks, the differences in solving CAPTCHAs between humans and automated bots may become irrelevant: Should a human who is browsing another website or is presented with another program's GUI be ineligible to solve our CAPTCHAs? Is a computer program that has been human-assisted still an automatic attack? Because it is difficult to distinguish between humans and bots, CAPTCHA schemes require additional mechanisms to improve their security:

- Measure a "human" quality, ability, or behavior to distinguish between humans and computers.
- Differentiate between humans and human-assisted algorithms to prevent magnifying or human-assisted attacks.
- Prevent relay attacks by differentiating between humans who see the CAPTCHA on the original CAPTCHA site and those who see it on another site/interface [111].
- Prevent human farm attacks by employing methods to thwart or make more difficult the use of farms of solvers in solving the CAPTCHA.

3.2 AI hardness not transmitted

The majority of CAPTCHAs have been vulnerable as a result of one of the following issues:

1. They are based on a much more specific and weaker underlying problem than the original one intended.
2. Flaws from design or implementation make them much easier to be bypassed by employing procedures analyzing their challenges. As a result, these procedures are known as side-channel attacks because they attempt to solve a much easier problem than the one intended by the CAPTCHA designers [104, 108].
3. The difficulty of an AI-unsolved problem is hard to convey to a CAPTCHA design. We do not know how to categorize or deeply understand an AI hardness, so a CAPTCHA challenge of this AI hardness may be not difficult enough for automated bots.

3.3 Design flaws

3.3.1 Biased answer distribution

One common mistake is to select a non-uniformly distributed subset of possible answers. QRBGS (MathCAPTCHA) is one such example, with its designers employing one-digit figures in their arithmetic operations. As a result, the answers are likely to be small integers. Megaupload CAPTCHA is another example, which avoids using the values O, I, J, and 0. Worse, it always employs the three-letter-then-a-digit scheme, which makes it more user-friendly while also making it significantly less powerful. Teabag's challenges [112] use only three-character lengths and avoid characters that are hard to distinguish in 3D projections. Characters "S," "Z," "3," "P," "b," "w," "M," "t," and "d" appeared more than 3% in a sample of 100 challenges, while a major set of other 34 characters, including "1" and "0," did not appear (possibly to avoid coincidence with "I" and "O").

3.3.2 *Biased challenge distribution*

Any biased idea in CAPTCHA design that is not based on randomness can allow challenge analysis, leading to side-channel attacks or challenge categorization analysis. Because the distribution of letter sizes in Teabag is not uniform, the frontal borders of the characters can be chosen based on their area size. There is also pixel correlation, which allows for back-border detection. Simple algorithms, such as pixel continuity, can detect growing background areas. In some challenges, the non-character image portion can be removed completely or nearly completely [112]. Another example is the Megaupload CAPTCHA, which always prints the letters and digits in the same font style, Antique Olive (as identified by Identifont). Characters are rotated at specific angles, clockwise or counter-clockwise, with the first letter clockwise and the second counter-clockwise. It also prevents the overlap of more than two characters [113].

3.3.3 *Correlation between challenge and answer*

The challenge may provide (unintentionally, "leaked" or "side-channel") information based on the answer content. Side-channel attacks can be used to bypass the challenges by leveraging the leaked information.

3.3.4 *Evaluation of the answer*

It is not always necessary to make it easy for a CAPTCHA to determine whether or not their answers are correct. Avoid knowing whether an answer to a challenge is correct or incorrect, or any other way of knowing if it is close to being correct, if at all possible. We can communicate this information to the user via an intermediary communication mechanism (such as email accounts, which must also be controlled to limit emailing times) or we can transfer it to the user such that it is hard to be distinguished from automated bots.

3.3.5 *User dependence*

In general, making CAPTCHA dependent on the challenger is a bad idea, and it is even worse if this dependence can be known or guessed. ASIRRA, for example, displays pets in Petfinder that are near the challenger's position in order to increase the chances of adoption for the pets displayed in the CAPTCHA (using IP geolocation). This flaw is critical because it facilitates many types of attacks, including database poisoning and database indexing.

3.4 Implementation flaws

Some CAPTCHA systems can be completely bypassed by leveraging the session ID of a previously used CAPTCHA [114]. That is due to poor implementation, but it was not unusual a few years ago. Some developers still encode the answer to the challenge in the URL or a form field. Using this mistake, many challenges can be requested with the same answer. As a result, a mean attack [115] can be launched by calculating the median values of those challenges. Another mistake in implementation is sending the client a hash of the answer, such as an MD5 hash, as a key. If the number of answers is limited or not distributed uniformly, the hashes of these answers can be easily learned enough to solve the challenges. Besides, using small fixed pools of challenges is one of the common implementation flaws. HumanAuth, for example, uses fewer than a hundred images, even masking them with logos, that are easily characterized or indexed [108]. Furthermore, HumanAuth only generates challenge answers with values 0 or a small integer. This allows another type of attack: if the answer 0 fails, we will answer with a series of integers beginning with the smallest absolute values. Another common mistake is that QRBGS challenges, as an example, are not created on demand, but rather are repeated [104]. Furthermore, some systems employ an extremely risky communication method with the CAPTCHA server, which is easily exploitable [116].

3.5 Preserving users' privacy

In contrast to traditional CAPTCHA schemes, new sensor and behavioral-based CAPTCHA schemes have been shown to raise privacy concerns such as user behavioral data, cookies, and sensor data sent to remote servers. Some researchers proposed sending only the test results to the server, rather than the sensor data, as a solution. However, trusted hardware is required to prevent client-side hacking. As a result, the privacy of users should be strongly considered during the design phase of new CAPTCHA schemes.

3.6 Device compatibility

A robust and usable CAPTCHA is obviously expected to be compatible with a wide range of devices. The most promising CAPTCHA schemes, on the other hand, rely heavily on a single device. For example, CAPTCHA schemes based on touch-and-tap dynamics or mouse dynamics require device specialization. Sensor-based CAPTCHA schemes, which require sensors found only in smartwatches, tablets, or smartphones, are difficult to implement on the majority of users' devices.

4 How to design a good CAPTCHA

4.1 Good properties

Any new CAPTCHA design should be put into production in a test site, without other protections (to focus on the CAPTCHA's hardness), for a long enough period of time to allow research. These new CAPTCHAs should include the following features to improve security against automated bots:

1. In all parameters, there should be randomness and a uniform distribution. For example, for a text CAPTCHA: uniform number of areas, lines, pixels with random properties (color, group, group size, etc.), variable number of characters, various typefaces, image size, etc.
2. There should be no simpler CAPTCHA challenges: subtypes or alternatives should have the same level of difficulty (such as visual and audio CAPTCHAs).
3. The challenge should be as close to the original AI problem as possible.
4. The design should include features that detect automatic bypass or prevent relay attacks.
5. Challenges should be distributed uniformly and independent of users and answers. Furthermore, the answers should be distributed randomly and uniformly. There should be no statistical relationship between the challenges and the answers.
6. Make it difficult for automated bots to determine whether or not their answers are correct by using adversarial samples, response mechanisms, or communication methods with CAPTCHA servers.

4.2 Security assurance

1. Answer repetition: if an attacker is able to collect a finite quantity of challenges with the same answers, it must be confirmed that this attacker will not be able to create a better answer than a random answer. It means that there is no better attack than trial and error.
2. Challenge repetition: If our CAPTCHA has only a finite set of different challenges and we do not know how to solve them, there should be no better strategy than trial and error, with a low success rate.
3. Non-categorization: If our CAPTCHA is made up of different types of challenges, there should be no way to tell them apart automatically or to classify the difficulty of various challenges.

4.3 Security test

For this test, we propose to create a large enough set of elements (T = test, A = answer) of tests. We look for non-uniformities in this distribution using general randomness and statistical analysis tools [108]:

– Inconsistencies in the distribution of A (potential blind attack).
– Inconsistencies in the distribution of T (type-of-challenge categorization and challenge analysis).
– Correlations among T and A (potential side-channel attack).

These tests can be performed for some simple properties of T, such as color histograms, area sizes, histograms, distances between similar areas, maximum and minimum for a block of bytes, and bit correlation with given vectors. This can be used to estimate the security parameters of any CAPTCHA proposal, avoiding pitfalls such as irrelevant parameter values that cause leakage of information [104, 108, 117].

5 Conclusion

CAPTCHA is a competition between humans and computers. Computers attempt to mimic everything humans can do. On the contrary, Humans rely on AI's hardness and cognition capability to challenge computers. Obviously, with the rapid and continuous development of technology, computers outfitted with the most robust and cutting-edge software and hardware are capable of solving AI's most difficult problems at any time. In this paper, we have provided an overview analysis of HAI interactions between computer users and computers under current CAPTCHA schemes' the security aspects of open concerns, difficulties, and opportunities in CAPTCHA design. We expect that this work will serve as a good starting point for new CAPTCHA designers in order to avoid some common design and implementation flaws, as well as for the development of new security assessment and assurance level evaluation methodologies.

Acknowledgements
This work has been supported by the funds of the Polish Ministry of Education and Science assigned to AGH University of Science and Technology.

About the Authors

Ms.C. Nghia Dinh: Software architecture enthusiast and computer scientist. He has contributed to the success of many open sources and technology companies. In 2020, he received a Master of Science in Software Engineering from Bordeaux University, France. Currently, he is a Ph.D. candidate at Ostrava Technical University, Czech Republic.

Prof. Lidia Dominika Ogiela: Computer scientist, mathematician, and economist. She received a Master of Science in Mathematics and Master of Business Administration both in 2000. In 2005, she was awarded the title of Doctor of Computer Science and Engineering at the Faculty of Electrical, Automatic Control, Computer Science and Electronic Engineering of the AGH University of Science and Technology, for her thesis and research on cognitive informatics and its application in intelligent information systems. In 2016, she received Habilitation in Computer Science at the Faculty of Electrical Engineering and Computer Science at VŠB – Technical University of Ostrava in the Czech Republic. In 2018, she received the title of Doctor in Computer Science and Telecommunication at Hosei University, in Tokyo, Japan, for her thesis and research on human-centered computing for future-generation computer systems. She is an author of more than 230 scientific international publications on cognitive informatics, information systems, computational intelligence methods, and visual codes. She is a Lifetime Fellow Member of the prestigious international scientific society SPIE and a member of other societies: IEEE Senior Member, SIAM, ACM, OSA, CSS, and Information Processing Society of Japan. Currently, she is in a professor position at the Institute of Computer Science at AGH University of Science and Technology in Krakow, Poland. The author of recognized monographs in the field of cognitive informatics and IT systems and the author of cognitive approaches to knowledge extraction and data analysis.

Authors' contributions

ND: conceptualization, funding acquisition, investigation, writing, and draft of the final paper. LO: methodology, conception, and correction of the final paper. The author(s) read and approved the final manuscript. The author(s) read and approved the final manuscript.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]VSB Technical University of Ostrava, 17. listopadu 15/2172, 708-33 Ostrava-Poruba, Czech Republic. [2]AGH University of Science and Technology, 30 Mickiewicza Ave, 30-059 Kraków, Poland.

Received: 13 June 2022 Accepted: 8 November 2022
Published online: 12 December 2022

References

1. L. von Ahn, M. Blum, J. Langford, *CAPTCHA: using hard AI problems for security* (2003)
2. M Naor. Verification of human in the loop or Identification via Turing Test, http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.ps.
3. US Patent no. 6195698. Method for selectively restricting access to computer systems, http://www.freepatentsonline.com/6195698.html.
4. E. Bursztein, M. Martin, J. Mitchell, in *Proceedings of the 18th ACM Conference on Computer and Communications Security, Chicago, Illinois, USA*. Text-based CAPTCHA strengths and weaknesses (2011)
5. I.J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, V.D. Shet, *Multi-digit number recognition from street view imagery using deep convolutional neural networks. CoRR abs/1312.6082* (2014)
6. B. Zhao, H. Weng, S. Ji, J. Chen, T. Wang, Q. He, R. Beyah, in *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, Toronto, Canada*. Towards evaluating the security of real-world deployed image CAPTCHAs (Association for Computing Machinery, New York, 2018), pp. 85–96
7. S. Ch, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.J. Goodfellow, R. Fergus, in *In the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16*. Intriguing properties of neural networks (2014)
8. D. Hitaj, B. Hitaj, S. Jajodia, L.V. Mancini, Capture the bot: using adversarial examples to improve CAPTCHA robustness to bot attacks. IEEE Intell. Syst. **36**(5), 104–112 (2020). https://doi.org/10.1109/mis.2020.3036156
9. M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, D. Pérez-Cabo, No bot expects the deep-CAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation. IEEE Trans. Inf. Forens. Security **12**, 2640–2653 (2017)
10. C. Shi, X. Xu, S. Ji, B. Kai, J. Chen, R. Beyah, T. Wang, *Adversarial CAPTCHAs. arXiv:1901.01107 [cs.CR]* (2019)
11. D. Na, N. Park, S. Ji, J. Kim, in *Information Security Applications, Ilsun You*. CAPTCHAs are still in danger: an efficient scheme to bypass adversarial CAPTCHAs (Springer International Publishing, Cham, 2020), pp. 31–44
12. M. Jain, R. Tripathi, I. Bhansali, P. Kumar, in *The 21st International ACM SIGACCESS Conference on Computers and Accessibility (Pittsburgh, PA, USA) (ASSETS '19)*. Automatic generation and evaluation of usable and secure audio ReCAPTCHA (Association for computing machinery, New York, 2019), pp. 355–366
13. Sergi Isasi Matthew Prince. Moving from reCAPTCHA to hCaptcha. (2020) https://blog.cloudflare.com/moving-from-recaptcha-to-hcaptcha.
14. Luis von Ahn, Manuel Blum, Nick Hopper, John Langford and Udi Manber. GIMPY.
15. M. Chew, H.S. Baird, in *Document Recognition and Retrieval X*, ed. by T. Kanungo, E. H. Barney Smith, J. Hu, P. B. Kantor. BaffleText: a human interactive proof, vol 5010 (International Society for Optics and Photonics, SPIE, 2003), pp. 305–316
16. L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, reCAPTCHA: human-based character recognition via web security measures. Science **321**(5895), 1465–1468 (2008)
17. OCR Research Team, Teabag 3D evolution. (2006). https://ocr-research.org.ua.
18. M.L. Wells, *Exciting features in super CAPTCHA* (2003)
19. V.D. Nguyen, Y.-W. Chow, W. Susilo, *On the security of text-based 3D CAPTCHAs* (2014)
20. S. Kim, S. Choi, in *DotCHA: a 3D text-based scatter-type CAPTCHA*, ed. by W. Engineering, M. Bakaev, F. Frasincar, I.-Y. Ko. (Springer International Publishing, Cham, 2019), pp. 238–252
21. Program Product, HelloCAPTCHA. (2010), http://www.hellocaptcha.com.
22. NuCaptcha Inc, NuCaptcha, (2018), https://www.nucaptcha.com.
23. Dracon Visual Flash CAPTCHA, (2006), https://www.dracon.biz/captcha.php.
24. M. Tariq Banday, N.A. Shah, *A study of CAPTCHAs for Securing Web Services* (2011)
25. R. Chow, P. Golle, M. Jakobsson, L. Wang, X.F. Wang, in *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications (Napa Valley, California) (HotMobile '08)*. Making CAPTCHAs clickable (Association for Computing Machinery, New York, 2008), pp. 91–94
26. A. Rusu, V. Govindaraju, in *Ninth International Workshop on Frontiers in Handwriting Recognition*. Handwritten CAPTCHA: using the difference in the abilities of humans and machines in reading handwritten words (2004), pp. 226–231
27. A. Rusu, V. Govindaraju, in *Human Interactive Proofs*, ed. by H. S. Baird, D. P. Lopresti. Visual CAPTCHA with handwritten image analysis (Springer, Berlin Heidelberg, 2005), pp. 42–52
28. M. Imsamai, S. Phimoltares, in *International Conference on Information Science and Applications*. 3D CAPTCHA: a next generation of the CAPTCHA (2010), pp. 1–8
29. I. Fischer, T. Herfet, in *IEEE Workshop on Multimedia Signal Processing*. Visual CAPTCHAs for document authentication (2006), pp. 471–474
30. A.B. Naumann, T. Franke, C. Bauckhage, in *Human-Computer Interaction – INTERACT 2009*, ed. by T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. O. Prates, M. Winckler. Investigating CAPTCHAs based on visual phenomena (Springer, Berlin Heidelberg, 2009), pp. 745–748
31. J. Cui, J. Mei, X. Wang, D. Zhang, W. Zhang, in *International Conference on Multimedia Information Networking and Security*. A CAPTCHA implementation based on 3D animation, vol 2 (2009), pp. 179–182
32. M. Conti, C. Guarisco, R. Spolaor, in *Applied Cryptography and Network Security*, ed. by M. Manulis, A.-R. Sadeghi, S. Schneider. CAPTCHaStar! A novel CAPTCHA based on interactive shape discovery (Springer International Publishing, Cham, 2016), pp. 611–628
33. M. Okada, S. Matsuyama, in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. New CAPTCHA for smartphones and tablet PCs (2012), pp. 34–35

34. V.A. Thomas, K. Kaur, in *Tenth International Conference on Wireless and Optical Communications Networks (WOCN)*. Cursor CAPTCHA — implementing CAPTCHA using mouse cursor (2013), pp. 1–5
35. J. Elson, J.R. Douceur, J. Howell, J. Saul, in *Proceedings of the 14th ACM Conference on Computer and Communications Security (Alexandria, Virginia, USA) (CCS '07)*. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization (Association for Computing Machinery, New York, 2007), pp. 366–374
36. Neo. Blog post, [HumanAuth] Verification code for natural patterns, (2006).
37. S. Vikram, Y. Fan, G. Guofei, in *Proceedings of the 27th Annual Computer Security Applications Conference (Orlando, Florida, USA) (ACSAC '11)*. SEMAGE: a new image-based two-factor CAPTCHA (Association for Computing Machinery, New York, 2011), pp. 237–246
38. V. Shet, *Are you a robot? Introducing "No CAPTCHA reCAPTCHA"* (2014)
39. D. D'Souza, P.C. Polina, R.V. Yampolskiy, in *IEEE International Conference on Electro/Information Technology*. Avatar CAPTCHA: telling computers and humans apart via face classification (2012), pp. 1–6
40. G. Goswami, B. Powell, M. Vatsa, R. Singh, A. Noore, FaceDCAPTCHA: Face detection-based color image CAPTCHA. Fut. Generat. Comput. Syst. **31**, 59–68 (2014)
41. G. Goswami, B.M. Powell, M. Vatsa, R. Singh, A. Noore, FR-CAPTCHA: CAPTCHA based on recognizing human faces. PLoS One **9** (2014)
42. H.S. Baird, J.L. Bentley, in *Document Recognition and Retrieval XII*, ed. by E. H. Barney Smith, K. Taghva. Implicit CAPTCHAs, vol 5676 (International Society for Optics and Photonics, SPIE, 2005), pp. 191–196
43. M. Shirali-Shahreza, S. Shirali-Shahreza, in *28th International Conference on Information Technology Interfaces*. Drawing CAPTCHA (2006), pp. 475–480
44. J.C. Yuan, *Variation analysis-based public turing test to tell computers and humans apart* (2018)
45. MotionCAPTCHA v0.2, Stop spam, Draw Shapes, (2011).
46. R. Gossweiler, M. Kamvar, S. Baluja, in *Proceedings of the 18th International Conference on World Wide Web (Madrid, Spain) (WWW '09)*. What's up CAPTCHA? A CAPTCHA based on image orientation (Association for Computing Machinery, New York, 2009), pp. 841–850
47. Blog post, Minteye offers no-type CAPTCHA as a security twist, (2012).
48. Garb CAPTCHA, (2013).
49. C. Inc, *Capy Puzzle CAPTCHA* (2018)
50. KeyCAPTCHA, (2010).
51. M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang, P. Wang, Research on deep learning techniques in breaking text-based captchas and designing image-based Captcha. IEEE Trans. Inf. Forens. Security **13**, 2522–2537 (2018)
52. F.A.B. Hamid Ali, F.B. Karim, in *International Conference on Computer, Communications, and Control Technology (I4CT)*. Development of the CAPTCHA system based on puzzles (2014), pp. 426–428
53. H. Gao, D. Yao, H. Liu, X. Liu, L. Wang, in *13th IEEE International Conference on Computational Science and Engineering*. A novel image based CAPTCHA using jigsaw puzzle (2010), pp. 351–356
54. K.A. Kluever, R. Zanibbi, in *Proceedings of the 5th Symposium on Usable Privacy and Security (Mountain View, California, USA) (SOUPS '09)*. Balancing usability and security in a video CAPTCHA (Association for Computing Machinery, New York, 2009), pp. Article 14–Article 11
55. M. Shirali-Shahreza, S. Shirali-Shahreza, in *Conference on Human System Interactions*. Motion CAPTCHA (2008), pp. 1042–1044
56. N. Krzyworzeka, L. Ogiela, M.R. Ogiela, Cognitive based authentication protocol for distributed data and web technologies, article number 7265. Sensors **21**(21) (2021). https://doi.org/10.3390/s21217265
57. M.R. Ogiela, N. Krzyworzeka, L. Ogiela, Application of knowledge-based cognitive CAPTCHA in cloud of things security. Concurr. Comput. Pract. Exp. **30**(21), article number e4769 (2018). https://doi.org/10.1002/cpe.4769
58. A. Acien, A. Morales, J. Fiérrez, R. Vera-Rodriguez, *BeCAPTCHA-mouse: synthetic mouse trajectories and improved bot detection* (2020)
59. M. Mohamed, N. Saxena, in *Proceedings of the 32nd Annual Conference on Computer Security Applications*. Gametrics: towards attack-resilient behavioral authentication with simple cognitive games (2016)
60. A. Siripitakchai, S. Phimoltares, A. Mahaweerawat, in *3rd IEEE International Conference on Computer and Communications (ICCC)*. EYE-CAPTCHA: an enhanced CAPTCHA using eye movement (2017), pp. 2120–2126
61. M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, B. Messabih, in *International Conference on High Performance Computing Simulation (HPCS)*. A completely automatic public physical test to tell computers and humans apart: a way to enhance authentication schemes in mobile devices (2015), pp. 203–210
62. T. Hupperich, K. Krombholz, T. Holz, in *Trust and Trustworthy Computing*, ed. by M. Franz, P. Papadimitratos. Sensor Captchas: on the usability of instrumenting hardware sensors to prove liveliness (Springer International Publishing, Cham, 2016), pp. 40–59
63. S. Kulkarni, H.S. Fadewar, in *2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. Pedometric CAPTCHA for mobile Internet users (2017), pp. 600–604
64. V.C. Mantri, P. Mehrotra, *User authentication based on physical movement information* (2018)
65. B.Z. Frank, J.A. Latone, *Verifying a user utilizing gyroscopic movement* (2018)
66. M. Guerar, A. Merlo, M. Migliardi, F. Palmieri, Invisible CAPPCHA: a usable mechanism to distinguish between malware and humans on the mobile IoT. Comput. Secur. **78**, 255–266 (2018)
67. C.-J. Liao, C.-J. Yang, J.-T. Yang, H.-Y. Hsu, J.-W. Liu, in *Proceedings of EdMedia & Innovate Learning*, ed. by J. Herrington, A. Couros, V. Irvine. A game and accelerometer-based CAPTCHA scheme for mobile learning system (Association for the Advancement of Computing in Education (AACE), Victoria, 2013), pp. 1385–1390
68. T.-I. Yang, C.-S. Koong, C.-C. Tseng, Game-based image semantic CAPTCHA on handset devices. Multimed. Tools Appl. **74**, 5141–5156 (2013)
69. E. Ababtain, D. Engels, in *International Conference on Computational Science and Computational Intelligence (CSCI)*. Gestures based CAPTCHAs the use of sensor readings to solve CAPTCHA challenge on smartphones (2019), pp. 113–119
70. Y. Feng, Q. Cao, H. Qi, S. Ruoti, in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*. SenCAPTCHA: a mobile-first CAPTCHA using orientation sensors, vol 4 (2020), pp. 1–26
71. M. Guerar, M. Migliardi, A. Merlo, M. Benmohammed, F. Palmieri, A. Castiglione, Using screen brightness to improve security in mobile social network access. IEEE Trans. Dependable Secure Comput. **15**(4), 621–632 (2018)
72. M. Guerar, M. Migliardi, F. Palmieri, L. Verderame, A. Merlo, Securing PIN-based authentication in smartwatches with just two gestures. Concurr. Comput.: Pract. Exp. **32**, 18 (2020)
73. M. Guerar, L. Verderame, M. Migliardi, A. Merlo, in *IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. 2GesturePIN: securing PIN-based authentication on smartwatches (2019), pp. 327–333
74. M. Guerar, A. Merlo, M. Migliardi, Completely automated public physical test to tell computers and humans apart: a usability study on mobile devices. Fut. Generat. Comput. Syst. **82**, 617–630 (2018)
75. M. Guerar, L. Verderame, A. Merlo, F. Palmieri, M. Migliardi, L. Vallerini, CirclePIN: a novel authentication mechanism for smartwatches to prevent unauthorized access to IoT devices. ACM Trans. Cyber-Phys. Syst. **4, 3**, Article 34, 19 (2020)
76. S. Popoveniuc, in *Industrial Track ACNS*. SpeakUp: remote unsupervised voting (2010)
77. E. Uzun, S.P.H. Chung, I. Essa, W. Lee, in *NDSS*. rtCaptcha: a real-time CAPTCHA based liveness detection system (2018)
78. G. Mori, J. Malik, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Recognizing objects in adversarial clutter: breaking a visual CAPTCHA (2003)
79. A.S. El Ahmad, J. Yan, L. Marshall, in *Proceedings of the Third European Workshop on System Security (Paris, France) (EUROSEC '10)*. The robustness of a new CAPTCHA (Association for Computing Machinery, New York, 2010), pp. 36–41
80. Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay D. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. CoRR abs/1312.6082 (2014).
81. V.D. Nguyen, Y.-W. Chow, W. Susilo, in *Cryptology and Network Security*, ed. by J. Pieprzyk, A.-R. Sadeghi, M. Manulis. Attacking Animated CAPTCHAs via Character Extraction (Springer, Berlin Heidelberg, 2012), pp. 98–113
82. E. Bursztein, *How we broke the nucaptcha video scheme and what we propose to fix it* (2012)
83. P. Golle, in *Proceedings of the 15th ACM Conference on Computer and Communications Security (Alexandria, Virginia, USA) (CCS '08)*. Machine

learning attacks against the Asirra CAPTCHA (Association for Computing Machinery, New York, 2008), pp. 535–542

84. C.J. Hernandez-Castro, A. Ribagorda, Y. Saez, in *2010 International Conference on Security and Cryptography (SECRYPT)*. Side-channel attack on the HumanAuth CAPTCHA (2010), pp. 1–7
85. Suphannee Sivakorn, Jason Polakis, and Angelos D. Keromytis. I'm not a human: breaking the Google reCAPTCHA. In BlackHat, (2016).
86. H. Gao, L. Lei, X. Zhou, J. Li, X. Liu, in *IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications, Dependable Autonomic and Secure Computing, Pervasive Intelligence and Computing*. The robustness of face-based CAPTCHAs (2015), pp. 2248–2255
87. Jack. Breaking the MintEye image CAPTCHA in 23 lines of Python, (2013).
88. C.J. Hernández-Castro, M.D.R. Moreno, D.F. Barrero, Using JPEG to measure image continuity and break capy and other puzzle CAPTCHAs. IEEE Internet Comput. **19**, 46–53 (2015)
89. T. Gougeon, P. Lacharme, in *ICISSP*. How to break CaptchaStar (2018)
90. S. Sano, T. Otsuka, H.G. Okuno, in *Advances in Information and Computer Security*, ed. by K. Sakiyama, M. Terada. Solving Google's continuous audio CAPTCHA with HMM-based automatic speech recognition (Springer, Berlin Heidelberg, 2013), pp. 36–52
91. K. Bock, D. Patel, G. Hughey, D. Levin, in *Proceedings of the 11th USENIX Conference on Offensive Technologies (Vancouver, BC, Canada) (WOOT'17)*. UnCaptcha: a low-resource defeat of recaptcha's audio challenge, vol 7 (USENIX Association, USA, 2017)
92. Ismail Akrout, Amal Feriani, and Mohamed Akrout. Hacking Google reCAPTCHA v3 using Reinforcement Learning. ArXiv abs/1903.01003 (2019).
93. G. Moy, N. Jones, C. Harkless, R. Potter, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. Distortion estimation techniques in solving visual CAPTCHAs, vol 2. II–II (2004)
94. K. Chellapilla, K. Larson, P. Simard, M. Czerwinski, in *In the 2nd Conference on Email and Anti-Spam*. Computers beat humans at single character recognition in reading based human interaction proofs (HIPs) (2005)
95. K. Chellapilla, K. Larson, P.Y. Simard, M. Czerwinski, in *Human Interactive Proofs*, ed. by H. S. Baird, D. P. Lopresti. Building segmentation based human-friendly human interaction proofs (HIPs) (Springer, Berlin Heidelberg, 2005), pp. 1–26
96. J. Yan, A. Salah, E. Ahmad, in *Proceedings of the 15th ACM Conference on Computer and Communications Security (Alexandria, Virginia, USA) (CCS '08)*. A low-cost attack on a Microsoft Captcha (Association for Computing Machinery, New York, 2008), pp. 543–554
97. J. Yan, A. Salah, E. Ahmad, *Is cheap labour behind the scene? - low-cost automated attacks on Yahoo CAPTCHAs. Technical Report* (School of Computing Science, Newcastle University, England, 2008)
98. O. Starostenko, C. Cruz-Perez, F. Uceda-Ponga, V. Alarcon-Aquino, Breaking text-based CAPTCHAs with variable word and character orientation. Pattern Recognit. **48**, 1101–1112 (2015)
99. Y. Zi, H. Gao, Z. Cheng, Y. Liu, An end-to-end attack on text CAPTCHAs. IEEE Trans. Inf. Forens. Security **15**, 753–766 (2020)
100. V.D. Nguyen, Y.-W. Chow, W. Susilo, in *Information Security and Cryptology - ICISC 2011*, ed. by H. Kim. Breaking a 3D-Based CAPTCHA Scheme (Springer, Berlin Heidelberg, 2012), pp. 391–405
101. J. Tam, S. Hyde, J. Simsa, L. Von Ahn, in *Proceedings of the 21st International Conference on Neural Information Processing Systems (Vancouver, British Columbia, Canada) (NIPS'08)*. Breaking audio CAPTCHAs (Curran Associates Inc, Red Hook, 2008), pp. 1625–1632
102. E. Bursztein, S. Bethard, in *Proceedings of the 3rd USENIX conference on Offensive technologies*. Decaptcha: breaking 75% of eBay audio CAPTCHAs, vol 1 (USENIX Association, 2009), p. 8
103. E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, J. Mitchell, in *2011 IEEE Symposium on Security and Privacy*. The failure of noise-based non-continuous audio captchas (2011), pp. 19–31
104. C.J. Hernandez-Castro, A. Ribagorda, Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study. Comput. Secur. **29**, 141–157 (2010)
105. M. Mohamed, N. Sachdeva, M. Georgescu, S. Gao, N. Saxena, C. Zhang, P. Kumaraguru, P.C. van Oorschot, W.-B. Chen, in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (Kyoto, Japan) (ASIA CCS '14)*. A three-way investigation of a game-CAPTCHA: automated attacks, relay attacks and usability (Association for Computing Machinery, New York, 2014), pp. 195–206
106. M. Swain, in *Encyclopedia of Systems Biology*, ed. by W. Dubitzky et al.. Knowledge-based system (2013)
107. F. Vitas, *How to bypass "slider CAPTCHA" with JS and Puppeteer* (2019)
108. C.J. Hernandez-Castro, A. Ribagorda, Y. Saez, *Side-channel attack on labeling CAPTCHAs* (2009)
109. P. Golle, in *ACM CCS*. Machine learning attacks against the Asirra CAPTCHA (2008)
110. D. Danchev, *Inside India's CAPTCHA solving economy* (2008)
111. TROJ CAPTCHAR. A Trojan horse to relay CAPTCHAs at Trend-Micro, http://blog.trendmicro.com/captcha-wish-your-girlfriend-was-hot-like-me/.
112. C.J. Hernandez-Castro, A. Ribagorda, *Analysis of the Teabag CAPTCHA version 1.2* (2010)
113. C.J. Hernandez-Castro, A. Ribagorda, *Preliminary analysis on the Megaupload CAPTCHA* (2010)
114. H. Yeen, *Breaking CAPTCHAs without using OCR* (2009)
115. W. Wieser, *Captcha recognition via averaging* (2007)
116. A. Caine, U. Hengartner, The AI hardness of CAPTCHAs does not imply Robust Network Security. IFIP, Trust. Manag. **238**, 367–382 (2007)
117. M.R. Ogiela, U. Ogiela, *Shadow Generation Protocol in Linguistic Threshold Schemes, CCIS - Communication in Computer and Information Science*, vol. 58, (Springer-Verlag, Berlin Heidelberg, 2009), pp.35–42

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.