African Journal of Advanced Pure and Applied Sciences (AJAPAS)

Online ISSN: 2957-644X

Volume 2, Issue 3, July-September 2023, Page No: 309-323

Website: *https://aaasjournals.com/index.php/ajapas/index*

||Arab Impact factor 2022: 0.87|| SJIFactor 2023: 5.689|| ISI 2022-2023: 0.557

A Security Analysis of Text-based Captcha Schemes

Abdalnaser Muhammad Algwil*

Computer Science Department, Faculty of Information Technology, Alasmarya Islamic University, Zliten, Libya

*Corresponding author: *a.algwil@it.asmarya.edu.ly*

Received: July 16, 2023	Accepted: September 02, 2023	Published: September 05, 2023

Abstract:

Captcha has become a standard security mechanism to protect many services and resources on the Web. A Captcha challenge is created and validated automatically by computer to distinguish whether the user's identity is human or an automated program. Thus, it should be easy to solve by humans and very difficult to solve by automated software. The majority of current Captcha schemes on the Internet are principally based on distorted text challenges. However, text-based Captchas usually have many shortcomings in terms of *security*, *usability*, or the balance between them. That is, to resist attacks from auto-recognition programs, the text in the image has to be distorted and camouflaged. However, too sophisticated distortion may also degrade the readability for humans. It is thus critical for a Captcha scheme to be well balanced between *usability* and *security*. In this paper, we discuss *security* aspects and various attacks on currently used text-based Captcha schemes. The discussion included the different types of Captcha attacks, followed by defensive and offensive techniques commonly used by Captcha designers and attackers, respectively, to achieve their various goals, as well as describing the various dedicated research efforts to break Captcha schemes, have been explored. At the end, this paper discusses a list of desirable properties that are preferred in any robust Captcha scheme. We expect this work will provide good aspects for Captcha developers to avoid many design flaws.

Keywords: Attack, Captcha, Security.

Cite this article as: A. M. Algwil, "A security analysis of text-based Captcha schemes," *African Journal of Advanced Pure and Applied Sciences (AJAPAS)*, vol. 2, no. 3, pp. 309–323, July-September 2023.

تحليل الأمان على الكابتشا النصية

عبد الناصر محمد الغويل*

قسم علوم الحاسوب، كلية تقنية المعلومات، الجامعة الأسمرية الإسلامية، زليتن، ليبيا

الملخص

أصبحت الكابتشا آلية أمان شائعة الاستخدام لحماية العديد من الخدمات والموارد على الويب. يتم إنشاء اختبار الكابتشا والتحقق من صحته تلقائيًا بواسطة الحاسوب لتمييز ما إذا كانت هوية المستخدم بشراً أو برنامجًا آليًا. حيث يجب أن يكون حل الكابتشا سهلًا بالنسبة للبشر وصعب الحل بواسطة البرامج الآلية. تعتمد غالبية أنظمة الكابتشا الحالية على الإنترنت بشكل أساسي على تحديات النص المشوه. ومع ذلك، عادةً ما يكون لدى الكابتشا النصية العديد من أوجه القصور من حيث الأمان أو قابلية الاستخدام أو التوازن فيما بينهما. فللمقاومة من هجمات التي تعتمد على برامج التعرف التلقائي، يجب تشويه النص الموجود في الصورة وتمويهه. غير أن إجراء التشويهات المعقدة للغاية على صورة الكابتشا قد يؤدي أيضًا

إلى صعوبة قراءتها من قبل البشر. ومن ثم، فمن الأهمية بمكان أن يكون نظام الكابتشا متوازنًا بين سهولة الاستخدام وقوة الأمان. في هذه الورقة، سنناقش الجوانب الأمنية والهجمات المختلفة على أنظمة الكابتشا النصية المستخدمة حاليًا. تضمنت المناقشة أنواعًا مختلفة من هجمات الكابتشا، متبوعة بأساليب دفاعية وهجومية يشيع استخدامها من قبل كل من مصممي أنظمة الكابتشا والمهاجمين لتحقيق أهدافهم المختلفة، بالإضافة إلى استكشاف مختلف الجهود البحثية التي تم تكريسها لكسر مخططات الكابتشا. في النهاية، تناقش هذه الورقة قائمة الخصائص المرغوبة التي يفضل أن تكون ضمن أي نظام كابتشا متين. نتوقع أن يوفر هذا العمل جوانب جيدة لمطوري الكابتشا لتجنب العديد من عيوب التصميم مستقبلاً.

الكلمات المفتاحية: الهجوم، الكابتشا، الأمان

Introduction

With the rapid progression of the internet, many free services such as a Web mail services, social networking, online voting, and so on have been developed and deployed so that they are accessible anytime from anywhere by anybody. In fact, such services are usually designed and intended only for human use. However, malicious automated programs, known as Web bots, have been designed and deployed in order to abuse such free online services. In fact, these bots typically have considerable capabilities to perform repetitive tasks automatically, pretending to be humans, and thus pose a serious threat to various services on the Internet where a human interaction should nominally be implicitly assumed. For instance, signing up for free webmail accounts (in order to later use for malicious purposes such as originating spam), automated posting to forums and blogs, manipulating online polls, unauthorized access to certain online resources and sending large volumes of traffic to a specific Web service to affect a Denial of Service (DoS) attack.

With the ever-increasing attacks on these services, Web service providers employ CAPTCHA, which is an acronym for ***C****ompletely* ***A****utomated* ***P****ublic* ***T****uring tests to tell* ***C****omputers and* ***H****umans* **A**part, to protect their services against adversarial attacks. A Captcha, as a challenge-response test, is created and validated automatically by a computer to distinguish whether a request is originating from a human or an automated program. Such a test should be easy for a human to solve, but almost impossible for current automated software.

Over the last two decades, many Captcha schemes have been widely used to filter out malicious interactions from computers. Almost all Captcha classes, including research proposals and real productions, have been intrinsically developed based on open or intractable problems in the Artificial Intelligence (AI) arena. While the AI community endeavors to invent machines that are capable of demonstrating human-level abilities, various current limitations allow humans to outperform machine abilities. In fact, these same limitations can actually be useful, and represent an advantage in other applications. That is, AI-unsolved problems have been used as usable mechanisms for security purposes (i.e., *Captcha*). Examples of such hard, open problems are commonly found in areas such as text recognition, image understanding and speech recognition.

The security of a Captcha is essentially based on the assumption that an attacker cannot solve an underlying AI problem with higher accuracy than that currently known to be the state-of-the-art in the AI community [1]. Accordingly, to break a Captcha scheme, an adversary must find a new algorithm to solve its underlying open AI problem. In fact, a beneficial side effect can be gained from such an approach by inducing security researchers, in addition to attackers, to advance the AI field. In such a case, the Captcha mechanism can be considered a win-win situation. That is, if the Captcha scheme remains unbroken, then it can be used as a security approach to distinguish human users from computer bots; otherwise, (i.e., if it is broken automatically by computers) a hard, open AI-problem has been solved, leading to a further progress in the field of Artificial Intelligence [2]. However, in practice, this is not often the case. That is, many Captcha schemes, which were based essentially on AI-hard problems, have already been broken as a result of design and implementation flaws.

Among all Captcha classes, text-based Captcha is the most widely used on the Internet. However, the research question is: *How robust is the text-based-Captcha in protecting Internet services?* To answer this question, a comprehensive security investigation of text-based Captcha is required.

Research Aims

This paper aims to provide:

- A comprehensive security analysis on text-based Captcha schemes.
- An in-depth explanation of the types of Captcha attacks.
- Defense and attack techniques.
- General desirable attributes of robust Captcha designs.

Research Methodology:
The search adopts the descriptive analytical approach, which is in line with the nature of the subject.
The remaining paper is organized as follows: First, a brief illustration of Captcha types. Second, an explanation of Captcha security evaluation. Third, a detailed description of the types of Captcha attacks. Fourth, an expanded presentation of the different mechanisms that used in defense and attack. Fifth, a review of several attacks that have been successfully launched against text-based Captcha schemes in their chronological order. Sixth, discussion of a list of general desirable attributes of Captcha designs. Finally, the conclusion of the paper.

Types of Captcha
Various types of Captcha have been proposed and developed, as based on existing ability gaps between humans and machines with regards to hard-to-solve AI problems. von Ahn et al. [2] pointed out that a problem is defined as "*hard*" according to the general consensus of the community working on it, which concluded that it is difficult to find automatic solutions with the current AI state-of-the-art. They also stated that not all hard problems in the AI area can be utilized to structure a Captcha, defining the properties required to construct an effective Captcha as follows:

- ***Full automation***. Captcha should be generated and validated automatically.
- ***Humanly solvable***. Humans can solve the challenge easily and quickly.
- ***Indiscrimination between human users***. All humans should be accepted with high reliability and without discrimination based on disabilities, age, and so on.
- ***Automatically unsolvable***. No computer programs should be able to solve the challenge with the current state-of-the-art technologies.
- ***Robustness for many years***. Unless there is an advance in the algorithmic state of the art, the underlying problem should be resistant to automatic programs for many years, even if the code and/or its data are publicly available.

In this light, some Captcha designs have been developed and used in real-world services, whereas others have merely been sketched out at the idea stage. Generally speaking, the various types of Captcha can be classified under five main categories according to the challenge content presented to the user as follows:

1. Text-based Captcha
This category of Captchas is the most widely used, in which the challenge appears as an image of distorted text and users are required to recognize and correctly retype the text. Many kinds of distortions, transformations, and complicated colour combinations are usually used in a Captcha image with the aim of making the test more challenging. Over the last two decades, many text-based Captcha schemes have been developed using various anti-segmentation mechanisms, however, most of them have been broken with high success rates. The *Crowding Character Together* (CCT) is considered the most secure anti-segmentation technique that is widely applied in Captcha designs as shown in figure 1. The CCT technique removes whitespace between characters in order to connect characters with each other, which make it very difficult for automated programs to locate individual characters.

(a) Google (b) Microsoft (c) eBay

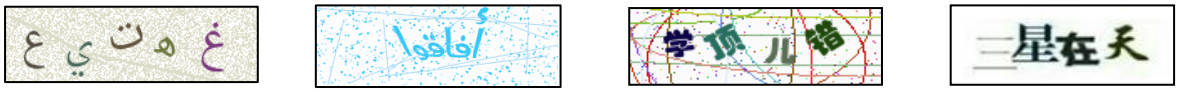
Figure 1 Examples of CCT-based Captcha schemes

Although conventional text Captcha schemes have been compromised in many scenarios, still have many virtues, especially in terms of implementation and maintenance. Chellapilla et al. [3] demonstrated that the popularity of OCR Captcha tests compared to others is due to several reasons, for instance:

- Hard problems in the OCR field are well-known, as well as well-studied.
- OCR-based Captcha tests, with minimal or no instructions, can be easily understood and solved by humans who have been trained at character recognition tasks since childhood.
- Owing to the popularity of Roman characters and numbers, OCR Captcha tests can be used worldwide with minimal associated localization issues.
- Discovering all possible permutations of a variable-length string derived from Roman characters and numbers can yield a very large search space of solutions that provide a strong security barrier against brute-force attacks.

- ORC Captcha tests can be produced very quickly with minimal consumption of computational resources as well as occupying a relatively small display area.

In the same vein, many non- Roman text-based Captcha schemes have been also designed in several languages such as Arabic [4, 5] and Chinese [6], as shown in figure 2. More details about text-based Captcha can be found in [7].



(a) Ar Captcha (b) Arabic Captcha (c) it168.com (d) Renren.com

Figure 2 Examples of Arabic and Chinese Captchas.

2. Image-based Captcha

This category principally exploits semantic interpretation ability gaps between humans and computers in the area of image recognition. That is, an image Captcha test is presented to the user , who is asked to recognize or classify some pictures or objects as different from others based on their characteristics. Recently, various types of image-based Captcha schemes have been developed to improve security aspects and usability levels across different devices, for instance reCaptcha from Google [8], PiSHi Captcha [9] and Style Matching Captcha[10], as shown in figure 3.

(a) Style Matching Captcha (b) PiSHi Captcha (c) reCaptcha

Figure 3 Examples of image-based Captchas.

It seems that image-based Captchas do not provide a high level of security as it is believed to be. Zhao et al. [11, 12] reveal that a variety of image-based Captcha classes are vulnerable to their attacks, which can break many popular image Captchas in the real world such as ReCaptcha, Facebook, Tencent SlidePuzzle, and Netease SlidePuzzle with high success rates.

Image-based Captcha is not as popular or in as widespread use as those based on text. This might be because of its limitations, which can be summarized as follows:

- Most such schemes are just proposals presented in the literature and have not been produced and tested extensively in the real world, in contrast to text-based Captcha schemes which are widely used and tested.
- To be secure, an image-based Captcha requires a large image database. However, human intervention (e.g., labelling or selecting images) is usually required to create such a large, labelled database of images. This contrasts with one of the principles of Captcha design, namely that the Captcha should be, by definition, fully automated.
- Some image-based Captcha designs are vulnerable to random guessing attacks (i.e., selecting a label/image(s) among a limited number of candidates). However, the tactic of increasing the number of rounds to reduce guessing success rates may also deteriorate the user-friendliness of the scheme.

- People who have colour blindness may face serious problems when solving Captcha challenges based on colorful images.
- Large picture sizes used in image-based Captcha implicitly require more significant resources with regards to web page area and network bandwidth.
- Problems with misspelling, synonyms, polysemy, and mislabelling are often associated with some categories of image-based Captcha schemes.
- The images used in Captcha may not be appropriate and consistent with the specific subject of a web site.

3. Sound-based Captcha

Sound-based Captcha exploits open Artificial Intelligence problems found in the area of audio recognition. That is, a sound clip, which usually includes a sequence of distorted characters, is listened to by the user, who is asked to recognize and retype them in the right order characters from that distorted clip. Sound-based Captcha is usually used as an alternative to visual-based Captchas for users with visual impairments. Hence, it can be found integrated along with visual Captcha schemes in websites. In fact, several designs of sound-based Captchas [13, 14] have been proposed in the literature and used in the real world for visually impaired users.

Generally speaking, Captcha schemes based on sound have numerous issues related to *usability*, *security*, and *accessibility*. That is, sound Captcha challenges are usually more time consuming and harder for humans than visual Captcha tests, especially if the users are non-native speakers. In fact, there have been a few usability studies conducted on audio Captcha schemes, such as [15, 16, 17], all of which have clearly demonstrated that users were only able to solve audio challenges with relatively low success rates. On the other hand, there were also a small number of investigations into the security of audio Captcha schemes compared to other Captcha categories, which in turn have shown that the current audio challenges can be vulnerable to automated attacks. Some of these studies are [18, 19, 20], which were able to break a wide range of available audio Captcha tests with high success rates. Accessibility is another issue related to audio Captcha designs; we will not only mention people who are both blind and deaf in this regard, but also the audio hardware and/or specific software that is typically required to play them.

4. Motion-based Captcha

The basic principle in this breed of Captcha schemes is to exploit the ability of the superiority of humans over automated software to recognize the semantic contents of a movie. In this Captcha, a motion scene involving an animated object or a moving text is shown to the user, who is asked to understand and detect the semantic items in the animated video challenge.

However, the use of video technology and animation to construct user-friendly Captcha does not mean that it is indecipherable by automated software because many such have already been successfully broken. As with image-based Captcha schemes, animated Captcha tests usually require a large database of labelled videos. From a usability viewpoint, animation file sizes, unless they are in GIF format, are relatively large, which in turn can adversely affect load time and bandwidth requirements. Besides, the solving time can take a relatively long time because the user might watch the entire video before attempting to solve the challenge. Furthermore, some types of animated video files cannot be seen unless a certain software suite (e.g., Flash Player) is installed beforehand.

5. Cognitive Captcha

Captcha challenges of this category need a high level of cognitive ability of the type that is very hard to imitate by computers for instance solving riddles, performing arithmetic operations, and semantic interpretation of texts. In other words, cognitive Captcha schemes typically require semantic interpretations of their challenges that rely on the semantics of the images, language constructs, or on solving riddles. Without a doubt, these cognitive functions are very difficult for computers to solve.

Linguistic Captcha [21][22] is an example of cognitive-based Captcha. in this breed of Captcha, a simple question for example: "*The list elephant, black, purple, shark and blue contains how many colours?*", is rendered to the user, who is required to type the answer. A significant advantage of these linguistic Captcha schemes is that people who are blind and/or deaf can gain access to websites that utilize Captcha tests in the text domain. Math Captcha [23], visual reasoning Captcha [24] and zxCaptcha [25] are other examples of cognitive Captcha

However, building a cognitive Captcha scheme usually encounters intractable problems in practice. Further, linguistic Captcha tests are always language dependent. Moreover, cognitive Captcha schemes based on games and puzzles are usually slower than their text-based equivalents, as well as requiring additional software to fulfil

their challenges. A further usability issue is that certain people with cognitive disabilities may have significant difficulties in solving this sort of Captcha.

Captcha Security Evaluation

In an optimal Captcha design, the generation and verification of Captcha challenges should be fully automated, without involving humans, in real time. On the other hand, solving these challenges should only be possible when manually performed by humans so that automated programs cannot solve them. This does not mean that a Captcha, to be considered secure, must be 100% resistant to computer attacks; however, a bot also should not be able to solve challenges at a success rate beyond a certain attack threshold. Although the attack threshold is considered the most straightforward evaluation metric by which to evaluate the security of a Captcha, there is currently no consensus on a specific success rate for attacks to be deemed as a standard security threshold. The most stringent threshold has been introduced by Chellapilla et al. [3], who consider that computers should not solve Captcha challenges with a success rate of more than 0.01%. Although this proportion is frequently referenced in the literature, Zhu et al. [26] have suggested a less rigorous measurement in which an automated program should not be more than 0.6% successful. Bursztein et al. [27] consider 0.01% as a security goal highly challenging, and deem a Captcha scheme broken when an automatic attack can reach a precision of at least 1%. While Chew and Tygar [28] determine this threshold according to the attacker's profitability in an economic sense. That is, they argue that a Captcha can still be secure as long as it raises the cost of an automated attack above that of using a human solver. Chew and Tygar [28] and Zhu et al. [26] point out that the response time, defined as the expected time for a human user to take to complete a Captcha, such that the human will pass and a computer will not, can be considered another metric by which to evaluate the Captcha efficacy. That is, if an attack responds within the time frame that human users respond to a challenge, this attack can be deemed effective; otherwise, it is deemed ineffective. This time frame has been determined as 30 seconds according to [29].

The above notwithstanding, it is extremely difficult and complex to measure the security of Captcha as there is no test tool that contains a vast repository of bots for evaluating Captcha schemes, especially given that Captcha security depends not only on the hard underlying AI problem, but also on other design and implementation considerations. Over the past few years, many types of attacks have emerged. That is, each time designers develop a new Captcha scheme, attackers attempt to find a mechanism to bypass it. In this way, the arms race between Captcha developers and aggressors escalates, and indeed may never end. To explore this issue in more depth, the following sections will highlight the types of Captcha attacks, defensive and aggressive approaches, and attacks on Captcha schemes.

Types of Captcha Attacks

Attacks on Captcha schemes can be generally classified into five categories as follows:

- ***Attacks based on solving the underlying AI problem***. In this category, attackers aim to solve a Captcha's underlying AI problem so that a previously unsolved problem in the field of AI that a Captcha utilizes is solved. In fact, this is the hardest kind of attack to undertake, yet is nevertheless the preferred method for Captcha designers. That is, the success of this type of attack fulfils the original design goal of the Captcha in being a win-win situation, since every step back in the Captcha security is really a step forward for the artificial intelligence field. However, this was not usually the case as many types of Captcha challenges have been bypassed in ways that the original problem remains open.

- ***Attacks based on mistakes in the design and implementation of Captcha challenges***. Here, attackers take advantage of existing security vulnerabilities in designing or implementing the Captcha challenges used to break the entire system using alternative methods to those intended by the designer. Although these attacks can pass the security provided by a particular Captcha, or even a spectrum of Captcha schemes, they do not solve a relevant AI problem, thus no relevant contribution is added to the field of artificial intelligence. In fact, most attacks launched on Captcha schemes fall under this category. A well-known example of this category is the pixel-count attack performed by Yan and El-Ahmad in [30] to break the Captcha challenges provided by the `Captchaservice.org` service. Another example is the mistakes found in the Microsoft Captcha design which were exploited by a simple attack [31] with a success rate of more than 60%.

- ***Side-channel attacks***. This kind of attack emerged as a result of the fact that Captcha security depends not only on the difficulty of the AI problem, but also on the implementation components that generate challenges, deliver them to users, and validate user responses. In this attack, attackers do not seek to solve the underlying AI problem or Captcha challenge but rather try to find pitfalls or flaws in the system implementation to bypass the Captcha challenge completely, and without really dealing with the challenge's content. Therefore, side-channel

attacks do not relate to the actual designs of the Captcha schemes, but instead target their implementations. API attacks on the CCaptcha service [32] is an example of this category of attacks. Algwil [33] illustrates how to securely model Captcha as a web service to avoid this type of attack.

➢ ***Relay attacks***. In this type of attacks, rather than using automated solving approaches to break Captcha tests, attackers' resort to rather 'lazy' methods of sidestepping Captcha challenges by outsourcing the task to remote human-solvers. Put simply, an attacker requires only a simple bot that forwards the Captcha image to the human solver, who then solves the challenge and provides the response; the attacker's bot then relays the corresponding response back to the original website [34]. Attackers may outsource the Captcha-solving process to one of the two categories of human solvers:

- ***Paid solvers***. An attacker can hire low-cost human labour to decipher Captcha challenges in exchange for a certain amount of money – practically speaking, just a few pennies – in return for doing this kind of work [35]. In reality, the human solvers are usually hired from sweatshops in developing countries, such as India and Bangladesh, where a pool of workers are willing to solve Captcha challenges with prices as low as $1 per thousand Captcha tests [36]. In fact, most human-based Captcha solving services, such as Antigate [37] and Imagetyperz [38], usually provide API packages in multiple programming languages so that an attacker can easily upload Captcha images and receive corresponding responses.
- ***Unwitting solvers (Deceived users).*** In this category, attackers could convince or entice unsuspecting human users to unknowingly solve the Captcha challenges for them. For instance, a high-traffic website, which is dominated by an attacker, may ask its visitors to opportunistically solve third-party Captcha challenges before being able to access some free service [36]. Although the attacker does not pay any money for a human solver, the latter can be given a reward for his/her service (e.g., acquiring a picture, free software, and so on).

In fact, third-party human attacks are difficult to avoid, as there is no a reliable approach that can be used to distinguish a human solver from a genuine user. Despite this, some Captcha schemes have been proposed to limit relay attacks, such as iCaptcha [39] and the emerging image game Captcha [40].

➢ ***Effortless attacks (Random guessing or Brute force attacks)***. This is the simplest type of attack and essentially free. Here, an attacker uses a trial-and-error approach by repeatedly attempting random solutions until one succeeds. This attack is effective against Captcha schemes that use a small search space, a few candidates in each challenge, and with no restrictions imposed on the number of attempts to solve a challenge. In the context of text-based Captcha schemes, if a Captcha test consists, for example, of a short string (say, three digits) derived from a small character set (i.e., 10 digits, 0-9), the probability of solving this challenge by blind guesswork is $(1/10)^3 = 0.10\%$, while a longer Captcha text (e.g., six letters) with a larger character set (i.e., 26 letters) can exponentially reduce the likelihood of success to $(1/26)^6 = \sim 0.00000032\%$. Random guessing attacks can be more practical for Captcha schemes based on image selection or those in the text domain which typically ask the user to select one, or a few, candidate(s) among a finite number of options. For this reason, most image-based Captcha tests require users to solve two challenges or more in a row, but this is indeed at the expense of Captcha usability.

As a concluding remark, all five types of attacks are of the same importance because attackers are always looking for the weakest link in the chain to breach the security of a Captcha. Thus, to prevent any of the above-mentioned attacks, Captcha developers should employ an appropriate hard AI problem in conjunction with other protection mechanisms such as the rate limiting strategy, timed lockout policy, and so forth. While Captcha designers typically utilize different defensive techniques to boost their Captcha schemes; attackers, on the other hand, employ whatever aggressive methods are required to overcome Captcha security. The following section provides further details about these defensive and aggressive approaches.

Defence vs. Attack

The arms race between Captcha experts and attackers is not only ongoing but is escalating to advanced levels; each strives to devise new mechanisms that boost their gains. Over the past two decades, numerous studies have been conducted on Captcha security for both offensive and defensive purposes. Owing to their vast popularity, text-based Captcha designs have received the overwhelming majority of attention in terms of such studies and attacks. Hence, this section sheds light on the strategies and attacks employed for text Captcha schemes.

The robustness of a text-based Captcha essentially relies on the strengths of the constituent segmentation and recognition problems. Although the recognition problem has been already solved [41], anti-segmentation strategies are only effective if the anti-recognition mechanisms are well-designed [27]. Therefore, strong text-based Captcha schemes should rely primarily on a combination of segmentation and recognition challenges to

enhance their security. The following sections present the defensive and offensive techniques commonly used by Captcha designers and attackers, respectively.

1. Defensive approaches

Captcha developers normally use two defensive approaches to the security of text-based Captcha schemes: the Anti-Segmentation approach and Anti-Recognition approach, each of which involves different techniques, as follows:

- ***Anti-Segmentation techniques.*** These techniques are used to prevent attackers from segmenting the text into its constituent characters. There are several mechanisms that are commonly used to resist segmentation; for example:
  - *The "arcs as clutter" mechanism*. In this method, the arcs used may, or may not, intersect with the text characters. In the case of an intersection, the arcs are typically drawn with random curvatures to cross certain characters of the Captcha text and bridge the space between adjacent characters, making character segmentation processes extremely difficult. It should be noted that the arcs' thicknesses, lengths and colours should be similar to the character segments to prevent the attacker from finding any discriminator that can be used to distinguish arcs from real characters. By these means, even non-intersected arcs can serve as fake characters. In fact, this mechanism was initially used by Microsoft's Captcha, which was later broken in [31].
  - *Character fragmentation mechanism*. Instead of connecting characters with each other, characters are horizontally and vertically cut into fragments and scattered in such a way that no automatic approach is capable of reassembling them into their original characters. ScatterType Captcha [42] applies this mechanism.
  - *Background confusion mechanisms*. These techniques attempt to conceal characters in a complex background. The segmentation resistance can be fulfilled in several ways; for example, using a complex image as a background, adding noise and clutter to the background, or using similar colours for both characters and background [27].
  - *The "Crowding Characters Together – CCT" or (Negative kerning) mechanism*. This is the most secure mechanism by which to deter segmentation algorithms and has been broadly adopted by many corporations, such as *Microsoft* and *Google*. In this technique, the spaces between characters are removed so that each character in the Captcha connects with two characters on both sides.
- ***Anti-Recognition techniques***. These are used to deteriorate the classifier accuracy and decrease scheme learnability. They must be used in conjunction with anti-segmentation techniques as their use alone is ineffective [3, 27]. There are a number of techniques that can be used as effective recognition-resistance mechanisms, for instance:
  - *Using nonsensical words and different fonts*. Although using random Captcha text instead of dictionary words may negatively impact usability, it also reduces recognizer accuracy. Additionally, using multiple fonts can also significantly decrease classifier accuracy, as well as make the segmentation process more difficult, as character size is unpredictable.
  - *Using distortions.* Distortions, per se, do not provide significant security gains. However, their use, together with anti-segmentation techniques, can greatly enhance Captcha security. Common distortion techniques used in the existing Captcha schemes are basic affine transformations (i.e., translation, scaling and rotation) and nonlinear geometry transformations (i.e., global warp and local warp). It should be noted that overused distortions might degrade Captcha's usability.

2. Offensive approaches

To break text-based Captcha challenges automatically, the attacker needs to solve both the segmentation and recognition problems by reliably identifying character locations and then recognizing each of the segmented characters correctly. Automated tools to solve text-based Captcha schemes typically use a four-stage approach, consisting of:

- ***Pre-processing stage***. In this stage, noise reduction techniques are used to estimate and remove the background noise, colour confusion, mesh, lines, arcs, shapes, and so forth. The purpose of this step is to make the Captcha image clearer and easier to analyse in the subsequent stages. In fact, these techniques are used to eliminate noise by exploiting the differences between noise and target characters in terms of colour, location, shape, size, etc. For example, a thresholding technique is commonly used to eliminate background colours when they are lighter or darker than the text colour. Erosion and dilation are other techniques that can be used to remove background noise and small chunks of lone pixels, as well as reconnect disconnected characters [43].

- ***Segmentation stage.*** After the noise removal phase, the automated program tries to segment the Captcha text into chunks, each containing exactly one character. The isolation of characters is the most difficult step, especially given that most existing text-based Captcha schemes are composed of joined/overlapping characters. Previous successful attacks on text-based Captcha schemes have used different partitioning techniques to split Captcha text into individual characters; for example, colour-filling segmentation (CFS) [31], vertical slicing [30], snake segmentation [30], thresholding [43], colour filtering [44], projection [45], to name but a few.
- ***Recognition stage***. Once characters are reliably segmented, solving the text Captcha becomes a pure problem of OCR that can be easily solved using machine learning techniques. A classifier, such as a convolutional neural network, is usually used to recognize each of the characters. Alternatively, Mori and Malik [46] used shape context matching as another method of character recognition. Yan and El-Ahmad [30] used another simple technique, based on pixel count, to recognize characters.
- ***Post-processing stage.*** The purpose of this step is to improve the accuracy rates produced by classifiers. That is, classifiers in the previous stage typically output a sequence of candidate characters (i.e., 0-9 and a-z). Some of these candidates might be incorrectly recognized. Therefore, if the Captcha scheme utilizes actual dictionary words for its challenges, comparing the candidate string to a dictionary of available words can allow for a better guess, which in turn will improve the accuracy of the Captcha solver.

In sum, most of the attacks over the past few years have relied heavily on these aggressive approaches to establish a prolific history of successful attacks on Captcha schemes, as detailed in the following section.

Attacks on Captcha schemes

Numerous forms of attacks have been successfully launched against different types of Captcha schemes. As a reflection of their almost overwhelming popularity, text-based Captcha schemes have received the bulk of these attacks. This section will discuss some of these attacks (as examples, but not limited to) in order of their chronology.

In 2003, Mori and Malik [41] developed two efficient techniques using shape context matching to break EZ-Gimpy and Gimpy challenges. The first way was applied to the EZ-Gimpy challenge with a success rate of 83%. The other method achieved a success rate of 92% on EZ-Gimpy, and 33% on the requisite three words in a Gimpy challenge. In 2004, Moy et al. [47] developed two different distortion estimation techniques by which to identify an object obscured by clutter to break EZ-Gimpy and Gimpy challenges. The first was a correlation algorithm that had a 99% success rate on EZ-Gimpy challenges. The second technique was a direct distortion estimation algorithm that achieved a success rate of 78% on the four characters in Gimpy-r challenges. In 2005, Chellapilla and Simard [48] worked on a variety of text Captcha schemes taken from the Internet, including Mail blocks, Register, EZ-Gimpy, Ticketmaster, Yahoo version 2 and Google/Gmail. Their approach to breaking all these challenges was to develop a custom algorithm to locate the characters and, subsequently, apply Convolutional Neural Networks for recognition. Success rates of 4.89% - 66.2% were obtained on these Captcha schemes.

In 2007 Yan and El-Ahmad proposed an attack that used only a naïve pixel counting method and simple pattern recognition algorithms [30]; their method achieved an almost 100% success rate against a number of Captcha schemes. In 2008, The same authors have subsequently reported successful attacks on a series of Captcha schemes designed and deployed by Microsoft, Yahoo and Google [31]. In 2009, Tam et al. [49] performed the first security analysis on three types of widely used audio Captcha schemes, including Google, Digg and reCaptcha. Using several machine learning techniques, they succeeded in breaking the three audio Captcha schemes with success rates of 67% for Google, 71% for Digg, and 45% for reCaptcha. In 2010, Yan et al. [50] also broke a novel text Captcha deployed by the Megaupload website which featured a new anti-segmentation technique. A success rate of 63.7% was obtained on this Captcha.

In 2011, Bursztein et al. [27] were able to break 13 out of the 15 most widely acknowledged text-based Captcha schemes using their Decaptcha tool. They reported 1% - 10% success rate on Baidu and Skyrock, 10 - 24% on CNN and Digg, 25 - 49% on eBay, Reddit, Slashdot and Wikipedia, and 50% or higher on Authorize, Blizzard, Captcha.net, Megaupload and NIH. Decaptcha failed (0% success rate) to break the Google and reCaptcha schemes, which were effectively resistant to their attack. However, the two schemes were later broken by Yan's team [51] with an overall success rate of 46.75% on the Google scheme, and 33% on the reCaptcha challenges. In 2012, two different automated attacks were performed separately against the motion-based NuCaptcha. The first was carried out by Xu et al. [52] who launched a four-phase attack that defeated NuCaptcha more than

three-quarters of the time. The second was implemented by Bursztein [53] who performed a five-step attack that achieved a greater than 90% success rate on NuCaptcha.

In 2013, Gao et al. [54] performed the first security analysis of hollow Captcha schemes. A novel attack using a CNN engine with a graph search algorithm was implemented against five hollow Captcha designs deployed by Yahoo, Tencent, Sina, CmPay and Baidu. Their attack achieved success rates ranging from 36% to 89%. In 2014, Mohamed et al. [34] provided the first investigation into the security and usability of game Captcha schemes. Although they explained how these Captcha schemes offered some level of resistance to relay attacks, the game Captcha schemes were vulnerable to a novel dictionary attack developed by Mohamed's team.

In 2016, Gao et al. [55] published a simple generic attack that was capable of breaking a wide variety of text-based Captcha designs, including reCaptcha, Yahoo!, Microsoft, Baidu, Wikipedia, eBay, Amazon, QQ, Taobao, and Sina. Their approach was that of a two-step attack which included the extraction of character components from the Captcha image along four directions using Log-Gabor filters, followed by the use of a k-Nearest Neighbours (KNN) engine to recognize individual characters by combining the most likely adjacent components. In this way, the attack was able to break these Captcha schemes with success rates ranging from about 5% to 77% within 15 seconds. In 2016, Algwil et al. [56] carried out a comprehensive investigation of Chinese Captchas using Convolutional Neural Networks. The study revealed that CNNs can recognize individual Chinese characters with a high success rate regardless of distortion levels. This indicates that many Chinese Captchas in the real world are not secure.

In 2017, Alsuhaibani et al. [57] conducted the first security evaluation of Arabic Captchas and reported that many Arabic Captcha schemes are insecure, being especially vulnerable to segmentation attacks. Their attack revealed that several Arabic Captchas were broken with an acceptable success rate. In 2018, Ye et al. [58] introduced a generic text Captcha solver based on the generative adversarial network. They launched an attack against 33 popular Captcha schemes such as Google, Microsoft, eBay, and Wikipedia and their attack achieved success rates ranging from 3% to 92% within 50 ms.

In 2019, Yu et al. [59] proposed a low-cost chosen-plaintext attack that takes advantage of the nature of open-source Captcha libraries. Their approach combines TensorFlow object detection and a new peak segmentation algorithm with CNN to enhance the recognition accuracy. They revealed acceptable success rates on two open-source Python Captcha Libraries (i.e., *Claptcha* & *Captcha*). In 2020, Wang et al. [60] developed a simple transfer learning-based attack that reduces the complexity and cost in breaking text-based Captcha schemes. Their approach achieved high success rates, ranging from approximately 36% to 97% against 25 popular websites such as Apple, Google, Baidu, Sina, and Wikipedia. In 2021, Wang et al. [61] proposed a fast Captcha solver for text-based Captcha schemes with complex security features. The core idea of their approach was to design a model based on generative adversarial networks to simplify complex Captchas into simple ones. Their attack achieved a high success rate of around 74% against several Captcha schemes within 4-8 ms.

In 2022, Deng et al. [62] proposed an effortless, easy-to-update, and end-to-end solver, called 3E-Solver, for automatically solving text Captcha schemes. 3E-Solver is based on semi-supervised learning that needs fewer labeled Captcha images however can achieve higher accuracy. The solver was able to break eight popular text-based Captcha schemes, including Microsoft, Wikipedia, Apple, Google, Ganji, Yandex, Weibo , and Sina with high success rates ranging from 76.4% to 99.4% within 17 ms. In 2022, Atri et al. [63] developed a simple and effective text-based Captcha solver based on depth first search algorithm for extracting the characters from the Captcha challenges and Convolutional Neural Network for recognizing them. The solver has been validated on over 30,000 Captcha schemes to achieve an average accuracy rate of over 92% within 50 ms. In 2023, Hoang et al. [64] proposed a novel end-to-end Captcha solver, named EnSolver, that uses ensemble uncertainty estimation to detect and skip out-of-distribution Captchas. The solver was able to break eight popular Captcha schemes that have been evaluated in [62] with an average of over 98% accuracy. In 2023, Yusuf et al. [65] introduced a Multiview deep learning architecture to break multiple text-based Captcha schemes. The proposed paradigm uses a combination of convolutional neural networks and recurrent networks. Their attack achieved high success rates ranging from 93.6% to 100%, with an average time of 3.2 ms to 210 ms.

Discussion

Without a doubt, it is crucial to investigate the security aspects of any Captcha scheme. On the other side, the evaluation of the usability aspects of Captcha schemes is also very important. The aim to evaluating Captcha usability is to make sure that the proposed challenges lie in the sweet spot defined by Chellapilla et al. in [66]. The sweet spot, as shown in figure 4, is the confined region where the challenge is solvable by humans, yet

unsolvable by computers. Chellapilla et al. also stated that, based on the attack cost and the service value, the success rate of computer programs in bypassing Captcha challenges should not be more than 0.01% in the sweet spot, while the human success rate should be at least 90% for effective usability.

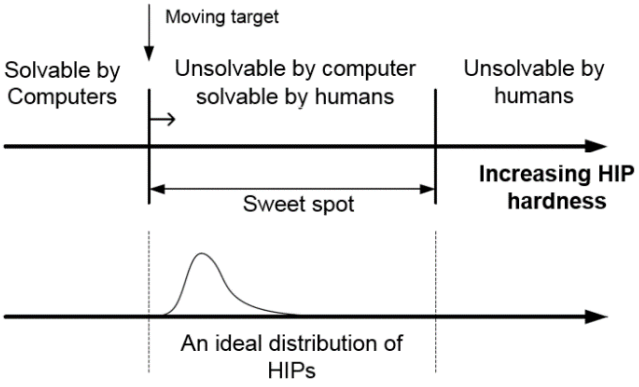


Figure 4 The sweet spot for Captcha challenges [66].

However, the use of sophisticated techniques by both the Captcha designers and attackers in furthering their conflicting goals has narrowed the region of the sweet spot significantly. This, in turn, has made it difficult to design a Captcha scheme that strikes a good balance between the security and usability aspects, not to mention the other desirable characteristics that are preferred as features in any new Captcha design.

Any new Captcha design should fulfil as many of the following requirements as possible. Some of these necessities must be strictly met, such as security, usability and practicality; the fulfilment of others, however, is highly desirable as well.

- ***Secure and robust***. This is the most important requirement that must be met in any Captcha scheme. Captcha challenges should be resistant against current attack technologies as well as being able to withstand attacks for the next few years. In general, a Captcha can be considered secure enough if it costs the attacker more to solve its challenges than the cost of hiring a human to accomplish the same task.
- ***Usable***. This is another urgent necessity that any Captcha must fulfil. Captcha tests should be effective, efficient, and satisfying, so that they can easily be solved by the average person. Simply, there is no benefit to using a secure, but unusable, Captcha scheme, and vice versa.
- ***Practical***. How easy it is to realize the Captcha scheme in practice. The generation and validation processes of Captcha challenges should be fully automated in real time and without human intervention. In addition, these processes should consume minimum network and computational resources.
- ***Scalable***. The auto-generation process of Captcha challenges should meet the requirements of large-scale applications without sacrificing the security and usability of the Captcha scheme.
- ***Learnable***. The Captcha should be intuitive, understandable, and easy to learn with no to little instructions so that it requires neither prior training, particular knowledge, nor specific education.
- ***Universal***. Captcha should be usable by any group of people around the world, regardless of language, age, culture, education, etc. and across different devices (i.e., desktop/laptop, tablets, mobile phones, and so on) regardless of the platforms and operating systems used.
- ***Localizable***. In case of the Captcha not being universally designed, it should be at least localizable into different language editions with only slight effort and resources, and requiring negligible changes to the source code.
- ***Accessible***. Captcha should allow all human users to solve its challenges while preserving access for users with disabilities (i.e., people who have blindness, limited vision, colour blindness, hearing impairments, dyslexia, dyscalculia, and so on).
- ***Customizable***. This refers to how flexible the Captcha is in changing or customizing its appearance and functionality according to consumer preferences or designer necessities.
- ***Integratable with websites***. The Captcha scheme should be properly integrated with web pages based on default browser features and without the need for non-standard software or additional plug-ins. Besides, the Captcha setup should be simple and easy to configure by web developers.

- ***Enjoyable vs. Serious.*** The presence of these two criteria in the same Captcha scheme seems unattainable. For instance, the gamification of Captcha is often described as not representing a serious challenge by many users. On the other hand, text-based Captcha schemes with hard levels of distortion, are usually reported as being a harsh and disagreeable test of humanity. Thus, it will be interesting to design a new text-based Captcha that combines both properties in the same challenge.

Conclusion

Although many Captcha systems have been broken, other improved forms of Captcha designs have been developed. In fact, the 'arms race' between the Captcha developers and attackers has escalated dramatically. Both sides are using increasingly sophisticated technologies to win the race. On the other hand, such advances made text-based Captcha designs very difficult to solve, even for ordinary humans. Accordingly, the distinguishable gap between machine and human abilities in deciphering traditional textual Captcha challenges seems to be inadequate in its current form, often making these schemes either insecure against state-of-the-art attack technologies, or secure but unusable for humans. Thus, there is an increasing need to innovate new, more secure and usable Captcha technology that can be more practical and universally applicable across different devices. We hope that this paper can provide important aspects for Captcha developers to avoid many deficiencies when designing a new Captcha scheme.

References

[1] L. v. Ahn, Human computation. Thesis, Carnegie Mellon University. 2005.

[2] L. v. Ahn, M. Blum , N. J. Hopper, and J. Langford, "Captcha: using hard AI problems for security," in International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, 2003, pp. 294–311.

[3] K. Chellapilla, K. Larson, P. Y. Simard and M. Czerwinski, "Building Segmentation Based Human-Friendly Human Interaction Proofs (HIPs)," in Proceedings of Human Interactive Proofs, Springer, 2005, pp. 1–26.

[4] ArCaptcha - Arabic open source automated test to tell computers and humans apart [Online]. Available: http://arcaptcha.anini.me. [Accessed: 16-08-2023].

[5] B. Khan, K. Alghathbar, M. K. Khan, A. AlKelabi and A. AlAjaji, "Cyber security using Arabic Captcha scheme," Inter. Ar. J. of Info. Tech., vol. 10, no. 1, pp. 76–84, 2013.

[6] Touclick [Online]. Available: http://www.touclick.com. [Accessed: 07-05-2015].

[7] A. Algwil, "A survey on Captcha: origin, applications, classification," J. of Bas. Sci., vol. 34, no. 1, 2023.

[8] Google reCaptcha [Online]. Available: https://www.google.com/recaptcha/intro/. [Accessed: 16-08-2023].

[9] M. Mehrnezhad, A. G. Bafghi, A. Harati, and E. Toreini, "PiSHi: Click the images and I tell if you are a human,". Int. J. of Info. Sec., vol. 16, no. 2, pp. 133–149, 2017.

[10] P. Ray, A. Bera, D. Giri and D.Bhattacharjee, "Style matching Captcha: match neural transferred styles to thwart intelligent attacks," Multi. Sys., pp. 1–24, 2023.

[11] B. Zhao, H. Weng, S. Ji, J. Chen, T. Wang, Q. He and R. Beyah, "Towards evaluating the security of real-world deployed image Captchas," in Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, Toronto, Canada, 2018, pp. 85–96.

[12] H. Weng, B. Zhao, S. Ji, J. Chen, T. Wang, Q. He and R. Beyah, "Towards understanding the security of modern image Captchas and underground Captcha-solving services," Big Data Min. & Anal., vol. 2, no. 2, pp. 118–144, 2019.

[13] V. Fanelle, A. Shah, S. Karimi, B. Subramanian and S. Das, "Blind and human: Exploring more usable audio Captcha designs," in Proceedings of the 16th Symposium on Usable Privacy and Security, 2020, pp. 111–125.

[14] M. Alnfiai, "A novel design of audio Captcha for visually impaired users," Inter. J. of Comm. Net. & Info. Sec., vol. 12, no. 2, pp. 168–179, 2020.

[15] J. P. Bigham and A. C. Cavender, "Evaluating existing audio Captchas and an interface optimized for non-visual use," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2009, pp. 1829–1838.

[16] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell and D. Jurafsky, "How good are humans at solving Captchas? A large scale evaluation," in the 2010 IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 399–413.

[17] G. Sauer, J. Holman, J. Lazar, H. Hochheiser and J. Feng, "Accessible privacy and security: A universally usable human-interaction proof tool," Univ Access Inf Soc, vol. 9, no. 3, pp. 239–248, 2010.

[18] K. Bock, D. Patel, G. Hughey and D. Levin, "UnCaptcha: A low-resource defeat of reCaptcha's audio challenge," in Proceedings of the 11th USENIX Conference on Offensive Technologies, USENIX Association, 2017.

[19] E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry and J. Mitchell, "The failure of noise based non-continuous audio Captchas," in Symposium on Security and Privacy, IEEE, 2011, pp. 19–31.

[20] S. Sano, T. Otsuka and H. G. Okuno, "Solving Google's continuous audio Captcha with HMM-based automatic speech recognition," in Advances in Information and Computer Security, Springer, 2013, pp. 36–52.

[21] R. Bergmair and S. Katzenbeisser, "Towards human interactive proofs in the text-domain," in Proceedings of the 7th Information Security Conference, Springer Berlin Heidelberg, 2004, pp. 257–267.

[22] TextCaptcha v41 [Online]. Available: http://textcaptcha.com/. [Accessed: 16-08-2023].

[23] R. Stevanovic, Quantum random bit generator service [Online]. Available: http://random.irb.hr/. [Accessed: 16-08-2023].

[24] H. Wang, F. Zheng, Z. Chen, Y. Lu, J. Gao and R. Wei, "A Captcha design based on visual reasoning," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2018, pp. 1967–1971.

[25] N. D. Trong, T. H. Huong, and V. T. Hoang, "New cognitive deep-learning Captcha," Sensors, vol. 23, no. 4, 2338, 2023.

[26] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi and K. Cai, "Attacks and design of image recognition Captchas," in Proceedings of the 17th Conference on Computer and Communications Security, ACM, pp. 187–200, 2010.

[27] E. Bursztein, M. Martin and J. Mitchell, "Text-based Captcha strengths and weaknesses," in Proceedings of the 18th ACM Conference on Computer and Communications Security, ACM, 2011, pp. 125–138.

[28] M. Chew and J. D. Tygar, "Image recognition Captchas," in Proceedings of the 7th International Information Security Conference, Springer, 2004, pp. 268–279.

[29] Y. Rui and Z. Liu, "Artifacial: Automated reverse Turing test using facial features," Multi. Sys., vol. 9, no. 6, pp. 493–502, 2004.

[30] J. Yan and A. S. El-Ahmad, "Breaking visual Captchas with naïve pattern recognition algorithms," in the Twenty-Third Annual Computer Security Applications Conference, IEEE, 2007, pp. 279–291.

[31] J. Yan and A. S. El-Ahmad, "A low-cost attack on a Microsoft Captcha," in Proceedings of the 15th ACM Conference on Computer and Communications Security, ACM, 2008, pp. 543–554.

[32] A. Algwil and J. Yan, "Failures of security APIs: A new case," in Proceedings of Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2016, pp. 283–298.

[33] A. Algwil "Click-based Captcha paradigm as a web service," J. of App. Sci., vol. 35, no. 2, pp. 1-26, 2022.

[34] M. Mohamed, N. Sachdeva, M. Georgescu, S. Gao, N. Saxena, C. Zhang, P. Kumaraguru, P. C. Oorschot and W. A. Chen, "Three-way investigation of a game-Captcha: Automated attacks, relay Attacks and usability," in Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, ACM, 2014, pp. 195–206.

[35] D. Danchev, Inside India's Captcha solving economy [Online]. Available: http://www.zdnet.com/article/inside-indias-captcha-solving-economy/. [Accessed: 16-08-2023].

[36] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker and S. Savage, "Re: Captcha - understanding Captcha-solving services in an economic context," in Proceedings of the USENIX Security Symposium, ACM, 2010.

[37] Antigate [Online]. Available: http://antigate.com. [Accessed: 16-08-2023].

[38] Imagetyperz [Online]. Available: http://www.imagetyperz.com. [Accessed: 16-08-2023].

[39] H. D. Truong, C. F. Turner and C. C. Zou, "iCaptcha: The next generation of Captcha designed to defend against 3rd party human attacks," in IEEE International Conference on Communications, IEEE, 2011, pp. 1–6.

[40] S. Gao, M. Mohamed, N. Saxena and C. Zhang, "Emerging image game Captchas for resisting automated and human-solver relay attacks," in Proceedings of the 31st Annual Computer Security Applications Conference, ACM, 2015, pp. 11–20.

[41] K. Chellapilla, K. Larson, P. Simard and M. Czerwinski, "Computers beat humans at single character recognition in reading based human interaction proofs (HIPs)," in the 2nd Conference on Email and Anti-spam (CEAS), 2005.

[42] H. S. Baird and T. Riopka, “ScatterType: A reading Captcha resistant to segmentation attack,” in Document Recognition and Retrieval XII, 2005, pp. 197–208.

[43] A. Hindle, M. W. Godfrey and R. C. Holt, “Reverse engineering Captchas,” in Proceedings of the 15th Working Conference on Reverse Engineering, IEEE, 2008, pp. 59–68.

[44] J. Yan and A. S. El-Ahmad, “Usability of Captchas or usability issues in Captcha design,” in Proceedings of the 4th Symposium on Usable Privacy and Security, ACM, 2008, pp. 44–52.

[45] H. Gao, W. Wang and Y. Fan, “Divide and conquer: An efficient attack on Yahoo! Captcha,” in Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2012, pp. 9–16.

[46] G. Mori and J. Malik, “Recognizing objects in adversarial clutter: Breaking a visual Captcha,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2003, pp. 1–8.

[47] G. Moy, N. Jones, C. Harkless and R. Potter, “Distortion estimation techniques in solving visual Captchas,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2004, pp. 23–28.

[48] K. Chellapilla and P. Y. Simard, “Using machine learning to break visual human interaction proofs (HIPs),” in Advances in Neural Information Processing Systems 17, MIT Press, 2005, pp. 265–272.

[49] J. Tam, J. Simsa, S. Hyde and L. v. Ahn, “Breaking audio Captchas,” in Proceedings of the 21st International Conference on Neural Information Processing Systems, Curran Associates Inc., 2008, pp. 1625–1632.

[50] A. S. El-Ahmad and J. Yan, “Colour , usability and security : A case study,” Technical Report #CS-TR-1203. University of Newcastle Upon Tyne, Computing Science. 2010.

[51] A. S. El-Ahmad, J. Yan and M. Tayara, “The robustness of Google Captchas,” Technical Report #1278. Computing Science at Newcastle University. 2011.

[52] Y. Xu, G. Reynaga, S. Chiasson, J. Frahm, F. Monrose and P. Oorschot, “Security and usability challenges of moving-object Captchas : Decoding codewords in motion,” in the 21st USENIX Security Symposium, USENIX, 2012, pp. 49–64.

[53] E. Bursztein, How we broke the NuCaptcha video scheme and what we propose to fix it [Online]. Available: https://www.elie.net/blog/security/how-we-broke-the-nucaptcha-video-scheme-and-what-we-propose-to-fix-it. [Accessed: 16-08-2023].

[54] H. Gao, W. Wang, J. Qi, X. Wang, X. Liu and J. Yan, “The robustness of hollow Captchas,” in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ACM, 2013, pp. 1075–1086.

[55] H. Gao, J. Yan, F. Cao, Z. Zhang, L. Lei, M. Tang, P. Zhang, X. Zhou, X. Wang and J. Li, “A simple generic attack on text Captchas,” in Network and Distributed System Security Symposium, the Internet Society, 2016.

[56] A. Algwil, D. Ciresan, B. Liu and J. Yan, “A security analysis of automated Chinese turing tests,” in Proceedings of the 32nd annual conference on computer security applications, 2016, pp. 520-532.

[57] S. A. Alsuhibany, N. Alrobah, F. Almohaimeed, S. Alduayji and M. T. Parvez, “Evaluating robustness of Arabic Captchas,” in the 2nd International Conference on Anti-cyber Crimes (ICACC), Abha, Saudi Arabia, IEEE, 2017, pp. 81–86.

[58] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng, P. Xu, X. Chen and Z.Wang, “Yet another text captcha solver: A generative adversarial network based approach,” in Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, 2018, pp. 332–348.

[59] N. Yu and K. Darling, “A low-cost approach to crack Python Captchas using AI-based chosen-plaintext attack,” App. Sci., vol. 9, no. 10, 2019.

[60] P. Wang, H. Gao, Z. Shi, Z. Yuan and J. Hu, “Simple and easy: Transfer learning-based attacks to text Captcha,” IEEE Acc., vol. 8, pp. 59044-59058, 2020.

[61] Y. Wang, W. Yuliang, Z. Mingjin, L. Yang and W. Bailing, “Make complex Captchas simple: a fast text Captcha solver based on a small number of samples,” Info. Sci., vol. 578, pp. 181-194, 2021.

[62] X. Deng, R. Zhao, Y. Wang, L. Chen, Y. Wang and Z. Xue, “3E-Solver: An effortless, easy-to-update, and end-to-end solver with semi-supervised learning for breaking text-based Captchas,” in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2022, pp. 3817–3824.

[63] A. Atri, A. Bansal, M. Khari and S. Vimal, “De-Captcha: A novel DFS based approach to solve Captcha schemes,” Com. & Ele. Eng., vol. 97, pp.107593, 2022..

[64] D.C. Hoang, C.V. Nguyen and A. Kharraz, “EnSolver: Uncertainty-aware Captcha solver using deep ensembles,” arXiv preprint arXiv:2307.15180, 2023.

[65] M. O. Yusuf, D. Srivastava, D. Singh and V.S. Rathor, “Multiview deep learning-based attack to break text-Captchas,” Inter. J. of Mach.Lear. & Cyb., vol. 14, no. 3, pp.959-972, 2023.

[66] K. Chellapilla, K. Larson, P. Simard and M. Czerwinski, “Designing human friendly human interaction proofs (HIPs),” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2005, pp. 711–720.