BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 05/05 pp118–144
Volume 2, Number 2, June 2019
DOI: 10.26599/BDMA.2019.9020001

Towards Understanding the Security of Modern Image Captchas and Underground Captcha-Solving Services

Haiqin Weng, Binbin Zhao, Shouling Ji*, Jianhai Chen, Ting Wang, Qinming He, and Raheem Beyah

Abstract: Image captchas have recently become very popular and are widely deployed across the Internet to defend against abusive programs. However, the ever-advancing capabilities of computer vision have gradually diminished the security of image captchas and made them vulnerable to attack. In this paper, we first classify the currently popular image captchas into three categories: selection-based captchas, slide-based captchas, and click-based captchas. Second, we propose simple yet powerful attack frameworks against each of these categories of image captchas. Third, we systematically evaluate our attack frameworks against 10 popular real-world image captchas, including captchas from tencent.com, google.com, and 12306.cn. Fourth, we compare our attacks against nine online image recognition services and against human labors from eight underground captcha-solving services. Our evaluation results show that (1) each of the popular image captchas that we study is vulnerable to our attacks; (2) our attacks yield the highest captcha-breaking success rate compared with state-of-the-art methods in almost all scenarios; and (3) our attacks achieve almost as high a success rate as human labor while being much faster. Based on our evaluation, we identify some design flaws in these popular schemes, along with some best practices and design principles for more secure captchas. We also examine the underground market for captcha-solving services, identifying 152 such services. We then seek to measure this underground market with data from these services. Our findings shed light on understanding the scale, impact, and commercial landscape of the underground market for captcha solving.

Key words: image captchas; captcha security; captcha-solving service; underground market

1 Introduction

Completely Automated Public Turing tests to tell Computers and Humans Apart (Captcha)[1–4] is a widely used method to increase the security of websites. As shown in Fig. 1, the most popular captchas that are deployed in real world can generally be classified as either text captchas or image captchas. Image captchas require a user to semantically understand the images in a received captcha and perform identification operations (e.g., select semantic images or click semantic regions) according to the on-screen guidance. Nowadays, image captchas are ever more popular because, compared to text captchas, they are more user-friendly and considered more secure. According to a report from Tencent's captcha service, about 1 billion users have solved image captchas (cloud.tencent.com). GEETest

• Haiqin Weng, Binbin Zhao, Shouling Ji, Jianhai Chen, and Qinmin He are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China. E-mail: {hq_weng, bbge, sji, chenjh919, hqm}@zju.edu.cn.
• Ting Wang is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 19019, USA. E-mail: ting@cse.lehigh.edu.
• Raheem Beyah is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30302, USA. E-mail: rbeyah@ece.gatech.edu.
∗ To whom correspondence should be addressed.
Manuscript received: 2018-09-21; accepted: 2019-01-18

(a) Text captcha (b) Image captcha

Fig. 1 Examples of text captchas and image captchas.

(www.geetest.com), another captcha service, reports that they provide image captchas for over 200 000 top websites, including tripadvisor.cn, airbnb.com, and jingdong.com. Google reveals that ReCaptcha challenges are solved as a rate of millions per day (www.google.com/recaptcha/intro/).

1.1 Popular image captchas

Currently, popular real-world image captchas can be roughly classified into three types as shown in Table 1: selection-based image captchas[2], slide-based image captchas, and click-based image captchas[3]. Selection-based captchas ask a user to select candidate images with specific semantic meanings from a set. ReCaptcha, released by Google in 2014, is the most widely used selection-based captcha. In April 2018, about 0.5% of the entire Internet, including 4.7% of the top 1 million sites, 7.3% of the top 1×10^5 sites, and 10.9% of the top 1×10^4 sites, used ReCaptcha to block abusive programs. Slide-based captchas request a user to slide a puzzle to the correct part of an image. Tencent SlidePuzzle is a typical slide-based captcha, employed by many large-scale web services (e.g., qzone.qq.com, which is reported to have 0.56 billion of active users per month). Click-based captchas require a user to click specific semantic regions in an image. Both GEE TouClick and Netease TouClick are representative click-based captchas. In this paper, we evaluate the security of all three of these captcha types.

1.2 Status quo

Ever since it was first mooted, image captcha has been considered as a good alternative to text captcha since it carries richer information, has more room for variation, and is much easier for a human user to complete while being harder for a machine. The security and robustness of text captchas have been widely studied by the research community with many kinds of generic solvers and anti-recognition or defensive techniques

Table 1 Summary of evaluation.

	Selection-based captcha	Slide-based captcha	Click-based captcha
Example			
Captcha provider	12306.com google.com facebook.com	geetest.com tencent.com 163.com	geetest.com tencent.com 163.com
Attack	Sivakorn et al.[5] Ya et al.[6] This paper	This paper	This paper
Application Programming Interface (API)	GoogleAPI TencentAPI AliAPI MirosoftAPI	–	BaiduOCR GoogleOCR TencentOCR AliOCR Face++OCR
Captcha-solving service	ruokuai yundama hyocr 2captcha AntiCaptcha Decaptcha imagetyperz	ruokuai hyocr dama2	ruokuai yundama dama2

(e.g., rotation and distortion) being proposed, along with design guidelines and suggestions for increased security[7–12]. The security of image captchas, on the other hand, is still in need of more comprehensive study. Specifically, existing works either focus on synthetic image captchas[13, 14], or on some particular cases of captcha schemes (e.g., ReCaptcha 2015)[5, 6].

At the same time, the ever-advancing capabilities of computer vision and machine learning are gradually diminishing the security of image captchas and making them vulnerable to attack[15–19]. It is reported that the cognitive ability of a machine can now outperform a human in some complex recognition tasks[20]. Exposed to such powerful techniques, image captchas might become vulnerable. For example, in 2016, Sivakorn et al.[5] utilized deep learning techniques to solve one of the most popular image captcha schemes, ReCaptcha. Ya et al.[6] applied advanced vision techniques to break image captchas from 12306.cn. Also, many commercial companies have deployed powerful online services to undertake vision tasks, such as image classification and object detection, and these services can be maliciously used by abusive programs to break image captchas.

Moreover, there exists a large-scale profit-seeking underground market for captcha-solving services, which support solving almost all types of captchas and significantly threaten the captcha security. For example, ruokuai.com provides services for breaking text, image, and audio captchas. The money involved in the now defunct captcha-solving service qadati.cn is reported to be as much as $3.18 million (kqga.qfc.cn/news/d-1786.html). ruokuai.com, a currently operating captcha-solving service, says that it receives 900 million service requests daily.

Image captchas have increasingly popular, being used by many of the world's biggest websites — including Google, Facebook, and Tencent — to prevent abusive programs. A comprehensive evaluation on the security of image captchas is urgently needed for (1) understanding the vulnerability of image captchas, (2) designing more robust and secure image captchas, and (3) helping website providers defend against abusive programs.

1.3 Methodology

In this paper, we propose three simple yet effective generic attacks, *SelAttack*, *SliAttack*, and *CliAttack*, against selection-based, slide-based, and click-based image captchas, respectively. Our attacks are mainly based on advanced vision techniques and a series of image classification and objection detection models.

First, we evaluate our attacks on 10 popular real-world image captcha schemes provided by top websites, including Google, Facebook, Tencent, and Netease. To our knowledge, seven out of these 10 schemes had never been reported broken prior to this work. Of the 10 schemes, our attacks achieve a $45\% - 70\%$ success rate on three (GEE TouClick, Tencent TouClick, and Netease TouClick), a $70\% - 89\%$ success rate on a further three (ReCaptcha 2015, ReCaptcha 2018, and Facebook), and a $90\% - 100\%$ success rate on the remaining four (China Railway, GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle).

We then compare our three attacks with online recognition services. Specifically, we compare our attacks with nine recognition services provided by five top websites, namely Google, Microsoft, Tencent, Alibaba, and Face++. We evaluate these recognition services on seven specific captcha schemes (i.e., ReCaptcha 2015, ReCatpcha 2018, Facebook, China Railway, GEE TouClick, Tencent TouClick, and Netease TouClick). Compared with our attacks, these recognition services, despite their claim, do not achieve satisfactory attack results with the exception of Google. Nonetheless, they still have the ability to break all of the tested captchas considering to the captcha design goal that holds a captcha scheme to be broken when the attacker is able to achieve a precision of at least 1%[9]. We also find that when equipped with Google's service we can achieve an acceptable success rate of about 0.45 against some schemes.

Further, we employ human labor from eight underground captcha-solving services, including ruokuai.com, 2captcha.com, and anti-captcha.com, to manually break the same 10 real-world captcha schemes as evaluated by our attacks. We find that our attacks outperform those of the most proficient human laborers on four captcha schemes (China Railway, GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle). For the remaining six schemes, the gap between our attack results and those of human labors is acceptably narrow.

Note that for selection-based captchas, we also compare *SelAttack* with two existing state-of-the-art attack methods[5, 6]. The evaluation results suggest that *SelAttack* is more effective, both in terms of success rate

and speed (the time of solving a captcha challenge). In summary, we compare *SelAttack* with two state-of-the-art methods, image recognition services and human labor. We compare the other two attack frameworks only with image recognition services, since to our knowledge they are not reported to have been broken before and thus there are no existing methods of attack to use as benchmarks. Table 1 summarizes the captcha schemes, existing state-of-the-art attacks, APIs of image recognition services, and captcha-solving services that are evaluated in this paper.

1.4 Measurement of the underground market

As part of our research, we identify 152 underground captcha-solving services. Based on these 152 services, we measure the underground market, finding that on average each service has about 3000 laborers available $24/7$ and a daily income of about \$1 million. We then investigate the landscape of the underground market. We find that this market supports almost all captcha categories, including text, image, and game captchas, and meets demand from a great many potential consumers, including account enumeration attackers, third-party services, malicious promotion apps, and coupon stealers.

1.5 Design flaws and countermeasures

Based on our evaluation and findings, we identify several design flaws in popular captcha schemes: (1) selection-based captchas use a limit number of image categories, machine-encoded text hints, and easily recognizable candidate images; (2) slide-based captchas repeatedly use the same images to generate challenges and employ vulnerable malice detection algorithms or even omit such detection for reasons of implementation convenience; (3) click-based captchas fail to apply advanced anti-recognition techniques (e.g., rotation) on distorted characters; and (4) some captcha providers even use the same image set to generate challenges for different schemes. From the results of our attacks, the evaluation of powerful recognition services, and the study of underground captcha-solving services, we devise a set of best practices and design principles for website providers to design secure captchas. We believe that our design principles will be useful for designing more secure image captcha techniques in the future.

1.6 Contributions

We summarize our contributions as follows:

- **Security of popular image captchas**. We implement three simple yet powerful generic attack frameworks that can be used to break a variety of real-world captcha schemes. Our attacks are powerful because they (1) conduct a comprehensive offline analysis for each captcha scheme, (2) collect sufficient data based on this offline analysis, and (3) train accurate and specific image recognition and detection models. Conducting proof-of-concept attacks, we successfully break 10 real-world captcha schemes from popular websites, including google.com, facebook.com, tencent.com, and 12306.cn. We also test the effectiveness of popular image recognition services, and underground captcha-solving services. The evaluation results suggest that our attacks are very powerful and are comparable to human captcha-solving services in terms of attack speed and cost-effectiveness.
- **Analysis of the underground captcha-solving services**. Based on the 152 identified underground services, we conduct a comprehensive measurement of the scale, landscape, and the impact of the underground captcha-solving services. Our findings shed light on the large yet not widely investigated underground economy of captcha-solving services.
- **Countermeasures towards secure image captchas**. Based on our evaluation results, we identify several design flaws in the currently most popular image captchas. We also distill the details of our attacks, our evaluation results, and the identified design flaws into a set of best practices and design principles for website providers to design more secure image captchas.
- **Disclosure of design flaws**. We have submitted reports with our findings and recommendations to each of the captcha providers involved in the study. Of these providers, Tencent and Netease have responded to our reports and acknowledged our findings. We hope that the disclosure of these findings will result in more robust and secure captcha services.

1.7 Roadmap

The rest of this paper is organized as follows: Section 2 provides the background information and reviews the related work. Section 3 introduces the range of popular captchas that we study in this paper. Sections 4, 5, and 6 detail the proposed attack frameworks and describe the corresponding evaluations of attacks on popular real-world captchas. Section 7 provides our study of the underground market for captcha-solving services.

Section 8 addresses the design principles of image captchas and suggests several attack countermeasures. Section 9 discusses the results of the paper, and Section 10 concludes the paper.

2 Background and Related Work

2.1 Threat model

In practice, there are three approaches that adversaries may take to solve image captcha challenges: using automated captcha breaking attacks, using image recognition services, and hiring human labor.

In this paper, we study all the three of these approaches. In relation to automated approaches, we design three attacks and evaluate them against 10 popular real-word captcha schemes. For recognition services, we leverage online image classification and object detection services to solve image captchas. For manual attacks, we hire human labor from a broad range of underground captcha-solving services to break real-world captchas.

To frame the research, we provide background knowledge and review related work covering four aspects: first, we review the most widely used image captchas; second, we outline the existing techniques for attacking image captchas; third, we give an account of the advanced vision techniques and online image recognition services that are currently available; and fourth, we detail popular underground captcha-solving services.

For completeness, we first summarize some representative works on text captchas, the earliest and most traditional form of captchas. Specific attacks[7, 8, 12] and generic solvers[9–11] have been proposed against text captchas. The security issues faced by text captchas have been extensively studied, such as anti-recognition or defensive techniques (e.g., rotation and distortion), and many works have also provided design guidelines and suggestions for more secure text captchas.

Image captchas have now become popular and many existing works therefore focus on the design and analysis of image captchas. Ahn et al.[1] first proposed the use of distorted images of animals for captcha design. Chew and Tygar[2] proposed three image captchas based on naming images, distinguishing images, and identifying an anomalous image out of a set. Elson et al.[21] presented Asirra, a selection-based image captcha that asks a user to identify only the cats out of 12 images of both cats and dogs. With the popularity of face recognition, human faces began to be employed in multiple captcha schemes. Misra and Gaj[22] presented the first captcha scheme based on face recognition, requesting a user to identify two images belonging to the same person. Kim et al.[23] proposed Age-CAPTCHA, a captcha scheme that requires a user to annotate images of human faces with their age group. Based on facial authentication, Uzum et al.[24] proposed the real time Captcha (rtCaptcha) system, which requires a user to perform a known authentic video or visual act (e.g., smile or blink) to figure out the captcha solution.

In addition to the research community, many commercial companies have released various image captcha schemes. The currently popular real-world image captchas can be roughly classified into three types, as shown in Table 1: selection-based image captchas[2], slide-based image captchas, and click-based image captchas[3]. Selection-based captchas (e.g., ReCaptcha) ask a user to select candidate images with specific semantic meanings from a set, slide-based captchas (e.g., Tencent SlidePuzzle) require a user to slide a puzzle to the correct part of an image, and click-based captchas (e.g., Netease TouClick) request a user to click specific semantic regions on an image.

Rather than studying captchas that are not yet deployed, this paper focuses on the security of real-world captchas, because we believe such a study to be more meaningful for understanding the security of the existing captcha ecosystem.

2.2 Existing attacks

Some methods of attack against image captchas have been previous devised. Golle[13] proposed a simple classifier to break the Asirra system. Lorenzi et al.[25] proposed a web service based attack against image captchas, employing three web-services (i.e., reverse image search, image similarity search, and automatic linguistic annotation), to identify the images embedded in a challenge. And they also proposed a recognition based attack against image captchas[14], in which they examined three synthetic captchas: SQ-PIX, ESP-PIX, and Asirra. Most recently, Sivakorn et al.[5] designed a novel attack on ReCaptcha that leveraged deep learning techniques. Ya et al.[6] also developed a novel learning approach in constructing a large association graph, and then applied this graph to break captchas from 12306.cn.

Our work diverges from the aforementioned attacks in the following ways. First, we focus on real-world captchas used by major websites (e.g., google.com, tencent.com, and 12306.cn) instead of targeting captchas that have not yet been deployed. Second, different from Refs. [5, 6], which proposed specially-designed attacks against a few particular cases of selection-based captchas, we develop three generic yet powerful attack frameworks. Our attacks are also more effective and efficient; for example, we achieve a high success rate of 90% against the China Railway scheme, which is employed by 12306.cn, the largest ticketing system in China. Third, we also comprehensively study the security of click-based and slide-based captchas; to the best of our knowledge, this is the first time it has been done. Fourth, we evaluate the captcha-breaking capacity of both image recognition services and also the manual attacks provided by underground captcha-solving services. Finally, we conduct a study measuring the underground market for captcha-solving services estimating its scale, impact, and commercial landscape.

2.3 Computer vision and image recognition services

Recently, research into computer vision has been revolutionized by deep ConvNets[15, 26], and great success is being achieved in many basic vision tasks, such as image classification and objection detection. For example, Convolutional Neural Network (CNN) has been successfully applied to analyze visual imagery, and has proven to be very effective in areas like image classification[15]. Regions with CNN features (RCNN)[16] and its many variants (i.e., Fast-RCNN[27], Faster-RCNN[17], YOLO[18], and SSD[19]) have significantly improved object detection accuracy. In our attacks, we employ advanced mage classification and object detection techniques to recognize images and detect distorted characters.

Benefiting from these advanced vision techniques, many commercial entities also deploy online services to perform various tasks, including image classification services, character recognition services, and object detection services. For example, Google, Microsoft, Baidu, Tencent, Alibaba, and Face++ all provide cloud vision APIs for powerful image analysis. These APIs can be utilized to some extent to perform attacks against image captchas. In our research, to evaluate the performance of these services and permit a comprehensive comparison with our attacks, we test four image classification services (GoogleAPI①, TencentAPI②, MicrosoftAPI③, and AliAPI④) and five character recognition services (BaiduOCR⑤, TencentOCR⑥, GoogleOCR⑦, AliOCR⑧, and Face++OCR⑨). We select these particular classification services since they are the most popular in the research community, and these particular character recognition services because they are widely used in recognizing Chinese characters (many of the studied captchas in this paper are in Chinese) and claim to achieve a high level of recognition accuracy.

The captcha arms race has created to a large-scale underground market for captcha-solving services, which mainly operate by hiring human labor to solve captchas. Motoyama et al.[28] explored the inner working mechanisms of captcha-solving services, while Shin et al.[29] analyzed the functionality of a popular forum spam automator, revealing that it can intelligently bypass many of the practices used to distinguish humans from bots.

Motivated by previous work, for our research, we also hire human labor from underground captcha-solving services to manually attack popular captcha schemes. Specifically, we employ eight captcha-solving services (ruokuai⑩, yundama⑪, dama2, hyocr⑫, 2captcha⑬, AntiCaptcha⑭, DeCaptcha⑮, and imagetype⑯) to target different captcha schemes. We select these particular services because (1) they support the captcha schemes studied in this paper, as detailed in Section 3; (2) they are popular and widely used by miscreants; and (3) some of them have been used in previous work[5, 28]. In our research, we also conduct a study measuring the underground market for captcha-solving services. Different from Ref. [29], our study focuses on the resolution of novel image captchas, and looks at the scale and commercial landscape of the

① cloud.google.com/vision/.
② youtu.qq.com/img-content-identity.
③ azure.microsoft.com/zh-cn/services/cognitive-services/computer-vision/.
④ data.aliyun.com/ai?spm=a2c0j.9189909.810797.13.64c6547a3VOVGD\#/image-tag.
⑤ cloud.baidu.com/product/ocr.html.
⑥ ai.qq.com/product/ocr.shtml#identify.
⑦ cloud.google.com/vision/docs/ocr.
⑧ www.aliyun.com/product/cdi/.
⑨ www.faceplusplus.com.cn/general-text-recognition/.
⑩ www.ruokuai.com/.
⑪ www.yundama.com/.
⑫ www.hyocr.com/.
⑬ 2captcha.com/.
⑭ anti-captcha.com.
⑮ decaptcher.com.
⑯ www.imagetyperz.com/.

underground market, and the impact of this market on benign users and industries.

3 Popular Real-World Image Captchas

To collect representative image captchas, we consult the Alexa list of the most used websites (www.alexa.com/topsites) , and identify four top sites that provide image captcha services to other sites, namely, ReCatpcha, GEETest, Tencent, and Netease (www.163.com). We collect a total of eight schemes from these sites. Additionally, we collect two schemes of selection-based captchas from sites that design their own captchas: 12306.cn and facebook.com. Table 2 summarizes the 10 schemes we collect to establish our study.

The 10 collected schemes all fit into the three popular image captcha categories: selection-based captchas, slide-based captchas, and click-based captchas. In this paper, we use this broad range of captchas to evaluate the effectiveness and efficiency of our attacks. Below, we show the design and workflow and provide an example of each.

3.1 Selection-based image captchas

For selection-based captchas, we collect four popular schemes, namely ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway.

ReCaptcha, offered by Google, aims to verify users if possible without requiring them to actually solve a tedious challenge. ReCaptcha first requires a user to click a *checkbox* and calculates a confidence score for this user according to many risk factors returned by the checkbox, e.g., browser characteristics and google.com cookies. ReCaptcha then returns a selection-based captcha for users with low scores, whereas users with high scores can directly pass the challenge without any further authentication.

In this paper, we mainly focus on the selection-based captchas returned by ReCaptcha. ReCaptcha has two versions, namely ReCaptcha 2015 and ReCaptcha 2018.

ReCaptcha 2015. Figure 2a shows an example of ReCaptcha 2015. This challenge contains one sample image and nine candidate images. To pass the challenge, a user is requested to select all images that are similar to the sample image. ReCaptcha 2015 was released by Google in 2015.

ReCaptcha 2018. Figure 2b shows an example of ReCaptcha 2018. This challenge consists of one hint and 16 candidate images. To pass this challenge, a user is asked to select all images that are relevant to the hint. ReCaptcha 2018 is the currently the newest version of ReCaptcha.

Facebook. Figure 2c shows an example of a captcha used by facebook.com. This challenge contains one hint and 12 candidate images. To pass this challenge, a user is required to select all images that are relevant to the hint.

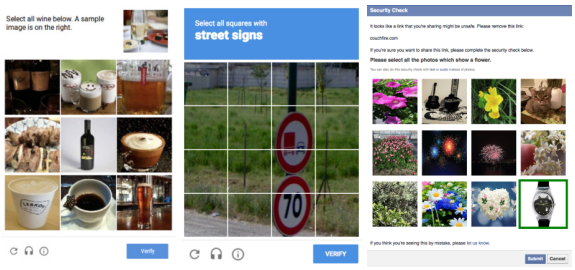
China Railway. Figure 2d shows an example of a China Railway captcha. It contains one hint of distorted characters and eight candidate images. To pass this challenge, a user is required to select all images that are relevant to the hint. The China Railway scheme is used by China's largest railway ticketing system, 12306.cn.

3.2 Slide-based image captchas

For slide-based image captchas, we collect three popular real-world schemes, namely GEE SlidePuzzle,

Table 2 Summary of real-world image captchas.

Type	Scheme	Provider	Scale
Selection-based captcha	ReCaptcha 2015	ReCaptcha	O(million) users
	ReCaptcha 2018	ReCaptcha	
	China Railway	12306.cn	O(billion) users
	Facebook	Facebook	–
Slide-based captcha	GEE SlidePuzzle	GEETest	$O(2 \times 10^5)$ sites
	Tencent SlidePuzzle	Tencent	O(billion) users
	Netease SlidePuzzle	Netease	O(billion) users
Click-based captcha	GEE TouClick	GEETest	$O(2 \times 10^5)$ sites
	Tencent TouClick	Tencent	O(billion) users
	Netease TouClick	Neteast	O(billion) users



(a) ReCaptcha 2015 (b) ReCaptcha 2018 (c) Facebook

(d) China Railway

Fig. 2 Examples of selection-based image captchas.

Tencent SlidePuzzle, and Netease SlidePuzzle.

Figure 3 shows examples of GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle. Each of these challenges contains one puzzle and one background image. To solve those challenges, a user is requested to slide the puzzle to the correct part of the background image. The captcha providers check whether the puzzle window is accurately placed or not, and make a risk analysis on the slide trajectory. A user is considered to pass the challenge if and only if the puzzle window is correctly placed and the slide trajectory is not suspicious. As shown in Table 2, all of these schemes are provided by top captcha services, and widely used by the most popular sites.

3.3 Click-based image captchas

For click-based image captchas, we collect three popular real word schemes, namely GEE TouClick, Tencent TouClick, and Netease TouClick.

GEE TouClick. Figure 4a shows an example of GEE TouClick. This challenge contains one hint of distorted characters, and one background image also made up of distorted characters. To solve this challenge, a user is asked to sequentially click the characters drawn in the background image according to the hint and in the right order. Note that there are the same number of distorted characters in the hint and in the background image.

Tencent TouClick. Figure 4b shows an example of Tencent TouClick. The structure and workflow of Tencent TouClick are similar to those of GEE TouClick, except that in this case that there are more distorted characters in the background image than there are in the hint.

Netease TouClick. Figure 4c shows an example of Netease Touclick. This challenge consists of one hint made up of machine-encoded characters, and one background image made up of distorted characters. To pass this challenge, a user is asked to sequentially click the distorted characters in the correct order.

As shown in Table 2, GEE TouClick, Tencent TouClick, and Netease TouClick are all supplied by proficient captcha service providers and widely used by many major sites.

4 Security of Selection-Based Captchas

In this section, we first design *SelAttack*, an attack framework *SelAttack* targeting selection-based captchas. We then evaluate SelAttack against four popular real-world captcha schemes: ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway. Finally, we discuss some design flaws in the tested schemes.

4.1 SelAttack

Selection-based captchas require a user to correctly select images with specific semantic meanings. Hence,

(a) GEE SlidePuzzle (b) Tencent SlidePuzzle (c) Netease SlidePuzzle

Fig. 3 Examples of slide-based image captchas.

(a) GEE TouClick (b) Tencent TouClick (c) Netease TouClick

Fig. 4 Examples of click-based image captchas.

it is intuitive that an image classification model can be utilized to understand the semantic meanings of candidate images and determine the correct ones.

Below, we first give several notations, and then show the detailed steps of our attack.

4.1.1 Notations

A selection-based captcha contains two parts: a hint consisting of a short phrase (e.g., "car" or "street signs") and several candidate images. There are usually two types of hints: a *text* hint, which is presented in the format of machine-encoded text, and an *image* hint, which is presented as an image of distorted characters.

4.1.2 Design of SelAttack

Based on the workflow of selection-based captchas, we design our attack, as illustrated in Algorithm 1. The attack proceeds as follows: (1) To bootstrap our attack, we pre-train an image classification model. We also pre-train a character recognition model if the target scheme contains an image hint. (2) Upon receiving a challenge, we first extract the candidate images and the hint directly from the HTML DOM elements of the received captcha. If it is an image hint, we then perform image recognition on the hint. Note that this process is designed to transform the distorted characters in the image hint into machine-encoded text. (3) Next, we utilize the classification model to recognize candidate images and predict their semantic labels. (4) Finally, we select as the solution of the given captcha those images found to be relevant to the hint. The entire attack pipeline is illustrated in Fig. 5.

Algorithm 1 SelAttack

Input: Image captcha I.
Output: The captcha solution S.
1 $h \leftarrow$ **extract_hint**(I)
2 **if** h *is an image* **then** h = **recognize_hint_image**(h)
3 $C \leftarrow$ **extract_candidate_images**(I)
4 $L \leftarrow$ **recognize_candidate_images**(C)
5 $S \leftarrow$ **select_candidate_images**(C, L, h)
6 **return** S

4.2 Evaluation of SelAttack

To evaluate the effectiveness, efficiency, generality, and the cost-effectiveness of *SelAttack*, we conduct a set of experiments on four different captcha schemes: ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway. Based on our evaluation, we identify several design flaws in the currently popular selection-based captchas.

Setup. First, we conduct a preliminary empirical analysis on the four tested schemes, especially looking at the hint capacity (i.e., the number of unique hints). Based on the preliminary analysis, we collect five datasets with sufficient labeled images for bootstrapping our attack. To be specific, these datasets are used for training five image classification models, namely CNN_1, CNN_2, CNN_3, CNN_4, and CNN_5. Equipped with these models, we run SelAttack against the four tested schemes.

Then, we leverage four online image recognition services — AliAPI, GoogleAPI, MicrosoftAPI, and TencentAPI — to attack the considered captchas, and compare the results with our own attack.

As a comparison with prior methods, we test two state-of-the-art attacks: Ya et al.[6] and Sivakorn et al.[5], which claim to be cost-effective and widely applicable. While those attacks have previously been evaluated against only two schemes, we fine-tune them and apply them on all four of the tested schemes.

To compare our attack method with human labor, we evaluate the effectiveness and efficiency of seven popular human captcha-solving services, i.e., ruokuai, yundama, 2captcha, hyocr, AntiCaptcha, DeCaptcha, and imagetype. We select these seven services on the basis that they support selection-based captchas.

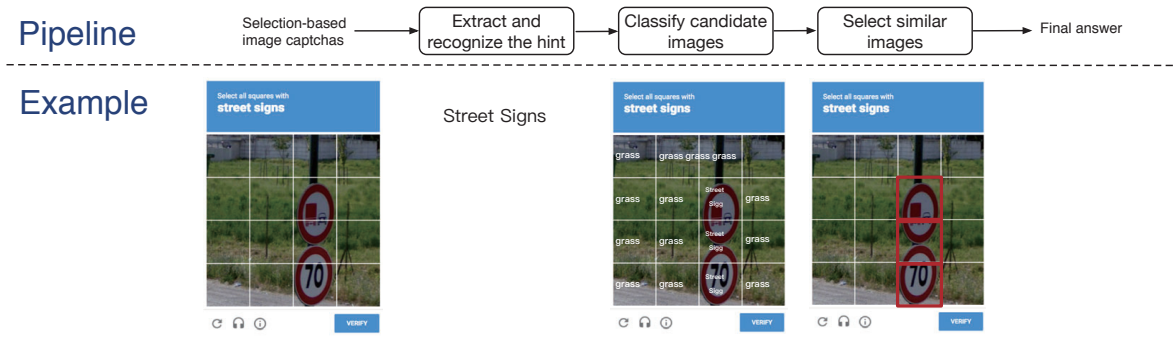


Fig. 5 Attack pipeline for selection-based image captchas.

4.2.1 Preliminary analysis

Before performing the attack, we make a preliminary analysis of the four tested schemes, especially looking at their hint capacities. This analysis serves as a guidance for training our attack models.

We focus on two primary questions when analyzing the captcha schemes: how many hints are there, and what is the content of the hints. To answer these questions, we employ a methodology that combines the continuous observation of live online captchas with the statistical analysis of historical datasets of pre-downloaded captchas. Below, we list the results of our preliminary analysis of the four schemes (as summarized in Table 3).

ReCaptcha 2015. ReCaptcha 2015 has 22 frequent used hints, including "banana", "beer", and "bread". We obtain this result from a statistical analysis of about 700 pre-downloaded captchas, because ReCaptcha 2015 is temporarily unavailable and therefore only the historical dataset is accessible.

ReCaptcha 2018. ReCaptcha 2018 has seven frequent hints, including "house", "cat", and "chair". We obtain this result through a continuous one-month observation of real captchas from 2018-02-10 to 2018-03-13.

Facebook. Facebook has 12 distinct hints, including "bicycle", "cat", and "chair". As for ReCaptcha 2015, we obtain this result from a statistical analysis on about 200 pre-down loaded captchas.

China Railway. China Railway has 80 distinct hints, including "Chinese knot", "dashboard", and "refrigerator", which is the largest of hints used by any of the tested schemes. We obtain this result through a three-month observation on real captchas on 12306.cn from 2017-08-15 to 2017-10-20.

We conjecture that the reasons for the tested schemes having only a limited number of hints are as follows. (1) Implementing a selection-based captcha with a small size of hints is simple and convenient. (2) Collecting and labeling images from a wide range of categories is time consuming and expensive, even though it is theoretically more secure.

4.2.2 Data collection

According to the guidance given by the preliminary analysis, we collect labeled datasets for training image classification models. We employ a methodology that combines automated crawling, synthetic generation, and collection of benchmark datasets. In summary, we collect five labeled datasets: D_1, D_2, D_3, D_4, and D_5 (as illustrated in Table 4). Below, we briefly describe these five datasets.

(1) D_1. To break ReCaptcha 2015, we collect images in 22 categories from the image searching results of google.com and baidu.com. Additionally, we employ labeled images from the ImageNet[30] benchmark, which is a large visual database designed for visual object recognition research that contains over 14 million of hand-annotated images. These are combined into a dataset, denoted by D_1, consisting of 33 000 images with 1500 per category.

(2) D_2. To break ReCaptcha 2018, we collect labeled images in 10 categories from the image searching results of google.com and baidu.com, and from ImageNet. This dataset, denoted by D_2, contains 15 000 images with 1500 per category.

(3) D_3. To break Facebook, we collect labeled images of 12 categories from the image searching results of google.com and baidu.com, and ImageNet.

Table 3 Statistics of the four schemes.

Scheme	Number of categories	Category
ReCaptcha 2015	22	artichoke, avocado, banana, beer, bread, cabbage, cake, cat, coffee, dog, guinea pig, hamburger, ice cream, pasta, pizza, rice dish, rose, sandwich, soup, steak, sushi, wine
ReCaptcha 2018	7	house, road, sky, street sign, telephone pole, tree, vehicle
Facebook	12	bicycle, cat, chair, cloud, dog, fireworks, flower, guitar, lion, tiger, waterfall, wristwatch
China Railway	80	Chinese knot, dashboard, bus card, refrigerator, band Aid, embroidery, paper cut, seal, tape measure, double-sided adhesive, whistle, beer, helmet, corkscrew, palm print, typewriter, cuff, mop, wall clock, ventilator, pencil case, calendar, notebook, portfolio, cotton swab, cherry, woolen, sandbags, salad, poster, seaweed, seagull, funnel, candlestick, hot-water bottle, archway, lion, coral, electronic scales, wire , rice cooker, plate, basketball, jujube, red bean, red wine, mung bean, tennis racket, tiger, earplug, aircraft carrier, fly swatter, tea table, tea cup, pill, pineapple, steamer, french fries, ant, bee, candle, lizard, stapler, plum, palette treadmill, street light, chili sauce, pyramid, clock, bell, spatula, gong, pennant, rain boots, firecrackers, campanula, pressure cooker, blackboard, dragon boat

Table 4 Labeled datasets.

Dataset	Number of image categories	Number of images per category	Number of images	Source	Usage
D_1	22	1500	33 000	ImageNet, baidu.com, google.com	ReCaptcha 2015
D_2	10	1500	15 000	ImageNet, baidu.com, google.com	ReCaptcha 2018
D_3	12	1500	18 000	ImageNet, baidu.com, google.com	Facebook
D_4	80	1500	120 000	ImageNet, baidu.com, google.com	China Railway
D_5	80	about 750	60 000	12306.cn	China Railway
D_6	3755	about 1400	5 257 000	Synthetic character generator (github.com/AstarLight)	GEE, Tencent, Netease TouClick
D_7	–	–	2000	geetest.com	Gee TouClick
D_8	–	–	2000	open.captcha.qq.com	Tencent TouClick

This dataset, denoted by D_3, contains 18 000 images with 1500 per category.

(4) D_4 and D_5. To break China Railway, we collect labeled images in 80 categories from the searching results of google.com and baidu.com, and from ImageNet. This dataset, denoted as D_4, consists of 120 000 images with 1500 per category. In addition, we collect captcha challenges from 12306.com, and manually label the distorted image hints. These are combined into a dataset, denoted as D_5, consisting of 60 000 images of distorted hints with about 750 per category.

4.2.3 Attack models

We train CNN_1 for breaking ReCaptcha 2015, train CNN_2 for breaking ReCaptcha 2016, train CNN_3 for breaking Facebook, and train CNN_4 and CNN_5 for breaking China Railway. All of the five classification models are trained on an ubuntu server equipped with an Intel i5-7500 CPU, a GTX 1060 GPU, and 16 GB memory. The models are trained through the standard five-fold cross validation; that is, four-fifths of the data is used for training the CNN model and the remaining one-fifth for evaluating the accuracy of the trained model. There is no overlap between the training and validation datasets. Below, we detail the use of, training processes for, and results of these five models (as summarized in Table 5).

(1) CNN_1. CNN_1 is used to predict the label for each candidate image from ReCaptcha 2015 challenges. We train CNN_1 on D_1 with a batch size of 16 and a learning rate of 1×10^{-4}. This training process lasts for 18 hours, and finally achieves a high image-recognition accuracy of 0.9597 in recognizing images. This training result also shows that our pre-trained image classifier, CNN_1, has good potential for annotating candidate images for ReCaptcha 2015.

(2) CNN_2. CNN_2 is used to label each candidate image for ReCaptcha 2018 challenges. Since the image categories of ReCaptcha 2018 are different from those of ReCaptcha 2015, it is necessary to train another well-designed classifier. We train CNN_2 on D_2 with a batch size of 16 and a learning rate of 1×10^{-4}. The training process of CNN_2 lasts for 8 hours, and finally achieves an image-recognition accuracy of 0.9177.

(3) CNN_3. CNN_3 is used for labeling candidate images from Facebook challenges. As with CNN_1 and CNN_2, we train CNN_3 on D_3 with a batch size of 16 and a learning rate of 1×10^{-4}. The training process of CNN_3 lasts for 9 hours, and finally achieves an image-recognition accuracy of 0.9727.

Table 5 Summary of pre-trained deep models.

Model name	Model type	Accuracy	Training time (h)	Usage
CNN_1	CNN	0.9597	18	Recognize images for ReCaptcha 2015
CNN_2	CNN	0.9177	8	Recognize images for ReCaptcha 2018
CNN_3	CNN	0.9727	9	Recognize images for Facebook
CNN_4	CNN	0.9327	93	Recognize phrases for China Railway
CNN_5	CNN	0.9661	25	Recognize images for China Railway
CNN_6	CNN	0.9986	17	Recognize distorted characters
Fast-RCNN_1	R-CNN	0.9201	7	Localize objects for GEE TouClick
Fast-RCNN_2	R-CNN	0.9712	12	Localize objects for Tencent TouClick

(4) CNN_4 and CNN_5. Both CNN_4 and CNN_5 are used for breaking the China Railway captcha scheme. Specifically, CNN_4 is used for labeling candidate images, and CNN_5 is used for recognizing hints made up of distorted Chinese phrases. We train CNN_4 on D_4, and train CNN_5 on D_5. The training process of CNN_4 lasts for 93 hours, and finally achieves a high image-recognition precision of 0.9327. The training process of CNN_5 lasts for 25 hours, and finally achieves a high precision of 0.9661 in recognizing hints made up of distorted Chinese phrases.

4.2.4 Attack results

Equipped with the five pre-trained models, we now run SelAttack against captchas from ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway. For the two temporarily inactive services, ReCaptcha 2015 and Facebook, we perform our attack against 684 and 200 pre-downloaded challenges, respectively. For the two live captcha services, ReCaptcha 2018 and China Railway, we perform our proof-of-concept attack against 200 real online captchas from each, limiting the number so as to minimize our impact.

To validate our attack results on inactive services, we manually inspect the captcha challenges and figure out the correct solutions. Table 6 shows the success rate and speed of our attack on the four schemes. The success rate here is the fraction of attempts at breaking the selection-based captcha challenges among a number of attempts that are successful.

Success rate. Our attack's success rate ranges from 0.79 to 0.90, which is relatively high. Taking China Railway as an example, it is reported that only 2%, 27%, and 65% of human users successfully pass the captcha on their first, second, and third attempts, respectively (baike.baidu.com/item/12306). Our attack achieves its highest success rate of 0.90 on China Railway. The lowest success rate of 0.79 is achieved on ReCaptcha 2018, which is still very high when benchmarked against the minimum successful breaking rate for automated attacks of 1% suggested by Ref. [8]. The greater difficulty involved in breaking ReCaptcha 2018 might be explained as follows: (1) ReCaptcha 2018 has a larger number of candidate images, which might introduce more classification errors; and (2) ReCaptcha 2018 has several confusing image categories, e.g., bridges and roads, which are difficult even for human users to recognize.

Speed. On average, our attack takes between one and five seconds to break each of the tested schemes, which is relatively fast. We note that the time to solve ReCaptcha 2018 and China Railway includes a network delay overhead estimated at three seconds per captcha. If we exclude the network overhead, the fastest speed is achieved on China Railway, while the slowest speed, about 2 seconds, is achieved on ReCaptcha 2018.

We find that the solving time excluding network overhead scales linearly as the candidate image size increases. Figure 6 illustrates this linear growth characteristic, which suggests that our attack is scalable in practice, and that a parallel implementation of SelAttack could therefore be applied to solve large numbers of image captchas.

Table 6 Attack results on selection-based image captchas. "–" stands for not given.

Method		ReCaptcha 2015		ReCaptcha 2018		Facebook		China Railway	
		Success rate	Speed (s)	Success rate	Speed (s)	Success rate	Speed (s)	Success rate	Speed (s)
Our method		0.88	1.26	0.79	4.92	0.86	1.41	0.90	4.14
Prior art	Ya et al.[6]	0.14	0.59	–	–	0.09	0.47	0.52	6.62
	Sivakorn et al.[5]	0.71	20.80	–	–	0.83	25.30	0.37	20.60
Image recognition service	TencentAPI	0.19	13.64	0.06	20.19	0.25	15.32	0.03	14.97
	GoogleAPI	0.62	16.13	0.49	23.31	0.73	19.53	0.07	17.82
	AliAPI	0.37	14.27	0.11	18.40	0.35	13.04	0.16	12.65
	MirosoftAPI	0.21	19.95	0.08	25.42	0.44	21.09	0.02	17.01
Captcha-solving service	ruokuai	0.81	4.54	0.91	6.97	0.88	4.21	0.86	5.57
	yundama	0.89	4.36	–	–	0.77	5.18	0.88	5.29
	hyocr	–	–	0.85	7.05	–	–	–	–
	2captcha	0.86	8.35	0.88	4.27	0.90	7.98	0.79	11.37
	AntiCaptcha	0.84	6.43	0.92	5.69	0.93	8.71	0.65	9.94
	DeCaptcha	0.41	23.16	0.62	31.12	0.46	25.24	–	–
	imagetyperz	–	–	0.95	41.68	–	–	–	–

Note: Speed is defined as the time of solving a captcha challenge.

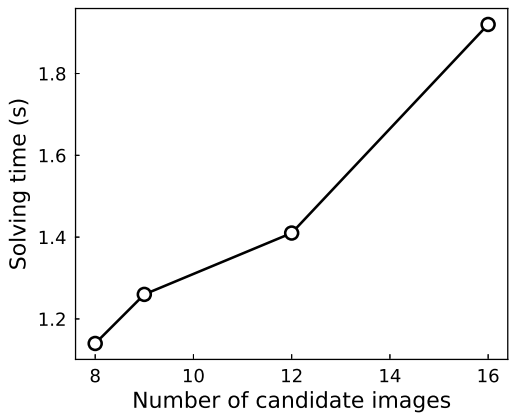


Fig. 6 Captcha-solving time with various number of candidate images.

In summary, the high success rate and short solving time indicate that SelAttack poses a realistic threat to selection-based captcha schemes.

4.2.5 Comparison with image recognition services

Alongside the pre-trained attack models, we also solve captchas by calling third-party online image classification APIs to solve captchas. We evaluate attacks against selection-based captchas leveraging API calls from four popular services, namely GoogleAPI, TencentAPI, MicrosoftAPI, and AliAPI.

For each recognition service, we request API calls for 400 challenges with 100 per captcha scheme, constituting a total of 17 600 calls for image recognition. Table 6 shows the success rate and speed of the attacks leveraging the four image recognition services.

When compared with SelAttack, the success rate of the recognition service based attacks is relatively low, ranging from 0.02 to 0.73. Among all the tested services, GoogleAPI achieves the best performance, with a success rate larger than 0.49 on a majority of schemes. For the remaining recognition services, the highest success rate is 0.44.

In summary, from the above analysis, we conclude that (1) our attack models are well-designed and effectively trained, proving to be very powerful for breaking selection-based captchas; and (2) some online services (e.g., GoogleAPI) can provide a compromise option when time is short or the computing environment is limited.

4.2.6 Comparison with previous methods

Recently, Sivakorn et al.[5] designed a novel attack that leverages deep learning techniques for image annotation to break selection-based image captchas. Ya et al.[6] also developed a novel learning approach for constructing a large association graph, which was then applied to breaking selection-based captcha schemes. Therefore, we compare our attack results on ReCaptcha 2015, Facebook, and China Railway with these two alternative methods, which are both considered cost-effective. For the experimental comparison, all the parameters in the previous methods are properly selected and each of the tested methods uses the same training dataset.

ReCaptcha 2018 is different from the other tested schemes in that its candidate images are derived from the same source image (in other words, the candidate images form a single background image and are correlative with each other). We do not evaluate the two previous methods on ReCaptcha 2018 since this feature makes it impossible for Ya et al.[6] to find a prominent co-occurrence relationship between candidate image pairs and difficult for Sivakorn et al.[5] to train a ⟨label, hint⟩ similarity classifier, as each label of the correlative candidate image is similar to the hint.

Table 6 shows the success rates and speeds of the two previous methods. Note that since Sivakorn et al.[5] also evaluated their method on the same pre-downloaded captchas of ReCaptcha 2015 and Facebook as we used in our test, we simply report their results in Table 6.

From the comparison with previous methods, we can see that (1) our attack achieves the highest success rate and operates at a comparatively high speed on all schemes; (2) although Ref. [5] achieves a tolerable success rate, it has the highest time consumption among the tested methods, which might raise timeout errors in practical attack scenarios; and (3) the success rates of Ref. [6] are relatively low as compared to those of our attack. Our proposed SleAttack is more successful than both Sivakorn et al.[5] and Ya et al.[6] since (1) Sivakorn et al.[5] utilized online services (e.g., Google's reverse image search service and the Clarifai service) for the semantic annotation of images, which is a low accuracy method, and (2) Ya et al.[6] mainly leveraged the co-occurrence relationships between candidate image pairs to solve selection-based image captchas, which might not work in case where those relationships are weak.

Overall, the results suggest our attack is superior in terms of both success rate and speed when compared with existing attacks.

4.2.7 Comparison with human labor

Finally, we compare our attack results with human labor from seven proficient captcha-solving services, namely

ruokuai, yundama, hyocr (from China), AntiCaptcha, DeCaptcha, imagetyperz (from the USA), and 2captcha (from Russa)[①].

Due to budget limits, for each captcha-solving service, we submit 400 challenges with 100 per captcha scheme. Table 6 summarizes the success rates and speeds of the seven captcha-solving services. Note that we do not report the result of yundama, hyorc, DeCaptcha, and imagetyperz on some schemes that they do not support.

The success rate of human labor on all schemes ranges from 0.41 to 0.93, and the solving time ranges from 4.54 to 41.68 seconds. The poorest captcha-solving service is DeCaptcha, which simultaneously has the lowest success rate and the slowest speed. Compounding this, we receive a large percentage of timeout errors (i.e., cases where a solution is not provided within the allowed time window) from DeCaptcha. We conjecture that a lack of sufficient online human labor gives DeCaptcha its poor user experience.

We then compare our attack results with those of human labor. Surprisingly, on China Railway we find that our attack has an even higher success rate than that of the most proficient human labor. As for other schemes, the gap between our attack and proficient human labor is negligible. In terms of speed, our attack is naturally much faster than human labor.

The above analysis suggests that our attack is comparable to captcha-solving services in attack accuracy and more efficient in attack duration.

4.3 Design flaws

Based on the evaluation results, we summarize the following design flaws in real-world selection-based captchas. First, all of the tested schemes have a limited number of hints, meaning that an adversary can easily enumerate all of the hints and train an accurate image classification model. Second, most of the tested schemes feature a text hint, which can be extracted with little effort. Third, the candidate images have little resilience from the security perspective (usually there is no noise); therefore, a well-trained model can accurately uncover their semantic meanings.

5 Security of Slide-Based Image Captchas

In this section, we detail the design of *SliAttack*, which targets slide-based captchas. We then evaluate SliAttack against three popular real-world captchas: GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle. Based on our evaluation, we also disclose several design flaws in currently popular slide-base image captchas.

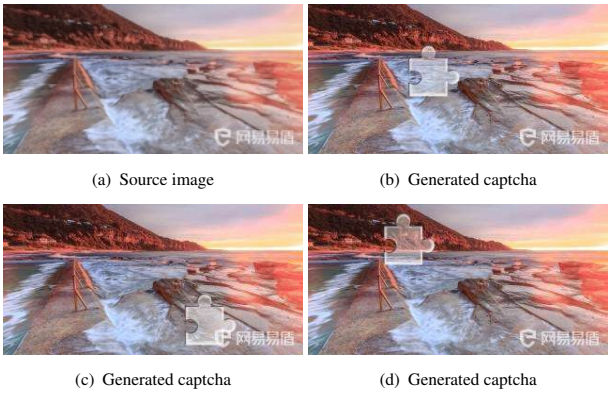
5.1 SliAttack

Slide-based captchas ask a user to slide a puzzle to the correct part of an image. For convenience, we name this correct part the *puzzle region*. The key to automatically breaking this captcha is to accurately find the puzzle region, and to mimic human behavior when sliding the puzzle window. Before introducing our attack design, we first describe how to find the puzzle region and mimic human behavior.

5.1.1 Puzzle region detection

Through an analysis of 2000 slide-based captchas, we observe that a single source image is repeatedly used to generate a great many captcha challenges in real captcha systems, such as Tencent SlidePuzzle and Netease SlidePuzzle. This is shown in Fig. 7, in which Fig. 7a is an example of the source image, and Figs. 7b–7d are three different challenges generated from that same image. Based on this observation, it is intuitive that a source image can be recovered by analyzing a set of captchas that it generates, and the comparison between a captcha and its source image can be used to accurately locate the puzzle region. Note that we do not utilize advanced object detection models here, since they can only approximately locate a puzzle region, which significantly decreases the attack's success rate. Hence, we detect the puzzle region in two steps: (1) source image recovery and (2) comparison-based region detection.

Source image recovery. Let s denote a source image, and $I^s = \{I_i^s | i = 1, 2, \ldots, m\}$ be the set of background images generated from s. We further define



(a) Source image (b) Generated captcha

(c) Generated captcha (d) Generated captcha

Fig. 7 3 captchas and their corresponding source image.

① The locations of captcha-solving services are determined based on their IP addresses.

$I_i^s = \{I_i^s(j)|j = 1, 2, \ldots, n\}$, where $I_i^s(j)$ is the j-th pixel of I_i^s, $m, n \in \mathbf{R}$.

We now briefly introduce our source image recovery algorithm, as illustrated in Algorithm 2. It involves: (1) Single pixel reconstruction (lines 4–6): we construct the j-th pixel of s by selecting the most frequent value from the pixel set, denoted by p, consisting of all the j-th pixels of I^s. (2) Image reconstruction (lines 2–7): we recover s by the continuous construction process of all pixels.

Comparison-based region detection. The puzzle region can be detected through a comparison between the background image and its corresponding source image; i.e., a simple XOR operation can be used to detect the region. Figure 8 illustrates the process of puzzle region detection using an XOR operation on the background image and its source image.

5.1.2 Human behavior simulation

Some slide-based schemes detect malicious behaviors (e.g., a rapid and direct movement to the puzzle region) that they consider to be machine generated. To bypass such detection in SliAttack, we mimic human behaviors leveraging four simulation functions: Sigmoid (en.wikipedia.org/wiki/Sigmoid_function), Softmax (en.wikipedia.org/wiki/Softmax_function), ReLu (en.wikipedia.org/wiki/Rectifier_(neural_networks), and Tanh (brenocon.com/blog/2013/10/tanh-is-a-rescaled-logistic-sigmoid-function).

Let b denote the distance between the puzzle window and region. Let $D = \{D_i|i = 1, 2, \ldots, k\}$, where $\sum_{D_i \in D} D_i = b$ denote the length set of moving steps, and $T = \{T_i|i = 1, 2, \ldots, k\}$ denote the time set of moving steps.

To bypass the malice detection, we generate D and T as follows. Consider the Sigmoid function, $f(x) = \dfrac{1}{1 + \mathrm{e}^{-x}}$, as an example. We assign the length of each step as $D_i = b \times \left(\dfrac{1}{1 + \mathrm{e}^{-i/2+4}} - \dfrac{1}{1 + \mathrm{e}^{-(i-1)/2+4}}\right)$, where i is an integer and $1 \leqslant i \leqslant k$. Note that, to meet the constraint that $\sum_{D_i \in D} D_i = b$, we set $D_1 = b \times \dfrac{1}{1 + \mathrm{e}^{-1/2+4}}$ and $D_k = b \times \left(1 - \dfrac{1}{1 + \mathrm{e}^{-k/2+4}}\right)$. We randomly shuffle D to get the final sequence of moving steps. For T, we randomly generate the moving time between each moving step.

The working mechanisms of the other three functions, Softmax, ReLu, and Tanh, are similar to that of Sigmoid.

5.1.3 Design of SliAttack

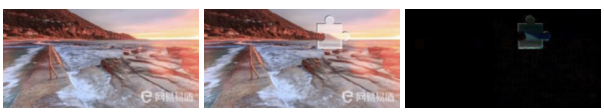
Based on the workflow of slide-based captchas, we design our attack, as illustrated in Algorithm 3. We collect a set of captcha challenges from the target scheme and use Algorithm 2 to recover the set of source images. This process is mainly used to bootstrap our attack. Afterwards, we can automatically solve each real-world captcha from the target scheme. When receiving a captcha challenge, we first extract the background image, and find its corresponding source image. Then, we locate the puzzle region through a comparison between the background image and the source image. Next, we mimic human behaviors in sliding the puzzle to the detected puzzle region. The pipeline of our attack is shown in Fig. 9.

5.2 Evaluation of SliAttack

To evaluate SliAttack's effectiveness against slide-based captchas, we conduct a series of experiments on

Algorithm 2 Source image recovery

Input: I^s
Output: s
1 Initialize $s \leftarrow \varnothing$
2 **for** $j \in \{1, 2, \ldots, n\}$ **do**
3 $p \leftarrow \varnothing$
4 **for** $i \in \{1, 2, \ldots, m\}$ **do**
5 $p \leftarrow p \cup I_i^s(j)$
6 candidate $\leftarrow$ the most frequent value in p
7 $s \leftarrow s \cup$ candidate



(a) Source image (b) Backgournd image (c) Detected region

Fig. 8 The process of puzzle region detection.

Algorithm 3 SliAttack

Input: The slide-based captcha I and the set of collected historical captcha images I^s
Output: The captcha solution S.
1 $B \leftarrow$ **source_image_recovery**(I^s)
2 $b \leftarrow$ **extract_background_image**(I)
3 $s \leftarrow$ **find_source_image**(b, s)
4 $r \leftarrow$ **localize_puzzle_region**(b, s)
5 $S \leftarrow$ **mimic_human_behavior**(r, b)
6 **return** S

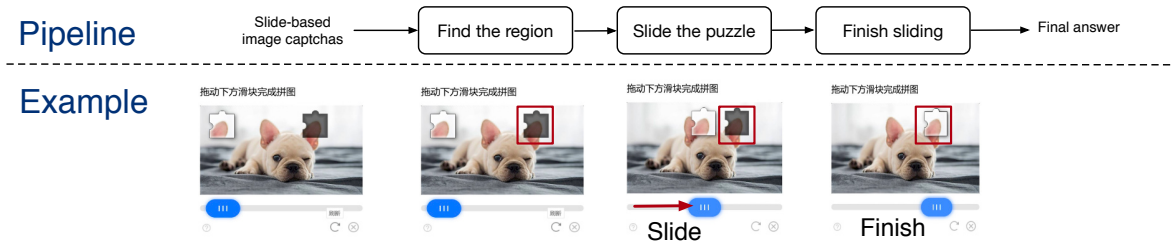


Fig. 9 Attack pipeline for slide-based image captchas.

three different schemes, including GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle.

Setup. To bootstrap the attacks against Tencent SlidePuzzle and Netease SlidePuzzle, we prepare a source image datasets for each. For GEE SlidePuzzle, no bootstrapping is required since its source images can be directly obtained from the received challenges.

We then evaluate SliAttack using six different movements for the slide: direct, random, Sigmoid, Softmax, Tanh, and ReLu. Note that we evaluate the direct and random movements only to determine whether the target scheme employs any malice detection strategy.

Further, we compare our attack with human labor. We hire human labor from three captcha-solving services, i.e., ruokuai, hyocr, and dama2. We select these services based on their popularity and their support for slide-based captchas.

5.2.1 Attack results

We perform our attack against slide-based image captchas as follows. First, if possible, we recover source images for the tested schemes to bootstrap our attack. Of the three tested schemes, only Tencent SlidePuzzle and Netease SlidePuzzle require the process of source image recovery. Specifically, we recover 10 source images from 2000 pre-downloaded challenges for Tencent SlidePuzzle, and 11 source images from 2000 pre-downloaded challenges for Netease SlidePuzzle. We then run our attack using each of the six movement settings against all of the schemes. To minimize our impact on real systems, we limit our attack to 200 live online challenges per scheme.

Table 7 summarizes our attack's success rate (the fraction of attempts at breaking the slide-based captcha challenges that are successful) and its average speed on each scheme. In the remainder of this section, we discuss the effectiveness of our behavior simulator and the success rate and speed of our attack method.

Effectiveness of the behavior simulator. We can observe from Table 7 that our attack with movement based on the Sigmoid function has the highest success rate on all schemes. This result suggests that SliAttack's behavior simulator is effective in practice.

GEE SlidePuzzle is the most robust of the three tested schemes. On GEE SlidePuzzle, our attack achieves the highest success rate of 0.96 when using the Sigmoid function, while the success rate decreases significantly when we use other functions.

We find that Tencent SlidePuzzle probably lacks any mechanism for malice detection. Our attack has a 100% success rate even when we directly move the slide puzzle to the puzzle region. We conjecture that this is a design flaw in Tencent SlidePuzzle, and this conjecture is confirmed by Tencent after we report to it with our

Table 7 Attack results on slide-based image captchas.

Method		GEE SlidePuzzle		Tencent SlidePuzzle		Netease SlidePuzzle	
		Success rate	Speed (s)	Success rate	Speed (s)	Success rate	Speed (s)
Our method	Sigmod	0.96	5.30	1.00	4.01	0.98	1.98
	Softmax	0.59	5.27	0.95	4.18	0.72	2.15
	Tanh	0.00	5.16	1.00	4.06	0.98	2.24
	ReLu	0.54	5.68	0.99	4.27	0.54	5.68
	Random	0.16	5.33	0.97	4.33	0.81	2.35
	Direct moving	0.00	2.37	1.00	0.88	0.00	1.71
Captcha-solving services	ruokuai	0.88	8.82	0.96	7.94	0.91	6.06
	hyocr	0.93	9.69	0.92	5.73	0.87	7.71
	dama2	0.91	11.03	0.97	6.13	0.95	8.17

findings.

Success rate. Our attack's success rates are all above 0.96, and the highest success rate reaches 1. Such a high success rate not only indicates the effectiveness of our attack, but also reveals the vulnerabilities of real-world slide-based captcha schemes.

Speed. On average, it takes between 1 and 6 seconds for our attack to break each of the schemes. The fastest speed is achieved against Tencent SlidePuzzle, which takes about 1 second to break. The slowest speed is on GEE SlidePuzzle, which takes nearly 5 seconds — still very fast as compared to the time taken by human users (about 30 seconds). Tencent SlidePuzzle takes significantly less time to break than the other schemes because it does not inspect the slide trajectory, therefore our attack can directly slide the puzzle window to the puzzle region. For GEE SlidePuzzle and Netease SlidePuzzle, our attack randomly stops and waits for 1 – 2 seconds in order to evade the malice detection.

5.2.2 Comparison with human labors

Next, we evaluate the attack results of human labor from ruokuai, yundama, and dama2, the three largest captcha-solving services in China. Due to budget limits, for each service, we submit 300 challenges with 100 per scheme. Table 7 shows the success rate and average speed of proficient human labor.

After comparing our attack with human labor, we come up with the following two findings: (1) proficient human labor fails on all captcha schemes to achieve a better success rate than that of our attack; and (2) proficient human labor requires an average of 7 – 10 seconds to solve the captchas, which is much slower than our attack. We conjecture that the reason why human labor is less effective is that the measurement errors produced by human labor can significantly affect the positioning accuracy of the puzzle region, leading to an incorrect solution.

Again, from the comparison with human labor, we conclude that SliAttack is highly effective and that the currently common practice of slide-based captchas however is inadequate.

5.3 Design flaws

Based on our evaluation results, we summarize the following design flaws in slide-based captchas. First, most schemes repeatedly use the same source images to generate challenges, which makes it easy for an adversary to locate puzzle regions. Second, the malice detection methods used by the tested real-world schemes are not an effective defense against adversaries, and one such scheme does not even employ a detection algorithm.

6 Security of Click-Based Image Captchas

In this section, we introduce the design of the *CliAttack* method for attacking click-based captchas. We then evaluate CliAttack against three popular real-world captchas. Based on our evaluations, we discuss several design flaws in existing click-based captchas.

6.1 CliAttack

Click-based captchas ask a user to sequentially click the distorted characters drawn in the background image matching their appearance in the hint. Intuitively, advanced deep learning techniques can be adopted to detect the semantic regions of distorted characters.

6.1.1 Notations

A click-based captcha consists of two parts: an image hint and a background image. Similar to selection-based captchas, the hint can come in two formats: as a text hint and as an image hint. The characters contained in the hint are also drawn on the background image.

6.1.2 Design of CliAttack

The design principle of CliAttack is shown in Fig. 10, while Algorithm 4 gives the pseudo code. There are four steps to the CliAttack procedure. (1) To bootstrap our attack, we pre-train a character recognition model for recognizing distorted characters from both the hint and the background image. We also pre-train a character detection model on a dataset of captcha challenges with annotated semantic regions, which is collected from the target captcha scheme. (2) Upon receiving a challenge, we extract the hint from the HTML DOM

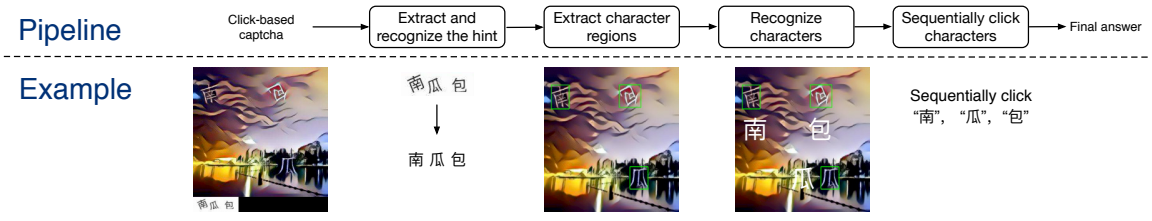


Fig. 10 Attack pipeline for click-based image captchas.

Algorithm 4 **CliAttack**

Input: The click-based captcha I.
Output: The captcha solution S.

```
h ← extract_hint(I)
if h is an image then h = recognize_hint_image(h)
b ← extract_background_image(I)
R ← localize_semantic_regions(b)
L ← recognize_semantic_regions(R)
S ← find_right_regions(h, R, L)
return S
```

elements and, if necessary, use the character recognition model to recognize the hint's distorted characters. (3) We then locate potential semantic regions leveraging the pre-trained character detection model. The semantic meanings of those regions are also recognized by the character recognition model as well. (4) After comparing the potential semantic regions with the hint, we sequentially click the correct semantic regions, which are made up of distorted characters drawn on background images.

6.2 Evaluation of CliAttack

To evaluate CliAttack, we run a set of experiments against three real-world schemes: GEE TouClick, Tencent TouClick, and Netease TouClick.

Setup. To bootstrap the attack, we collect and manually label three datasets, and use these labeled datasets to train three models to detect and recognize distorted characters. Equipped with the pre-trained models, we then run our attack against live online captchas.

For a comparison with online recognition services, we run five Optical Character Recognition (OCR) services, including BaiduOCR, TencentOCR, GoogleOCR, AliOCR, and Face++OCR. We select these five OCR services since they are reported to rapidly and accurately recognize distorted characters drawn on images.

For a comparison with human labor, we evaluate the attack results of three captcha-solving services: ruokuai, yundama, and dama2. We choose these three captcha-solving services due to their popularity in the underground captcha-solving market and their support for click-based captchas.

6.2.1 Data collection

For breaking GEE TouClick, Tencent TouClick, and Netease TouClick, we collect a total of three labeled datasets: D_6, D_7, and D_8 as illustrated in Table 4. These datasets are then used to train object detection and character recognition models. Below, we detail the basic statistics of these datasets.

- D_6. D_6 is a synthetic dataset of distorted characters, which contains 5 257 000 images of 3755 commonly used Chinese characters. In D_6, nearly every 1400 distorted images of a single character is generated by 16 different font generation algorithms. D_6 is mainly used for training a CNN model to recognize distorted characters.
- D_7. D_7 consists of 2000 captcha challenges collected from GEE TouClick, with the regions of each distorted character manually annotated. D_7 is used to train a Fast-RCNN model to locate distorted characters drawn on GEE TouClick challenges.
- D_8. Similar to D_7, D_8 contains 2000 captcha challenges collected from Tencent TouClick, with the regions of each distorted character manually annotated. D_8 is used to train a Fast-RCNN model to locate distorted characters drawn on Tencent TouClick challenges.

Both Netease TouClick and Netease SlidePuzzle use the same set of images to generate their challenges, as we discovered during our attack against slide-based captchas. Hence, we can directly detect distorted characters through a comparison between the captcha challenge and its source image.

6.2.2 Attack models

We train three models for breaking the tested schemes: CNN_6, Fast-RCNN_1, and Fast-RCNN_2. CNN_6 is further used for character recognition on all schemes, Fast-RCNN_1 is used for character detection on GEE TouClick, and Fast-RCNN_2 is used for character detection on Tencent TouClick.

CNN_6 is trained on an ubuntu server with two Inter-Xeon E5-2640V4 CPUs, a GTX 1080Ti GPU, and 128 GB memory. Fast-RCNN_1 and Fast-RCNN_2 are trained on an ubuntu server with an Intel i5-7500 CPU, a GTX 1060 GPU, and 16 GB memory. All of the models are trained using the standard five-fold cross validation.

Below, we detail the training process and results of the three models.

- **CNN_6**. We train CNN_6 on D_6 with a batch size of 16 and a learning rate of 1×10^{-4}. This training process lasts for 17 hours and achieves an accuracy of 0.9986.
- **Fast-RCNN_1**. We train Fast-RCNN_1 on D_6 with a batch size of 16 and a learning rate of 1×10^{-4}. This training process lasts for 7 hours and achieves an accuracy of 0.9201. Note that we validate the running

results through manually inspecting the regions of distorted characters.

• **Fast-RCNN$_2$**. We train Fast-RCNN$_2$ on D_7 with a batch size of 16 and a learning rate of 1×10^{-4}. This training process lasts for 12 hours and achieves an accuracy of 0.9712. Note that, similar to Fast-RCNN$_1$, we validate the running results by manually inspecting the regions of distorted characters.

6.2.3 Attack results

We evaluate our attack against real-world online captchas from GEE TouClick, Tencent TouClick, and Netease TouClick. To minimize our impact on live systems, we run our attack against 200 real-world challenges from each scheme. Table 8 summarizes the results of our attack, where the success rate is the fraction of attempts at breaking the captcha challenges that are successful.

Success rate. Our attack's success rates are all above 0.46, with the highest success rate of 0.74 achieved on Tencent TouClick. These results suggest that CliAttack is effective in practice.

From Table 8, we can also observe that the most challenging scheme is GEE TouClick, on which our attack's success rate is 0.46. We conjecture that the following two reasons make GEE TouClick very challenging: (1) the similarity in color of the distorted characters and background images adds to the difficulty in locating distorted characters; and (2) the distorted characters might be generated by a large number of different font generation algorithms, so given that our CNN$_6$ character recognition model is trained on distorted characters from a limited number of fonts (16), it is reasonable that our attack loses some accuracy on GEE TouClick.

Speed. Our attack's duration ranges from 4.13 to 4.78 seconds, which is relatively fast given that a common usability requirement is to demand a user to solve a captcha within 30 seconds. We note that the solving time of all schemes includes a network delay overhead estimated at 3 seconds per captcha challenge. The speed of our attack suggests that it poses a realistic threat to all of these schemes.

6.2.4 Comparison with online image services

There are several online services and libraries that offer character recognition functionality, and transform distorted characters into machine-encoded text. Hence, we also leverage these online services to evaluate the security of click-based image captchas. Specifically, we utilize five widely used character recognition services: BaiduOCR, TencentOCR, GoogleOCR, AliOCR, and Face++OCR. For each online service, we test 300 challenges with 100 per scheme.

Table 8 summarizes the results of our evaluation of these services. The success rate ranges from 0.02 to 0.51, which is relatively low when compared to that of CliAttack. In terms of speed, it takes between 5.70 and 12.15 seconds on average to break each captcha, of which a majority is made up of the network latency of the API calls.

The above analysis suggests that online OCR services are not particularly well suited for breaking click-based schemes, and a pre-trained character recognition model is usually more powerful for this task.

6.2.5 Comparison with human labor

Additionally, we compare our attack results with those of human labor from three captcha-solving services: ruokuai, hyocr, and dama2. For each captcha-solving service, we target 300 captchas with 100 per captcha scheme. Table 8 shows the success rate and speed of human solvers on the three tested captcha schemes.

The success rate of human labor is above 0.8 on all schemes, and their solving time ranges from 6.84 to 9.98 seconds. While the success rate of human labor

Table 8 Attack results on click-based image captchas.

Method		GEE TouClick		Tencent TouClick		Netease TouClick	
		Success rate	Speed (s)	Success rate	Speed (s)	Success rate	Speed (s)
Our method		0.46	4.63	0.74	4.78	0.69	4.13
Image recognition services	BaiduOCR	0.04	6.55	0.36	6.14	0.12	5.70
	GoogleOCR	0.05	13.37	0.27	11.22	0.03	12.15
	TencentOCR	0.02	6.09	0.51	6.53	0.07	6.41
	AliOCR	0.03	7.54	0.13	6.60	0.03	7.17
	Face++OCR	0.08	7.96	0.30	8.79	0.05	8.38
Captcha-solving services	ruokuai	0.81	9.47	0.91	7.09	0.89	8.04
	yundama	0.89	8.86	0.86	7.37	0.87	7.28
	dama2	0.85	9.98	0.90	6.84	0.94	9.11

is higher than that of our attack, the use of human labor is a much slower method. Moreover, the gap between the success rate of human labor and that of our attack is acceptable at less than 0.2.

6.3 Design flaws

Based on the evaluation results, we summarize the design flaws of click-based captchas as follows. The most serious design flaw is that some captcha providers, e.g., Netease, use the same image set to generate challenges for different schemes. An additional flaw is that the tested schemes do not perform any anti-recognition operations (e.g., rotation) on the distorted characters drawn on background images.

7 Captcha-Solving Services: An Underground Market

During our research, we find a huge profit-seeking underground market in captcha-solving services. To dissect this underground market, we analyze some statistics of captcha-solving services, including the number of services, their scale, and their lifecycle. We also investigate the entire underground market landscape, including the types of captcha-solving services being supplied and the scale of the demand. Then, we estimate the income accruing from these illegal services. Finally, we analyze their potential impacts on a variety of industries, including e-commerce, online ticketing services, and online advertising.

7.1 Statistics of captcha-solving services

7.1.1 Landscape

To find services that support resolving our tested captcha schemes, we design and implement two crawlers to collect captcha-solving services through related keyword queries. One, named *BaiduCrawler*, collects possible services from the search results of baidu.com; the other, called *BingCrawler*, utilizes bing.com to find services. Our crawlers work as follows. The crawlers first use the names of popular captcha-solving services (e.g., ruokuai, yundama, and AntiCaptcha) as keywords to query the search engine. Then, the crawlers extract the related keywords and the top sites (i.e., sites that are in the first page of the search results) returned by the search engine. These top sites are saved for later rule-based filtering, and the related keywords are used for subsequent queries. Our crawlers are designed to repeatedly query the search engine in order to collect a sufficient number of sites providing the service.

BaiduCrawler starts by querying two keywords, ruokuai and yundama, and BingCrawler starts with one keyword, AntiCaptcha. In total, we collect about 1000 candidate sites after filtering out irrelevant contents. We then manually check these sites, and finally confirm 152 captcha-solving services that are distributed worldwide.

The above data collection strategy might be incomplete and has some limitations. For example, the keywords chosen to bootstrap BaiduCrawler and BingCrawler might be insufficient because many more seeds could be employed, e.g., hyocr and DeCaptcha. Also, it would be possible to collect captcha-solving services from other sources, e.g., the underground forums and the external links from identified sites. There is a case for further dedicated research to design specific detection techniques for collecting captcha-solving services, which would aid in the comprehensive exploration and understanding of the underground market.

In addition to testing the robustness and security of image captchas, these 152 services can also be used to measure the captcha-solving market. The 152 identified captcha-solving services are located all over the world, as determined by the geolocations. We observe that most of the services are located in China (58%), followed by the United States (21%) and Russia (2%). This distribution is different from that given in the study of labor distribution presented in Ref. [28]. There are three possible reasons for this difference. First, we collect the geolocations of the service sites while Ref. [28] focuses on the geolocations of human laborers; it is possible that the geolocation distribution between captcha-solving services and laborers is different since captcha-solving services usually make extensive use of labor from foreign countries with abundant labor and low labor costs. Second, it has been eight years since the study in Ref. [28] was conducted, and the geographical distribution both of the captcha-solving services and of the human laborers is likely to have changed in this time. Third, the data collection method used in this paper might be biased, and to some extent this could explain the inconsistency of the findings. However, since our primary focus in this paper is not to measure the underground market of captcha-solving services but rather to evaluate the security of real-world image captchas, the 152 services that we identify are

sufficient for our analysis.

7.1.2 Estimated number of laborers

Figure 11 illustrates the number of laborers working for the popular captcha-solving service platforms used in our work. Of these services, most (70%) have less than 4500 laborers, while the remaining (30%) have about 8000 laborers. We also estimate the average number of laborers per service at about 3700; this number is estimated from the average numbers of service calls and the claimed number of laborers of a randomly selected 20% of captcha-solving services. It is a conservative estimate since some of the services, e.g., ruokuai, deliberately report a small number of laborers to avoid suspicion. Since there exist at least 152 active captcha-solving services, the total number of laborers in the underground market is estimated to be in excess of 562 400.

7.2 Landscape of captcha-solving services

To better understand the underground market, we briefly analyze the entire landscape of these captcha-solving services. To be specific, we survey the categories of captcha that the underground market supports and the potential customers for these services.

7.2.1 Supporting services

The underground market supports the solution of various types of captchas, including text captchas, image captchas, and video captchas. Table 9 shows five commonly used captcha types, representative schemes, and corresponding captcha-solving services. Among the 152 services, more than 90% support text and image captchas, especially ReCaptcha. For example, both 2captcha and AntiCaptcha claim to have sufficient human labor to solve ReCaptcha. Based on this analysis, we conjecture that the underground market for captcha-solving services is quite mature and can resolve every type of captcha.

Interestingly, we find that some underground

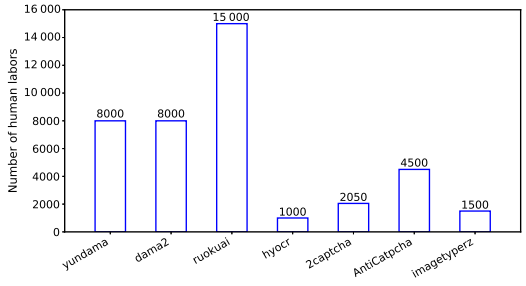


Fig. 11 Number of laborers on popular captcha-solving service platforms.

Table 9 Captcha solving services.

Captcha type	Representative scheme	Representative service
Text captcha	SinaWeibo①, ReCaptcha	ruokuai, 2captcha
Image captcha	China Railway, ReCaptcha	ruokuai, AntiCaptcha
Audio captcha	ReCaptcha	ruokuai
Game captcha	FunCaptcha②	AntiCaptcha, jsdati③
Math captcha	Bilibili④	ruokuai, jsdati

Notes: ① SinaWeibo, https://weibo.com.
② FunCaptcha, https://www. funcaptcha.com/.
③ jsdati, https://www.jsdati.com/.
④ Bilibili, https://live. bilibili.com.

services, e.g., jsdati and ruokuai, deny that they provide captcha-solving services. Instead, as shown in Fig. 12, they claim on their websites to provide other services, including image recognition, advertising identification, and porn identification.

7.2.2 Demand for services

The underground market has a great many potential customers, including account enumeration attackers, third-party services, malicious promotion apps, and coupon stealers, etc. This equates to a huge market demand for underground captcha-solving services. Below, we analyze three found representative customer types and their demands on captcha-solving services.

Agent apps. Agent apps act as agents for users, providing fee-paying services that help a user to accomplish some tedious tasks. The bots utilized by these applications generate a huge demand for captcha-solving services. A recent news report claims that, during the spring festival in China, 43% of all online reservations made on the railway ticketing system, 12306.cn, might come from agent apps (www.xinhuatone.com/zt/12306/dagaozi-1/). According to this report, as many as 1000 million tickets might be purchased by agent apps, given

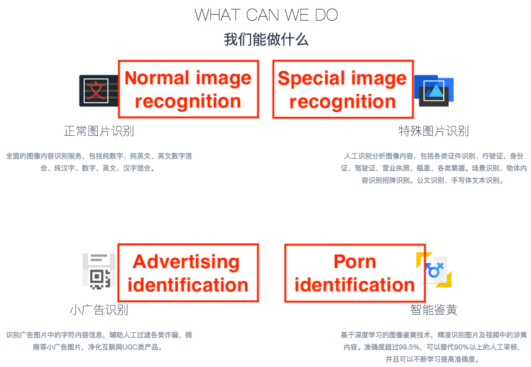


Fig. 12 ruokuai's denial of providing captcha-solving services.

that 12306.cn's sales volume in the spring festival is 2345 million. We can therefore make a rough estimate that the demand for captcha-solving service reaches 1000 requests in a 15-day period over the duration of the Chinese spring festival. This analysis suggests that, in the case of agent apps, there is a O(billion)-scale market demand for underground captcha-solving services. Moreover, since a reservation costs \$5 per railway ticket according to Ctrip(trains.ctrip.com/TrainBooking/SearchTrain.aspx), the economic scale of the underground ticket reservations could reach \$5 billion.

Malicious promotion apps. Malicious promotion apps provide various fake online behaviors (e.g., fake visits, fake "likes", and fake purchases) to miscreant users, to create a false impression that their published content (e.g., microblog, video, and e-commerce items) is relatively popular. For example, the malicious app provided by akwxll.com, manipulates over 100 000 mobile phones to automatically visit pages featuring particular e-commerce items. Based on the assumption that each mobile phone receives one captcha per hour, akwxll.com's demand for captcha-solving is 2.4 million per day, or 0.9 billion per year. This is a conservative estimation, since receiving one captcha per hour is a comparatively low frequency.

Coupon stealer. Coupon stealers are miscreant users who gain an economic benefit by earning coupons through malicious methods, e.g., registering multiple new accounts to receive cash bonuses. Usually, they rely on automation using bots or other software to steal millions of coupons, which creates a demand for captcha-solving services. A recent news item reveals that police have arrested a criminal group that earned profits of over \$100 million by stealing coupons (news.ifeng.com/gundong/detail_2013_12/17/32204801_0.shtml). The direct captcha-solving demand of coupon stealers is hard to estimate. We can only conjecture from the O(500 billion)-scale e-commerce market that such demand is extremely large.

7.3 Economic income from underground services

The primary goal of underground captcha-solving services is to obtain a high economic benefit. While we cannot accurately determine the income of the entire market, we can make an approximate estimate for this income, based on a few representative service providers.

Table 10 summarizes the five keys of the eight representative captcha-solving services: the number of laborers, the average speed for solving each captcha, the price for resolving 1000 captchas, the number of service requests per day, and the estimated daily income. Of these items, the first four are provided directly by service providers on their websites in oder to market their services. Figure 13 shows the statistics given by 2captcha. The remaining item, estimated daily income, is calculated with an equation: Number of laborers $\times$ PPC $\times$ (7 $\times$ 24 hours/Speed) $\times$ (Price/1000). Note that we estimate the income of ruokuai using its reported request volume since its reported labor size is not sufficiently accurate. The daily income of ruokuai, yundama, and dama2 reaches O(million)-scale, and the income of the remaining services is slightly below 0.3 million. In addition to this statistics, we also estimate that the average daily income of other identified underground services is about 1.07 million from some other identified services. The whole daily income of the entire underground market can be estimated at over 162.64 million, given that there are at least 152 active captcha-solving services.

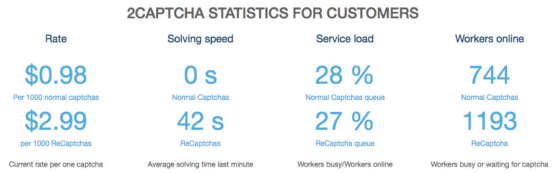


Fig. 13 2captcha's statistics of captcha-solving.

Table 10 Economic analysis on captcha-solving services. PPC = price per 1000 captchas, RPD = requests per day.

Name	Number of labors	Speed (s)	PPC (\$)	Number of RPD	Daily income (\$)
ruokuai	about 15 000	1 – 4	0.8 – 7.9	about 0.9 billion	about 3.6 million
yundama	about 8000	0 – 3	1.6 – 7.9	–	about 1.85 million
hyocr	about 1000	0 – 4	1.27 – 5.56	–	about 0.29 million
dama2	about 8000	0 – 3	0.63 – 29.37	–	about 1.62 million
2captcha	about 2050	12 – 50	0.97–2.99	–	about 0.012 million
AntiCaptcha	about 4500	about 8.3	0.63 – 2.20	–	about 0.10 million
DeCatpcha	–	–	about 2	–	–
imagetyperz	about 1500	9 – 45	1 $\sim$ 2.5	–	about 0.01 million

7.4 Impacts on benign industries and users

Finally, we discuss an example of the underground market's negative impact on benign industries and users.

Double 11 shopping carnival. Due to its promotions and huge sales volume, Alibaba's annual Double 11 shopping carnival is always a target for a great number of miscreant users. It is reported that the 2017 Double 11 shopping carnival attracted hundreds of thousands of miscreant users (www.tmtpost.com/2911463.html) obtaining coupons and bargain goods using the coupon stealer provided by hotniu.net. One individual among these miscreants was able to gain as much as $32 000 in economic benefits. Based on these statistics, the total economic loss for Alibaba could be as high as 3 billion. Besides this loss, these miscreant users negatively affected the shopping experience of honest users, who would have found it more difficult to acquire coupons or bargain-priced goods.

The huge economic loss suffered by Alibaba during its annual Double 11 shopping carnival can be partly blamed on the maturity of the underground captcha-solving market. It is the large-scale human labor provided by the underground market that allows abusive programs and coupon stealers to bypass captcha challenges issued by Alibaba.

8 Design Principle and Countermeasures

In this section, we distill our reflections on our own automatic captcha-breaking techniques, our evaluation of powerful online vision services, and our analysis of underground captcha-solving services into a set of best practices and design principles for website providers to design more secure captcha services.

Scalability of captcha corpus. Scalability measures the number of challenges a captcha scheme can generate without sacrificing its robustness and security. Among the 10 tested schemes, none of them is highly scalable because they either have a limited number of hint categories to enumerate, or their source images and candidate images are used repeatedly. Focusing on the scalability, we put forward three countermeasures and design principles for defending against our attacks.

(1) **Number of hint categories.** A large number of hint categories should be used; this means that it takes more time to enumerate the hint corpus, to collect sufficient datasets, and to train an accurate model. Hence, this countermeasure can slow down our attack against selection-based image captchas.

(2) **Size of source images.** The repeated use of any one single source image to generate challenges should be avoided. The best practice is to use a single source image for only once, which would provide adequate defense against our attack on slide-based captchas as our attack would be unable to detect the puzzle region. However, this strategy may also increase the security cost.

(3) **Number of candidate images.** A broad range of candidate images should be used, and candidate images should belong to categories that are excluded from the hint. This countermeasure might increase the number of labeling errors produced by pre-trained classification models, therefore reducing our attacks' success rate.

Risk analysis. Risk analysis should be performed on simple captcha scheme to evaluate the possibility that captcha solution has been derived by abusive programs. For Tencent SlidePuzzle, as an example, our attack will be mitigated if a risk analysis is performed on the slide trajectory.

Anti-recognition. To improve the security, anti-recognition techniques could be implemented in image captchas. We discuss four simple anti-recognition techniques, namely distorted text hints, distorted characters, random noise, and adversarial images.

(1) **Distorted image hint.** Distorted image hints should be preferred over text hints. For those captcha schemes that have already employed image hints, anti-recognition techniques should be applied to the hints. This countermeasure might make the hint more difficult to be recognized, and therefore reduces our attack's success rate.

(2) **Distorted character.** Anti-recognition techniques should be applied to distorted characters drawn on the background image. For those schemes that require a user to click specific semantic regions on the image, an anti-recognition technique (e.g., overlapping, rotating) may mitigate the threat posed by our attack.

(3) **Image noise.** Noise should be added to background images. To use the example of a slide-based image captcha, if we randomly add a deceptive empty region then our attack's success rate is reduced by half. Moreover, random noise on the background image of slide-based captchas can mitigate the threat, since it prevents our attack from recovering the source image.

(4) **Adversarial images.** Adversarial images should

be generated that are imperceptible to humans while fooling deep image classification models into producing incorrect recognition results. Not only can this countermeasure defend against our attack but it can also mitigate the captcha threat posed by the attacks based on online vision services. Figure 14 shows an example of this defense, where a dog is incorrectly recognized as flower after inserting elaborately crafted noises. The adversarial noise is generated by FGSM[31], with the noise level parameter ϵ set to 0.07 and the iteration step parameter set to 40.

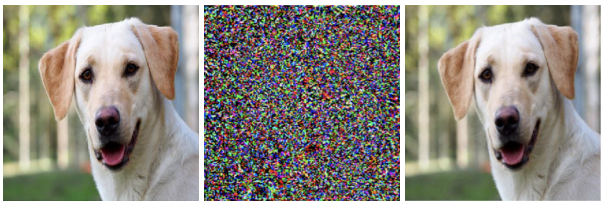
9 Discussion

Ethic issues. Most of the evaluation results and findings of this paper are made on datasets crawled from the public domain. While it was necessary to perform attacks against captchas, our attacks were designed to minimize the impact on the websites of captcha providers. Furthermore, we have not affected the websites in any way except for acquiring captcha challenges.

Furthermore, we have disclosed reports of our findings and recommendations to all of the captcha providers involved, in an effort to help them to make their captchas more robust to automated attacks. Only Tencent and Netease responded to our reports, and they also acknowledged our findings and recommendations. We hope that the disclosure of our findings will result in more robust captcha services.

Limitations. We believe our work developing and evaluating powerful captcha attack frameworks can be improved in many perspectives. We discuss some limitations of this work below, along with suggestions for future work.

First, we focus on the security of three categories of popular image captchas, and propose simple yet powerful attack frameworks. Also, we evaluate our attacks against 10 real-world captcha schemes, and reveal some of their design flaws. Although our research is useful and effective, it is limited in scope and it might therefore be useful to consider more captcha categories and schemes.



(a) Original image (b) Elaborately crafted noises (c) Adversarial image

Fig. 14 A defense example of adversarial image.

Second, these three attack frameworks still have the potential for improvement. None of the three attacks are fully automated, with each requiring preparatory steps (offline analysis) to train specific image classifiers or recover source images. We have not built a large captcha image corpus or design fully automated attacks. For SelAttack, we train the image classification model on a small labeled dataset of images. Therefore, the success rate of SelAttack could be improved by training a more accurate image classification model. For SliAttack, we have implemented four simulation functions that are very effective in bypassing the malice detection of real-world captchas. Among these, the most effective is Sigmoid, which achieves the highest success rate of over 0.96 on all the tested schemes. Nonetheless, additional simulation functions or other possible human behavior simulation methods could be devised to achieve better performance in bypassing the malice detection. For CliAttack, its success rate is limited by its Chinese character recognition accuracy. Therefore, our attack's success rate on click-based captchas could be improved by a more elaborate recognition model trained on a large-scale dataset of labeled distorted characters from a variety of fonts.

Third, to evaluate three attacks, we tested them against 10 representative real-world captchas. In practice, other schemes of selection-based, side-based, and click-based captchas may exist, and it would be useful to consider additional image captcha schemes and evaluate our attacks on them.

Fourth, our measurement study on the underground market is based on 152 identified captcha-solving services. This measurement methodology might not fully comprehensive or accurate. For example, we do not measure the labor market (e.g., www.zbj.com) supplying human labor to the underground market. The analysis of the geolocation distribution is based on a small set of identified services, which might not be representative of the entire market. Hence, we believe more dedicated research into the measurement of the underground market for captcha-solving services is required.

Future work directions. Our study reveals the vulnerability of currently popular captcha schemes. To mitigate the captcha threat, future work should be considered in the following three directions.

Malicious API call detection. Vision service

providers (e.g., Google, Microsoft, and Baidu) ought to make a risk analysis on the incoming API calls. This risk analysis may detect malicious API calls from miscreant users for a number of improper uses, e.g., labeling candidate images of captchas and recognizing distorted characters. Therefore, one future work direction is to propose and deploy a risk analysis system for online vision services.

Underground market detection. While the captcha threat posed by human attacks is difficult to defend against, we can turn to monitoring and detection of underground captcha-solving services, thereby mitigating the threat at its roots. Recently, Liao et al.[32] have developed a prototype system, *BarFinder*, to automatically detect malicious repositories by analyzing the topological features of online repositories. Motivated by *BarFinder*, another future research direction is to develop a detection tool that can automatically find underground captcha-solving services.

More secure authentication. Computer vision and machine learning techniques have advanced to a stage that makes existing schemes for the automated distinction of human users from bots seem fruitless. Much recent work focuses on various authentication methods, e.g., photo-based authentication schemes[33] and liveness detection-based authentication[24]. Motivated by Uzun et al.[24], who proposed a liveness detection captcha system called rtCaptcha, another future work direction is to design a robust captcha scheme based on voice authentication, video authentication, etc.

10 Conclusion

In this paper, we study the security issues facing popular real-world image captchas. To this end, we propose three proof-of-concept attacks against selection-based captchas, slide-based captchas, and click-based captchas. We evaluate our attacks against 10 popular real-world captcha schemes, provided by google.com, tencent.com, etc., and successfully break all of them. We also compare our attacks with two previous methods, nine online image recognition services, and eight captcha-solving services that employ human labor. The results show that our attacks pose a significant and realistic threat to various real-world image captchas. With the goal of aiding the design of more secure captchas, we distill our reflections on our attacks and our evaluation of recognition services and underground captcha-solving services into a set of best practices and design principles for website providers. We believe that our study in this paper will be highly useful for secure image captcha design.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China (Nos. 61772466 and U1836202), the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars (No. LR19F020003), the Provincial Key Research and Development Program of Zhejiang Province (No. 2017C01055), and the Alibaba-ZJU Joint Research Institute of Frontier Technologies.

References

[1] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, CAPTCHA: Using hard AI problems for security, in *Proc. 2003 Int. Conf. the Theory and Applications of Cryptographic Techniques*, Warsaw, Poland, 2003.

[2] M. Chew and J. D. Tygar, Image recognition CAPTCHAs, in *Proc. 7th Int. Conf. Information Security*, Palo Alto, CA, USA, 2004.

[3] K. F. Hwang, C. C. Huang, and G. N. You, A spelling based CAPTCHA system by using click, in *Proc. 2012 Int. Symp. Biometrics and Security Technologies*, Taipei, China, 2012.

[4] N. J. Hopper and M. Blum, Secure human identification protocols, in *Proc. 7th Int. Conf. the Theory and Application of Cryptology and Information Security*, Gold Coast, Australia, 2001.

[5] S. Sivakorn, I. Polakis, and A. D. Keromytis, I am robot: (Deep) learning to break semantic image CAPTCHAs, in *Proc. 2016 IEEE European Symp. Security and Privacy*, Saarbrucken, Germany, 2016.

[6] H. Q. Ya, H. N. Sun, J. Helt, and T. S. Lee, Learning to associate words and images using a large-scale graph, arXiv preprint arXiv: 1705.07768, 2017.

[7] G. Mori and J. Malik, Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA, in *Proc. 2003 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Madison, WI, USA, 2003.

[8] K. Chellapilla and P. Y. Simard, Using machine learning to break visual Human Interaction Proofs (HIPs), in *Proc. 17th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2004.

[9] E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell, A low-cost attack on a Microsoft CAPTCHA, in *Proc. 15th ACM Conf. Computer and Communications Security*, Alexandria, VA, USA, 2008.

[10] E. Bursztein, M. Martin, and J. C. Mitchell, Text-based CAPTCHA strengths and weaknesses, in *Proc. 18th ACM Conf. Computer and Communications Security*, Chicago, IL, USA, 2011.

[11] E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell, The end is nigh: Generic solving of text-based CAPTCHAs, in *Proc. 8th USENIX Conf. Offensive Technologies*, San Diego, CA, USA, 2004.

[12] H. C. Gao, J. Yan, F. Cao, Z. Y. Zhang, L. Lei, M. Y. Tang, P. Zhang, X. Zhou, X. Q. Wang, and J. W. Li, A simple generic attack on text captchas, in *Proc. 23rd Annu. Network and Distributed System Security Symp.*, San Diego, CA, USA, 2016.

[13] P. Golle, Machine learning attacks against the asirra CAPTCHA, in *Proc. 15th ACM Conf. Computer and Communications Security*, Alexandria, VA, USA, 2008.

[14] D. Lorenzi, J. Vaidya, E. Uzun, S. Sural, and V. Atluri, Attacking image based CAPTCHAs using image recognition techniques, in *Proc. 8th Int. Conf. Information Systems Security*, Guwahati, India, 2012.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proc. 25th Int. Conf. Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2012.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014.

[17] S. Q. Ren, K. M. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in *Proc. 28th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2015.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, SSD: Single shot multibox detector, in *Proc. 14th European Conf. Computer Vision*, Amsterdam, Netherlands, 2016.

[20] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in *Proc. 2015 IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015.

[21] J. Elson, J. R. Douceur, J. Howell, and J. Saul, Asirra: A CAPTCHA that exploits interest-aligned manual image categorization, in *Proc. 14th ACM Conf. Computer and Communications Security*, Alexandria, VA, USA, 2007.

[22] D. Misra and K. Gaj, Face recognition CAPTCHAs, in *Proc. 2006 Advanced Int. Conf. Telecommunications and Int. Conf. Internet and Web Applications and Services*, Guadelope, French, 2006.

[23] J. Kim, J. Yang, and K. Wohn, AgeCAPTCHA: An image-based CAPTCHA that annotates images of human faces with their age groups, *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 3, pp. 1071–1092, 2014.

[24] E. Uzun, S. P. H. Chung, I. Essa, and W. Lee, rtCaptcha: A real-time CAPTCHA based liveness detection system, in *Proc. 25th Annu. Network and Distributed System Security Symp.*, San Diego, CA, USA, 2018.

[25] D. Lorenzi, J. Vaidya, S. Sural, and V. Atluri, Web services based attacks against image CAPTCHAs, in *Proc. 9th Int. Conf. Information Systems Security*, Kolkata, India, 2013.

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[27] R. Girshick, Fast R-CNN, in *Proc. 2015 IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015.

[28] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, Re: CAPTCHAs: Understanding CAPTCHA-solving services in an economic context, in *Proc. 19th USENIX Conf. Security*, Washington, DC, USA, 2010.

[29] Y. Shin, M. Gupta, and S. A. Myers, The nuts and bolts of a forum spam automator, in *Proc. 4th USENIX Conf. Large-Scale Exploits and Emergent Threats*, Boston, MA, USA, 2011.

[30] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, ImageNet: A large-scale hierarchical image database, in *Proc. 2009 IEEE Conf. Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009.

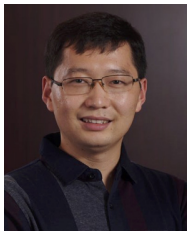
[31] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv: 1412.6572, 2014.

[32] X. J. Liao, S. Alrwais, K. Yuan, L. Y. Xing, X. F. Wang, S. Hao, and R. Beyah, Lurking malice in the cloud: Understanding and detecting cloud repository as a malicious service, in *Proc. 2016 ACM SIGSAC Conf. Computer and Communications Security*, Vienna, Austria, 2016.

[33] I. Polakis, P. Ilia, F. Maggi, M. Lancini, G. Kontaxis, S. Zanero, S. Ioannidis, and A. D. Keromytis, Faces in the distorting mirror: Revisiting photo-based social authentication, in *Proc. 2014 ACM SIGSAC Conf. Computer and Communications Security*, Scottsdale, AZ, USA, 2014.

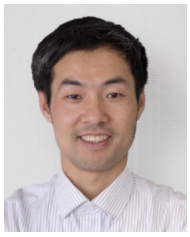
Haiqin Weng is currently a PhD student in the College of Computer Science and Technology, Zhejiang University, China. She received the BS degree from South China University of Technology in 2014. Her research interests include data mining and data security.

Binbin Zhao is currently a research assistant at Zhejiang University. He received the BS degree from the School of Computer Science at Zhejiang University in 2018. His research interest includes IoT security, CAPTCHA, and Adversarial Learning.

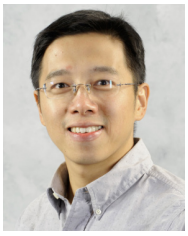


Shouling Ji is a ZJU 100-young professor in the College of Computer Science and Technology at Zhejiang University and a research faculty in the School of Electrical and Computer Engineering at Georgia Institute of Technology. He received the PhD degrees from Georgia Institute of Technology and Georgia State University in 2013 and 2015, respectively. His current research interests include big data security and privacy, big data driven security and privacy, and adversarial learning. He also has interests in graph theory and algorithms and wireless networks. He is a member of IEEE and ACM and was the membership chair of the IEEE Student Branch at Georgia State (2012–2013).

Qinming He is a professor in the College of Computer Science and Technology at Zhejiang University, China. He received the BS, MS, and PhD degrees all in computer science from Zhejiang University in 1985, 1988, and 2000, respectively. His research interests include data mining and computing virtualization.



Jianhai Chen is currently an assistant professor in the College of Computer Science and Technology at Zhejiang University. He received the BS degree from Hunan University in 1997, and the MS and PhD degrees from Zhejiang University in 2005 and 2016, respectively. His research interests include blockchain, virtualization, and cloud computing. He is a member of IEEE and ACM.



Ting Wang is an assistant professor of Lehigh University. He is also affiliated with Data X, an interdisciplinary initiative that pushes the envelope of data analytics research. Prior to joining Lehigh, he was a research staff member and security analytic leader at IBM Thomas J. Watson Research Center. He received the PhD degree from Georgia Institute of Technology in 2011. His current research focuses on computational privacy, cyber-security analytics, and network science.

Raheem Beyah is the Motorola Foundation Professor and Associate Chair for Strategic Initiatives and Innovation in the School of Electrical and Computer Engineering at Georgia Tech. Prior to returning to Georgia Tech, he was an assistant professor in the Department of Computer Science at Georgia State University, a research faculty member with the Georgia Tech CSC, and a consultant in Andersen Consulting's (now Accenture) Network Solutions Group. He received the bachelor degree from North Carolina A&T State University in 1998. He received the master and PhD degrees in electrical and computer engineering from Georgia Tech in 1999 and 2003, respectively. His research interests include network security, wireless networks, network traffic characterization and performance, and critical infrastructure security.