Relacha: Using Associative Meaning for Image Captcha Understandability

Songjie Wei[1(✉)], Qianqian Wu[1], and Milin Ren[2]

[1] School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing 210094, China
{swei,qqwu}@njust.edu.cn
[2] Beijing Engineering Research Center of NGI and Its Major Application
Technologies Co. Ltd., Beijing 100084, China
rmilin@163.com

Abstract. Text-based CAPTCHA has been used over decades with increasing difficulty to remain effective with OCR technique advance. Image-based CAPTCHA is supposed to step in as a better alternative. However, recognition-based image CAPTCHA is not robust enough to resist against either computer pattern recognition algorithms or brute-force attacks with exhaustivity approach. We present a new CAPTCHA design called Relacha to distinguish humans from bots by an image content correlation test. The new construction scheme adopts random walk among images with correlated contents, and utilizes human reasoning ability on inferring the relevance of images. Relacha challenges are generated dynamically by using images from real-time online search engine. The usability and robustness of the proposed scheme has been evaluated by both numerical analysis and empirical evidence. The results show that humans can solve Relacha conveniently and effectively with a high pass rate, while bot programs may succeed with slim chance.

Keywords: CAPTCHA · Web authentication · Image correlation
Associative meaning · Random walk

1 Introduction

CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) is a type of interactive computer challenge based on Artificial Intelligence (AI) problems that cannot be easily solved by current computer programs or bots, but is effortlessly solvable by humans [1]. CAPTCHA has been widely deployed for preventing malicious users from conducting automated network attacks and resource abuse, such as Denial-of-Service (DoS), which may lead to fatal exhaustion of server resources. Widely-used existing CAPTCHA techniques are either text-based or image-based ones, all request a match between the challenge and the inputted or selected answers to pass the test.

Text-based CAPTCHA as the most popular ones in current Internet, has been adopted for decades [2]. However, in order to survive along the continuously improved Optical Character Recognition (OCR) capability, text-based CAPTCHA challenges

G. Wang et al. (Eds.): SpaCCS 2017, LNCS 10656, pp. 353–367, 2017.
https://doi.org/10.1007/978-3-319-72389-1_29

have become extremely distorted and complex, which blocks not only computer programs but also human users from recognizing text-based CAPTCHAs. Instead, image-based CAPTCHA has been introduced as an alternative for better user experience, which relies on human ability to understand visual representations, such as distinguishing images of animals or objects. The gap between human and computational ability in recognizing visual content has been termed by Smeulders et al. as the semantic gap [3]. However, recent advance in pattern recognition foresees a promising migration of this gap between human and machine performance [4], which threatens almost all of the existing match-based CAPTCHA techniques. Furthermore, most of the existing image-based CAPTCHAs typically rest on a fixed collection of images, so an exhaustive attack by automated bots is possible [5].

By further surveying the machine capability in natural language and image processing technology, we find that inferring relevance between images is still an AI-hard problem in today's technology. Recent advance in AI image processing and indexing has shown that AI algorithms can effectively index and compare images for similarities without understanding the image contents. Relacha avoids using *is-a* or *similar-as* metric when presenting CAPTCHA questions and choices, but mimics human divergent thinking to utilize the underlying associative meanings in images to pair questions with correct answers in CAPTCHA test construction and result evaluation. Understandability of such associative meanings among images more relies on context, culture and history owned by humans than in memory, calculation specialized by computers. For example, an image of "suitcase" reminds humans more about "hotel" than "food store". When putting all together as CAPTCHA images, humans tend to pass the test by choosing the more closely associated pair.

We propose to exploit human's reasoning ability on image semantic correlation to create an image-based relevance-oriented CAPTCHA named Relacha (Relevance-oriented CAPTCHA) as a reformation of the existing match-based CAPTCHA. Relacha is constructed based on the correlation and dependency of word-annotated images from real-time online search engine. The correlation degree of word tags is measured based on their frequency of occurrence in Internet contents. Challenge words are visualized as images in Relacha construction. Answer images are retrieved from the Internet on-the-fly with tags as keywords to generate a dynamic resource library. Relacha challenge is presented with a visualized text question and a layout of multiple image choices, which are selected by randomly walking through the semantic relevance graph and retrieved online to avoid regular patterns.

Human involved experiments show that humans can solve Relacha tasks with high accuracy and efficiency, by quickly recognizing the relevance across images, where computer programs constantly fail.

Following we first summarize the up-to-date related image CAPTCHA solutions briefly in Sect. 2, with analysis on the security and robustness limitations of each. We present and explain the design details of Relacha in Sect. 3. The proposed new scheme is evaluated and validated in Sect. 4 with experiments and results, which are followed by a conclusion of the paper contributions in Sect. 5.

2 Related Works

As an alternative to text, latest CAPTCHA applications utilize image classification or recognition tasks as part of the challenge, as the examples shown in Fig. 1. ESP-PIX is an image-based solution in which a collection of images are displayed, and users are requested to select a correct description from a predefined list of categories. Another image-based CAPTCHA, KittenAuth, uses a fixed database to present images of cats to users [6]. These image-based CAPTCHAs are vulnerable to brute-force attack due to their static and fixed collection of images and descriptions.

(a) KittenAuth

(b) ESP-PIX

Fig. 1. Examples of image-based CAPTCHAs.

The "no captcha reCaptcha" developed by Google analyzes various aspects of a user's interaction with a displayed captcha and calculates a confidence score, then returns CAPTCHA challenges at different difficulty levels. For lower scores, a user may be presented with an image-based challenge as in Fig. 2, in which users are required to select proper images with similar content from the question image. ReCaptcha system

has been widely used. However, Sivakorn et al. [7] design a novel attack for image-based CAPTCHAs that extract semantic information from images. By using image annotation services and libraries, the attack approach is capable of identifying the content of images and selecting those depicting similar objects.

Fig. 2. A cat test of no captcha reCaptcha.

So far, all these CAPTCHA challenges are typically composed of questioning keywords and image choices. CAPTCHA answers are usually another representation of the questioning keywords or challenges, i.e. the correct answers match the question with identical semantic meaning in content. This is vulnerable to pattern recognition based image understanding, indexing, and matching. Therefore, using harder AI problems for web security is necessary.

Zhu et al. introduce a new security primitive based on hard AI problems, namely, a novel family of graphical password systems integrating Captcha technology, called CaRP (Captcha as gRaphical Passwords) [8]. CaRP is click-based graphical passwords and the sequence of clicks on an image is used to derive a password.

To avoid attackers collecting and recording the passwords, Catuogno and Galdi propose a graphical password scheme replacing static graphical challenges with on-the-fly edited videos [9]. The approach shows users a short film containing a number of pre-defined events and the users have to recognize such events, such as actions or concepts within a sequence of short videos. The graphical password utilizes human ability on recognizing the "meaning" of an object instead of its shape in a video.

Yang et al. focus on the ability to solve games, which is also one of the most advanced human cognitive process abilities. They propose GISCHA, a new way to create CAPTCHA using game-based image semantics [10]. The GISCHA challenge

can be easily operated by using only simple arrow keys, mouse movements, gestures and accelerometer. Figure 3 shows a GISCHA using the simple rolling ball game. The GISCHA challenge is to move the ball to the destination hole shaped as a circle.

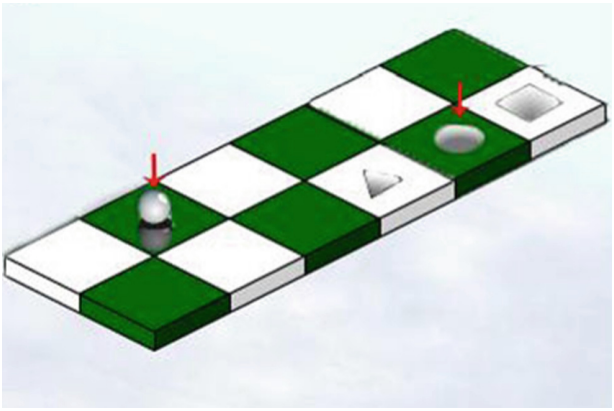

Fig. 3. A rolling ball game in GISCHA

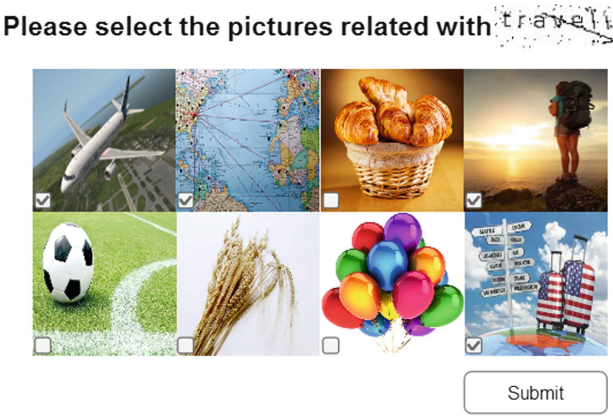

Fig. 4. An example of Relacha.

In this paper, we use the correlation other than equality, matching or similarity to improve the strength of image CAPTCHA, and design a novel CAPTCHA solution based on the relevance of associative meanings in images and the sequence of user clicks on images.

3 Relacha System

Now we present the overall system design of the Relacha CAPTCHA with detailed description about the Relacha construction and validation schemes.

A Relacha test consists of four procedures. The first procedure is to create a lexicon of image tags retrieved by web crawlers from hot words lists online. The second procedure generates a semantic relevance graph which depicts the correlation of all image tags provided by the lexicons retrieved. The third procedure produces and presents the test challenge by randomly walking through the semantic relevance graph. The last procedure is to receive and evaluate a user's choices according to an image associative meaning scoring algorithm, and to decide whether this user passes the Relacha test or not. Each of the four procedures of is designed with a specific goal that makes the entire framework easy to implement and use, and guarantees the generated CAPTCHA challenge robust against random attacks and exhaustive attacks.

3.1 Relacha Formation

Each Relacha challenge is composed of a question text and a grid of 8 choice images, as the example shown in Fig. 4, in which a user is challenged to choose images with content meanings associated with the question keyword "traveling".

3.2 Lexicon Creation

We first create a lexicon of image tag seeds, which can be automatically obtained, maintained, and refreshed. The lexicon consists of hot words online. First, we grab hot words as seeds from the Internet using web crawlers. Then we crawl and identify words correlated to those seed words according to the query suggestions from the public search engines (such as Google and Bing). Next, we filter these words to add to the lexicon. We drop those less-frequently used words based on the number of results that the search engine returns when a word is retrieved. If the number of retrieval results is below a predefined threshold (depending on the search engine used), then the word is excluded from the lexicon. We tag every remained word with a proper PoS (Part of Speech). We keep words having actual semantics, such as nouns, verbs, adjectives, and append them to the lexicon.

3.3 Graph Generation

We quantize the correlation of word semantics to generate a semantic relevance graph. The correlation of any two words in the lexicon is measured by mutual information (MI) [11]. Mutual information is the correlation between variable's values and is a measure of how well a given variable can be predicted using a linear function of a set of other variables.

With two words W_i and W_j, we retrieve them individually to get online statistics denoted as $c(W_i)$ and $c(W_j)$ from the search engine, then search $W_i + W_j$ in order to get retrieval results denoted as $c(W_{i,j})$, search $W_j + W_i$ to get retrieval results denoted as $c(W_{j,i})$. Then we infer

$$c(W_i, W_j) = \left(c(W_{i,j}) + c(W_{j,i})\right)/2 \tag{1}$$

We calculate the mutual information of W_i and W_j with $c(W_i, W_j)$, $c(W_i)$ and $c(W_j)$, the mutual information is represented by $MI(W_i, W_j)$ and is calculated as

$$MI(W_i, W_j) = log_2 \frac{c(W_i, W_j) \times N}{c(W_i) \times c(W_j)} \tag{2}$$

where N represents the maximum of the retrieved results.

The correlation of each word pair is represented in a tuple of three attributes $(word_1, word_2, corr(word_1, word_2))$. The storage is organized as in Table 1.

Table 1. Correlation storage format.

$Word_1$	$Word_2$	Degree of correlation
W_i	W_j	$MI(W_i, W_j)$
.....	.....	.....

A connected undirected graph $G = (V, E)$ is further constructed with n nodes and m edges. The nodes V are the collection of word tags of CAPTCHA images in the lexicon. Each edge e corresponds to relevance tags, with weights given by degrees of correlation denoted as C_{ij}, associating node v_i with v_j.

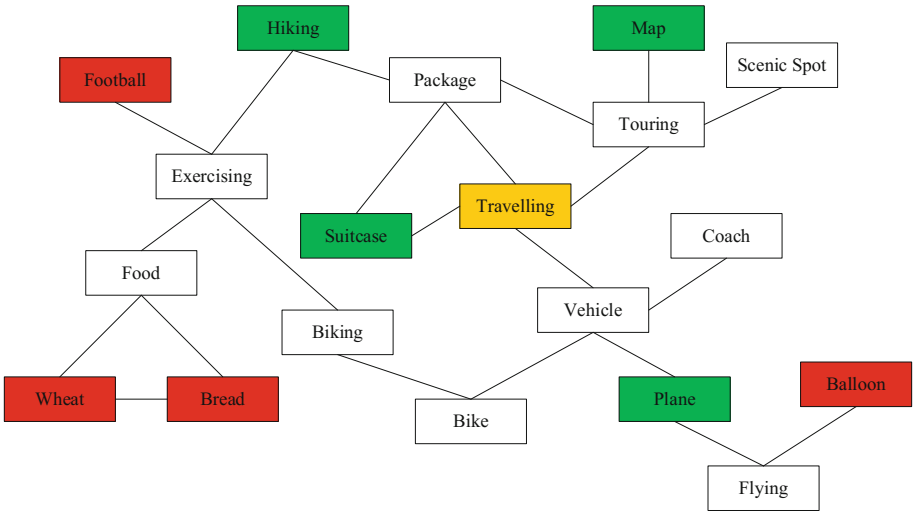

Fig. 5. An example of subgraph of the semantic relevance graph. "Traveling" is the keyword, "hiking", "suitcase", "map", "plane" are the closely-related answers, "football", "wheat", "bread", "balloon" are the interference answers.

If a degree of inter-word correlation is greater than a predefined threshold, the two nodes in the graph are connected with an edge. The weight value on each undirected edge is just the correlation degree of the word pair. The final graph G depicts the correlation of words in the lexicon. Figure 5 shows an example of the subgraph of the semantic relevance graph corresponding to the sample in Fig. 4.

3.4 Choice Generation

Given a graph and a node as the starting point, we first select a neighbor of it at random, and move to this neighbor node as current. Next random move happens subsequently from every current node. The random sequence of nodes selected in this procedure is a random walk in the graph [12]. To generate a CAPTCHA challenge randomly and dynamically, we adopt the random walk model for Relacha.

With a random walk on the connected undirected graph $G = (V, E)$, we define transition probabilities $p_{t+1|t}(j|i)$ from v_i to v_j by normalizing the correlation degree out of node v_i, so $p_{t+1|t}(j|i) = C_{i,j} / \sum_k C_{i,k}$, where v_k ranges over all nodes connected with v_i. The notation $p_{t+1|t}(j|i)$ denotes the transition probability from node v_i at step t to node v_j at time step $t+1$.

Supposing we start at a node v_i; if at the $t-$th step we are at a node v_j with probability $p_{t|0}(j|i)$. Clearly, the sequence of random nodes (v_t: t = 0, 1, 2, …) is a Markov chain. We denote by $M = [m_{i,j}]$, $v_i, v_j \in V$, the matrix of transition probabilities of this Markov chain. So

$$m_{i,j} = \frac{C_{i,j}}{\sum_k C_{i,k}}, \mathrm{v_i}, v_j, v_k \in V, C_{i,k} \in E \tag{3}$$

$$P_{t+1|0} = M^T P_{t|0} \tag{4}$$

Hence

$$P_{t|0} = (M)^{\mathrm{t}} \tag{5}$$

The random walk model takes the semantic relevance graph created in last step as input. The vertices are tags of images, and each edge weight is the degree of correlation between tags. The distance between two vertices is the reciprocal of the edge weight. These random walks do not have restarts (i.e. a teleport probability of returning back to their root) to avoid loop where the content of CAPTCHA answers equals to that of the question in semantics. Because by using image annotation services and libraries, an attack system is able to identify the content of images and to select those depicting similar objects [7].

We sample uniformly a random vertex as the question keyword of a Relacha task, the vertex is also the root of the random walk. For each Relacha, eight tags are needed including both answer tags and interference tags. We first generate the list of answer tags positively correlated with the root as follows:

- Get the sum of degrees of correlation of those vertices connected to the root vertex v_i based on the diagonal matrix $D = \left[d_{i,i}\right]$, so the sum is $d_{i,i} = \sum_k C_{i,k}$.
- Conduct several times of α-steps walk (walking α steps from the starting vertex on the graph, such as 2-steps walk) and record the vertices visited. At the same time, sum the $d_{i,i}$ of each recorded vertex and the sum is denoted as S_i. Stop randomly walking when the following inequality is satisfied.

$$S_i \geq D_{ii}/2 \tag{6}$$

- Count the number of the generated answer tags as β.

Then we generate the list of confusing tags negatively or little correlated with the root as follows:

- Calculate the number of interference tags as γ, C denotes the number of the choices of a Relacha test.

$$\gamma = C - \beta \tag{7}$$

- Carry out γ times random walk and each walk contains at least 2 steps. For each walk, sum the distance of each step as total distance L_i. Stop the walking when L_i is greater than a *min_distance* threshold and record the end vertex.
- Repeat γ times such random walk and we get a list of interference tags.

Then we get the images corresponding to the list of tags from the Internet and present these images to the end-users in web to construct a Relacha challenge.

3.5 Correctness Evaluation

We use a result scoring mechanism to evaluate whether a user has solved the Relacha or not. We first calculate the maximal sum of the degrees of correlation of the images as an optimal value *ov*

$$ov = \max_i \sum\nolimits_i degree_i, i \in \{answers\} \tag{8}$$

Then we define a threshold based on the optimal value as passing mark *pm*

$$pm = conf \times ov \tag{9}$$

where *conf* is a constant value in $(0, 1)$ of desired confidence.

Then we calculate the sum of the degrees of correlation of the images selected by users, and the answers are given weights from high to low according to the selection sequence. Supposing a user has submitted n answers. We get user's score *us* as

$$us = \sum\nolimits_j \left(\mu - (\alpha \div n) \times l_j\right) \times degree_j \tag{10}$$

where $j \in \{user\_answers\}, l_j \in \{1, 2, \cdots n\}$, α and μ are parameters set based on an actual lexicon.

At last, the user score is benchmarked against the passing mark. If *us* is beyond the passing mark, then this user passes the test, otherwise it's a failure. The scoring mechanism uses a threshold of optimal value to tolerate minor mistakes of users and increase the diversity of answers in Relacha tasks by avoiding standard answers.

4 Experiments and Analysis

4.1 Experiments

First we have created a lexicon containing about 200 words from top trending searches within the past five months. We construct the semantic relevance graph of these words. Then we built a website which would present users with the Relacha task. The participants of experiment needed to select the images that they thought correlated to the question.

We use a metric to measure CAPTCHA efficacy with respect to the number of rounds [13]. We define the number of rounds that compose a single CAPTCHA, and the minimum (threshold) number of rounds that a human user must pass to the CAPTCHA. We consider several factors in choosing optimal values for the number of rounds. First, human subjects have limits of how many rounds they are willing to tolerate. A human subject may find 5 rounds acceptable but is unlikely to agree to 500 rounds or more. Second, computers have a speed advantage over humans. A computer can guess more quickly than a human can take a test. Below, we assume that within the time it takes for a human to complete one round, a computer program is capable of completing n rounds.

The CAPTCHA efficacy metric is the probability that in the time it takes a human to take a CAPTCHA, the human will pass and a computer will not. Let p be the probability that a human user will pass a round, q be the probability that a computer will pass a round, n be the number of times faster a computer than a human, m be the number of rounds, and k be the threshold number of rounds. Then the efficacy metric EM is

$$EM = \sum_{i=k}^{m} \binom{m}{i} p^i (1-p)^{m-i} \times \left[1 - \sum_{i=k}^{m} \binom{m}{i} q^i (1-q)^{m-i}\right]^n \tag{11}$$

We conducted 100 rounds tests by 20 volunteer participants from our university using PCs, or smart phones, with average age of 22. Another 1,500 rounds testing were conducted by a robot program selecting answers randomly. During the experiment, we set the pass threshold *conf* in (9) as 75% of the optimal value due to our previous work.

We found the optimal m and k for the experimentally determined values of p. By the second testing, we considered $q = 0.03$. We let $n = 100$ and searched exhaustively over of m and k until $EM \geq 95\%$ and m was minimized. We set the number of steps denoted as α of the random walk in right tags generation as 1, 2 and 3 individually and compare the experiment results in the above three settings.

Figure 6 shows the relation the percent of 100 rounds pass rate by users. The abscissa corresponds to the percent of users, and the ordinate corresponds to the percent of rounds they passed for each type of test. For example, 90% of the users passed 86% of the 2-step walk rounds. There is an obvious decrease of human pass rate when α value increases, because higher α values lead to weaker correlation between the questions and answers, and the questions become more difficult for users to solve.

A robot program was written to attack the Relacha that could select the images presented in web pages randomly. And the program was set to select 1, 3, 5, 8 options once individually. Figure 7 shows the attack pass rate of a robot program in 1-step walk rounds. Figure 8 shows the attack pass rate of a robot program in 2-step walk rounds. Figure 9 shows the attack pass rate of a robot program in 3-step walk rounds. We observe that the pass rate of the robot program with 1,500 tries is below 3.5%, a lot lower compared to humans. When the bot program selects 1 or 8 options once, the attack pass rate is very close to 0. When the bot program selects 3 options once, the attack pass rate is relatively higher.

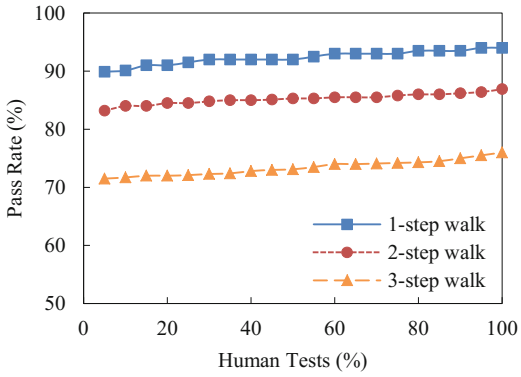

Fig. 6. Round pass rate.

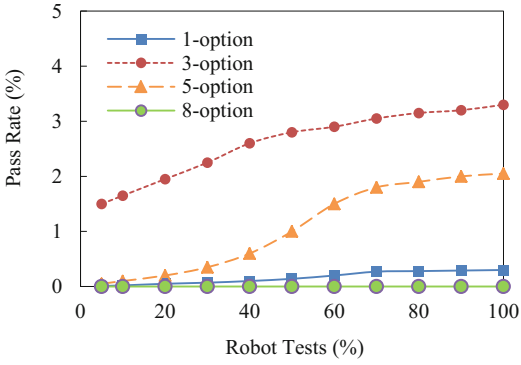

Fig. 7. Attack on 1-step walk.

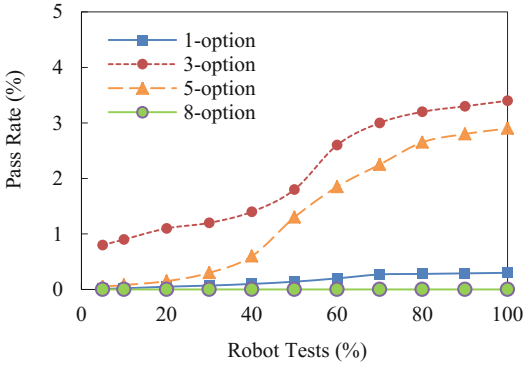

Fig. 8. Attack on 2-step walk.

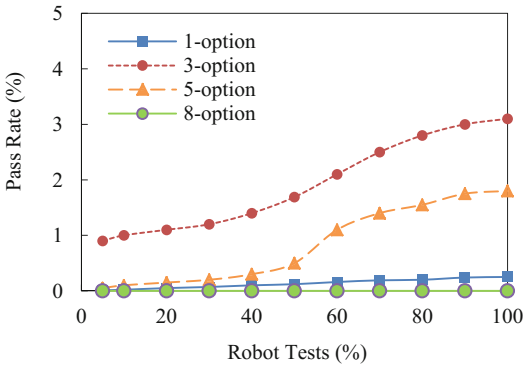

Fig. 9. Attack on 3-step walk.

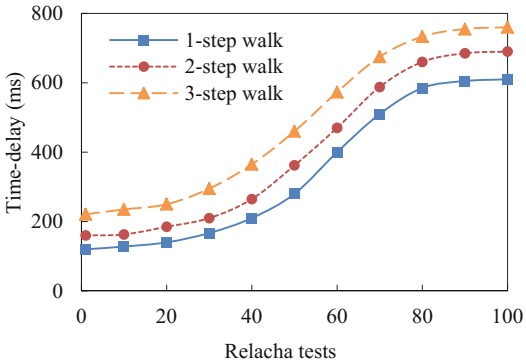

Fig. 10. Time-delays in Relacha.

Figure 10 shows the time-delays in Relacha, consisting of the execution time of Relacha generation and the load time of images. We can see that the maximum delay to present a Relacha challenge is about 760 ms and the minimum delay is about 150 ms. It is obvious that 3-step walk takes the most time to generate a Relacha test and 1-step walk takes the least time. Generally, the time-delays of Relacha stay within users' tolerance.

Table 2 gives the percentile of 100 rounds passed by each user for testing rounds. The upper and lower bounds denote the 95% confidence limits. It also shows the optimal values of m and k and the average time users spent to pass the test. To compare, we also listed the experiment results of reCAPTCHA in [10], including the pass rate of first time and the average time users took to complete a reCAPTCHA challenge. Considering both pass rate and average time, Relacha is user-friendly and easy-using.

Table 2. Experiment results

Test type	Percentile	Optimal m	Optimal k	Authentication time (s)
1-step walk	92.3%	4	2	4.6
2-step walk	85.2%	8	5	5.9
3-step walk	73.4%	10	7	6.1
reCAPTCHA	69%	–	–	17.5 [10]

In general, our experiments show a high performance and satisfaction from human users, and constant failure from conventional machine algorithms.

4.2 Discussion

As the results of experiments shown, Relacha can resist against a robot program that select the options randomly. In addition, it is promising to protect websites from machine learning-based attacks. Relacha strengthens CAPTCHA reliability against machine programs in two folds. First it avoids the dependency of measuring exact-match between CAPTCHA challenge and answers, which is vulnerable either in text or images under today's natural language processing and image pattern recognition technical level. Second, it relieves from the necessity of maintaining a local constant database for text and image material by retrieving contents and measuring their correlations dynamically from public search engine.

5 Conclusion

We propose a new CAPTCHA design called Relacha to distinguish humans from bots by an image correlation test, one that is promising to improve computer and information security. The image tags library used in Relacha system is made up of hot words on Internet which also makes the CAPTCHA challenge more attractive and

meaningful. A scoring mechanism is used for answer evaluation to tolerant small mistakes from different cultural background and knowledge levels of users.

Relacha uses a dynamic lexicon database based on online search engine content. Since public search engines such as Google update their indexes of images frequently [13], attackers as bot programs are unlikely to be able to exhaust all the CAPTCHA images and words. Human users with their rapid and reliable image correlation recognition and comparison (arise from years of experience with the cultural and information environment) can solve Relacha instantly. Relacha takes account of the sequence of clicks on images in correctness evaluation. Because human users tend to click the image they think most related to the question first, while bots start with the images in front.

Relacha was evaluated only against a robot that select the images presented in web pages randomly, namely the weakest possible adversary. As future work, we plan to see how Relacha behaves against more performing adversaries.

Acknowledgments. This material is based upon work supported by the China NSF grant No. 61472189, the CERNET Innovation Project under contract No. NGII20160601, the State Key Laboratory of Air Traffic Management System and Technology No. SKLATM201703, and the Innovation Projects of Beijing Engineering Research Center of Next Generation Internet and Applications. Opinions and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

1. Yan, J., Ahmad, A.S.E.: Captcha robustness: a security engineering perspective. Computer **44**(2), 54–60 (2011)
2. Bursztein, E., Martin, M., Mitchell, J.: Text-based CAPTCHA strengths and weaknesses. In: ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, pp. 125–138. DBLP, October 2011
3. Smeulders, A.W.M., et al.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12), 1349–1380 (2000)
4. He, K., et al.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, pp. 1026–1034 (2015)
5. Datta, R., Li, J., Wang, J.Z.: Exploiting the human–machine gap in image recognition for designing CAPTCHAs. IEEE Trans. Inf. Forensics Secur. **4**(3), 504–518 (2009)
6. Goswami, G., et al.: FaceDCAPTCHA: face detection based color image CAPTCHA. Future Gener. Comput. Syst. **31**(1), 59–68 (2014)
7. Sivakorn, S., Polakis, I., Keromytis, A.D.: I am robot: (deep) learning to break semantic image CAPTCHAs. In: IEEE European Symposium on Security and Privacy, pp. 388–403. IEEE (2016)
8. Zhu, B.B., Yan, J., Bao, G., Yang, M., Xu, N.: Captcha as graphical passwords—a new security primitive based on hard AI problems. IEEE Trans. Inf. Forensics Secur. **9**(6), 891–904 (2014)
9. Catuogno, L., Galdi, C.: On user authentication by means of video events recognition. J. Ambient Intell. Humaniz. Comput. **5**(6), 909–918 (2014)
10. Yang, T.I., Koong, C.S., Tseng, C.C.: Game-based image semantic CAPTCHA on handset devices. Multimedia Tools Appl. **74**(14), 1–16 (2013)

11. Lopezpaz, D., Hennig, P., Schölkopf, B.: The randomized dependence coefficient. In: Advances in Neural Information Processing Systems, pp. 1–9 (2013)
12. Fouss, F., Pirotte, A., Renders, J.M., Saerens, M.: Random walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng. **19**(3), 355–369 (2007)
13. Chew, M., Tygar, J.D.: Image recognition CAPTCHAs. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 268–279. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30144-8_23