CAPTCHA Suitable for Smartphones

Yusuke Tsuruta, Mayumi Takaya, Akihiro Yamamura

CAPTCHA Suitable for Smartphones

Yusuke Tsuruta, Mayumi Takaya and Akihiro Yamamura

Akita University, Department of Computer Science and Engineering,
1-1, Tegata Gakuen-machi, Akita, 010-8502 JAPAN
{tsuruta2013}@gmail.com
{msato,yamamura}@ie.akita-u.ac.jp

Abstract. We propose a CAPTCHA that tells data made by a computer program from one-stroke sketch data given by a human being using embodied knowledge. Utilizing touchscreens of smartphones, we realize this approach and resolve a conceivable inconvenience caused by the existing CAPTCHAs when using smartphones due to the limited display size of smartphones. We implement the proposed technique and analyze its validity, usefulness and security.

Keywords: CAPTCHA, Smartphones, Touchscreens, Embodied knowledge

1 Introduction

A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is one of the reverse Turing tests ([7]) that distinguish an access from a computer program such as a crawler from an access from human beings using the difference between humans' shape recognition ability and the machine recognitions ([2, 3]). The computer program might access to network services to acquire a large amount of accounts of the web mail service aiming at an illegal network utilization such as sending spam mails or carrying out APT attacks. A CAPTCHA can be applied to a web security technique preventing such illegal accesses to network service. When facing the existing CAPTCHA, a human recognizes a word, maybe a nonsense sequence of characters, in the image on the display and is required to respond by typing the word through the keyboard (Fig. 1). The character image is distorted in some fashion, and computer programs cannot recognize easily the characters. For example, an OCR program cannot recognize the character, whereas human beings can do it without difficulty. CAPTCHAs have been analyzed and several methods based on different principles have been proposed ([4]).

Smartphones play an important role in the information-communication society nowadays, and the development of cloud computing promotes the spread of smartphones and has influenced the use of the Internet. Actually, the accesses from smartphones to internet services rapidly are increasing, and actually many web sites have begun to correspond to smartphone users. There are many differences between smartphones and the past computer models from the point of

view of human-computer interface. For smartphones, data is inputted through the touchscreen by hands, and the display of a smartphone is comparatively small. A CAPTCHA login is requested when a user is accessing to internet services through a smartphone exactly same as it is requested to accesses from the desktop PCs. When we use smartphones and face a CAPTCHA, both the challenge image and the virtual keyboard are displayed, and we have to type in the word displayed in the image. However, the virtual keyboard occupies almost half of the display and so the CAPTCHA image must be small. To solve this problem, a new image based CAPTCHA is proposed in [5]. In this paper, we propose yet another CAPTCHA suitable for smartphones using embodied knowledge of human beings. Our approach is different form [5].

Embodied knowledge is the control of the muscle acquired by practice and the motor learning of the brain such as the skill remembered in childhood. Realizing embodied knowledge by a computer is one of the challenging problems in artificial intelligence research. The proposed technique is to decide whether or not the response to the challenge is created by human beings or computer programs by checking the existence of embodied knowledge. One-stroke sketch is taken up as an ingredient in the embodied knowledge of our proposal. Human-computer interaction through a touchscreen that is one of the features of smartphones is suitable for faithfully acquiring one-stroke sketch data. One-stroke sketch input data with humans' finger is characterized as a continuous locus resulted by the human hand's physicality and realized as a series of coordinates on the display. The entire character image is not drawn at the same time but it is drawn continuously on the curve along the shape of the character little by little following the tracks of the tip of a finger according to the operation of the arm and the hand.

It is also necessary to give data the continuous order at the pixel level so that the computer program may compose legitimate input data. Therefore, the proposing CAPTCHA is based on not only hardness of image recognition but also the embodied knowledge which is an important theme in artificial intelligence.

Fig. 1. A typical CAPTCHA

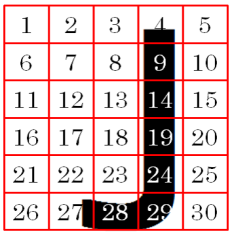

Fig. 2. Example("J")

2 The Proposed Technique

2.1 Basic Idea

As a standard authentication protocol, a CAPTCHA server sends a challenge and the user has to respond to it in a correct way. In the case of the proposed CAPTCHA, the server sends a challenge image that includes a character (or a symbol), and then the user is requested to trace the character by a finger tip and sends back the data representing a one-stroke sketch as a response to the challenge. The smartphone interprets the date inputted as an ordered series of coordinates in which the order is given as the time series. The server receives the ordered series of coordinates and check whether or not it is acceptable as a data obtained by a human using implicitly embodied knowledge. If the data received is determined as an output of a computer program then the access is rejected.

The touchscreen is delimited and the grid is composed. In tracing the character included in the image on the display by the finger, coordinates of the points in the display touched by the finger are acquired by the drug operation of the touchscreen. The server determines whether or not the series of coordinates is acceptable by checking the locus is correct and the data input is continuous in addition to the correctness of the the starting point and the terminal point.

For instance, suppose the image of the character "J" is displayed as a challenge in Fig. 2. The correct response is obtained by dragging the finger along the shape of "J" on a touchscreen of a smartphone. Coordinates of the input data, that is, the series of coordinates of the locus are checked if the first coordinate is included in the small area 4, and if the following coordinates are in the small area 9 and so on. If the series of ordered coordinates is nearly in the order of the small areas 4, 9, 14, 19, 24, 29, 28, and 27, it is accepted (Fig. 2). If the coordinates is in the order of the small areas 2, 8, 14, 19, and 20, then the data is rejected. We shall explain the proposed technique in detail in the expanded version of the paper.

2.2 Security

The main objective of CAPTCHAs is to prevent computer programs from accessing to network services for evil purposes. Therefore, the attacker is a computer program disguising as a human being and trying to obtain a legitimate authority to access. Then the security of a CAPTCHA technology is evaluated by the intractability for computer programs to obtain the access permit ([1, 6]). We analyze conceivable attacks against the proposal CAPTCHA.

Suppose that the image displayed as a challenge is monochrome, and the character is drawn in the white ground in black. In this case, the coordinate data of the area where the character is drawn can be acquired accurately by examining RGB of the challenge image. Then a computer program should enumerate a series of coordinates at which black RGB is appointed and give order to these coordinates according to the correct writing, that is, following one-stroke sketch. For example, if a human write "J" then the input data trace the small areas 4,

9, 14, 19, 24, 29, 28, and 27 like in Fig. 2. The number of the coordinates should be nearly same as the standard input by human beings to disguise. In general, a computer program has no information on one-stroke sketch, which is considered as an embodied knowledge of human beings. Each human being has learned such an embodied knowledge from their childhood. A computer program should pick one position from the area of a coordinates with black RGB as the starting point and also as the terminal and then compose a series of coordinates with black RGB that connects the starting and terminal points in a correct order. It is impossible to execute this task if there is no information on the stroke order. If many responses are permitted to the same challenge image, the brute force attack becomes possible in principle. However, the brute force attack can be avoided by permitting only one response to each challenge. Moreover, it is realistic to put the limitation on the challenge frequency.

Now assume that an attack program has the database concerning characters and the correct order of writing. If a series of coordinates can be correctly obtained, then information on the correct order of writing might be able to be obtained from the database. We note that it costs a lot to make such a database for attack against the CAPTCHA and so this already has some deterrent effect. In addition, the challenge is not necessarily based on a character or a symbol. An arbitrary curve can be used for a challenge instead of a character and then making the database is impossible in principle. We shall discuss this issues in the future work. We may execute transformations on the shapes and colors of the character to perplex computer programs. If the transformation processing is a continuous transformation, this occurs no trouble for human beings and so such a transformation is allowed. It seems difficult for computer programs to respond correctly (Fig. 5). If the challenge is a (not necessarily monochrome) color image, the attacker's program has to carry out an edge detection and specify the character. Using the existing CAPTCHA techniques such as adding the distortion to the character, the attacker's program has the difficulty to detect the character. Moreover, not only adding the distortion transformation but also camouflaging the background with the dazzle paint makes the attacker's program hard to detect the character. Therefore, the security of the proposed CAPTCHA is at least the existing CAPTCHAs because their techniques can be employed to our CAPTCHA as well. In addition, the method requiring the user to input more than one stroke traces is effective to improve the security level. The security level can be adjusted according to the system requirement. To understand the security of the proposed CAPTCHA well, we should examine human embodied knowledge from the standpoint of the cognitive psychology.

Fig. 3. Separator Image

Fig. 4. Deformed Image

Fig. 5. Distorted Image

2.3 Comparison with the Existing CAPTCHA

We discuss the usefulness of the proposed CAPTCHA comparing with the existing techniques provided that a user is accessing using a smartphone. Note that the screen size of smartphones is about 3.5-5 inch. When using smartphones, both a CAPTCHA image and a virtual keyboard are displayed (Fig. 6) and the size of the CAPTCH image is almost half of the screen. It is very inconvenient for most of the users to respond to a CAPTCHA challenge due to this limited size image and the virtual keyboard. One has to type more than once to input a character when using a virtual keyboard. For example, when typing "c" in the lowercase letter, one has to press the button for "c" three times (Fig. 7). When typing "C' ' in the uppercase letter, one needs more operations to change the "lowercase mode" to the "uppercase mode". Therefore, the total number of operations becomes enormous if the words are arbitrarily generated using lowercase letters, uppercase letters and figures. Moreover, a wrong character may be inputted by an unintentional typing mistake. For this reason, some existing CAPTCHA use only figures $(0, 1, 2, \ldots, 9)$ without using alphabets to improve user's convenience. Note that if uppercase and lowercase letters are allowed in addition to the figures, $62(= 10 + 52)$ characters can be used. This results in the deterioration of the security; if the challenge is a word consisting of n letters, there are only 10^n cases compared with 62^n cases.

When using the proposed CAPTCHA, the entire display is used for showing the challenge image, and the input is comparatively easy (Fig. 8); no additional operations such as changing modes are required.

Fig. 6. Existing(num)

Fig. 7. Existing(char)

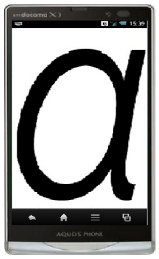

Fig. 8. Proposed Method

3 Line Trace Attack

It may seem possible to use the line trace program, which is often used in a robot, to trace the black coordinate area of the challenge image for attacking the CAPTCHA. For this attack, the line trace program has to trace on the black area from the starting point of the character to the terminal point in order to compose the response data. The attack using a line trace program seems the most plausible

attack as of now. It is necessary for a line trace program to find the starting point to begin the tracing, however, it seems intractable to find the starting point because the line trace program checks the local area and determines the next action and the starting point is usually given as the input to the program by a human being. A human being looks at the image, comprehends the character and finds the starting point using the embodied knowledge. On the other hand, choosing a starting point is intractable for a line trace program. If a human takes part in the attack, the attacker consists of not only a program but a human, and so this approach is excluded as an attack against the proposed CAPTCHA. For an objective of a CAPTCHA is to prevent programs from accessing without human beings assistance. Even if the starting point is obtained in some ways without human assistance, our approach allows challenges such as separated images (Fig. 3) or deformed images (Fig. 4) to perplex the line trace program, which give no trouble to human beings as we see in the subsequent section. Therefore, an attack using line trace programs seems intractable.

3.1 Experiment of Attack Using Line Trace Program

In the following experiments of attacking against the proposed CAPTCHA, a line trace program tries to make an acceptable response to the challenge images (Fig. 3, 5, 9, 10, 11, 12, 13, 14). Each experiment is executed provided the starting point is given to the program beforehand by a human. We use a simulation line trace program [8] in this experiment.

Fig. 9. Test Image 1

Fig. 10. Test Image 2

Fig. 11. Test Image 3

Fig. 12. Test Image 4

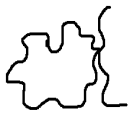

Fig. 13. Test Image 5

Fig. 14. Test Image 6

The line trace program succeeded in making an acceptable response only to the challenge image in the image 4 (Fig. 12), and it failed against the other images (see Table 1). By these experiments, we conclude that countermeasures leading

the line trace program to a dead end or putting the pause in the character shape are considerably effective whereas these do not cause any troubles to human beings. The line trace program also fails to trace when the angle formed in the character shape is too big. As we have already mentioned that the line trace program is given the starting point as an input by human beings. However the actual attack must be carried out without human beings' assistance. Therefore, a simple attack using a line trace program does not seem a serious threat against the proposed CAPTCHA. We shall report the detail of the experiments and discuss more about the results in the expanded version of this paper.

Table 1. Correspondence Table

	Image1	Image2	Image3	Image4
Trace Success	-	-	-	√
Trace Failure	√	√	√	-
	Image5	Image6	Fig. 3	Fig. 5
Trace Success	-	-	-	-
Trace Failure	√	√	√	√

4 Validity of the Proposed CAPTCHA

We examine the validity of the proposed CAPTCHA by experiments; 22 subjects (humans) are asked to respond to several challenge images that represent the symbol "α".

4.1 Experiments

We use a handheld computer (Android 3.1 and processor NVIDIA Tegra 2 mobile processor) equipped with 9.4 type WXGA liquid crystal with the internal organs display touchscreen of the ITO Grid method mirror electrostatic capacity method as the user machine. The platform of the server is constructed on Windows 7 Professional 64bit, 2048MB memory, and Intel Core i3, and the authentication program is written by using c/c++ compiler MinGW. The size of the challenge images is 1200×700 pixel. The response is accepted if the locus is passing in a correct order.

The purposes of each experiment are summarized as follows (see Table 2). In the experiment 1 and 2, the small zone is set 35×35 pixels and 70×70 pixels, respectively, and we investigate the differences between these two cases. In the experiment 3, the small zone is set 35×35 pixels and we specify the entry speed and the input position and investigate the difference between these cases. In the experiment 4, we investigate the effect caused by the change of characters. In the experiment 5, we investigate the case that the response is accepted only when all set coordinates are passed. In the experiment 6, we investigate the tolerance for human non-intentional errors.

experiment 1 The instruction "Please trace on the character shape by one stroke" is displayed and the challenge image Fig.9 is displayed. The small zone on the grid is 70×70 pixel(Fig.15).

experiment 2 The instruction "Please trace on the character shape by one stroke" is displayed and the challenge image Fig.9 is displayed. The small zone on the grid is 35×35 pixel (Fig.16).

experiment 3 The instruction "Please trace on the character shape by one stroke within 5 seconds" is displayed and the challenge image Fig.9 is displayed. The small zone on the grid is 35×35 pixel (Fig.16).

experiment 4 The instruction "Please trace on the character shape by one stroke within 5 seconds" is displayed and the challenge image Fig.10 is displayed. The small zone on the grid is 35×35 pixel (Fig.17).

experiment 5 The instruction "Please trace on the character shape by one stroke within 5 seconds" is displayed and the challenge image Fig.10 is displayed. However, the response is accepted only when every small zone from 1 to 40 is passed in order. The small zone on the grid is 35×35 pixel (Fig.17).

experiment 6 The instruction "Please perform the input which is not related to the displayed character" is displayed and the challenge image Fig.9 is displayed. The small zone on the grid is 35×35 pixel (Fig.16).

Table 2. Correspondence Table

	Im1	Im2	Im3	Im4	Im5	Im6
Character α	√	√	√	-	-	√
Character β	-	-	-	√	√	-
35×35 pixel	-	√	√	√	√	√
70×70 pixel	√	-	-	-	-	-
Specified time (about 5 seconds)	-	-	√	√	√	-

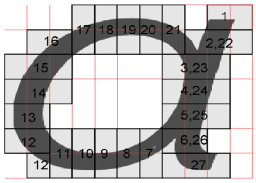

Fig. 15. exp 1

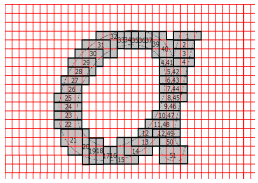

Fig. 16. exp 2-3

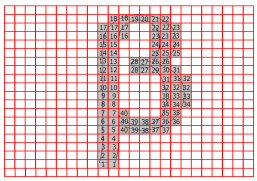

Fig. 17. exp 4-5

4.2 Result of Experiments

The results of the experiments in Section 4.1 are summarized in Table 3. In the experiment 1, 2, and 3, one subject is rejected because the responding data is

in the order corresponding to the alphabet "a". Recall that the image indicates the symbol "α". When one writes "α", the order is different from the alphabet "a" although the shape is similar. The difference of the handwritten input of "α" and "a" is due to the culture and a social background in which the subject has grown up, and this is considered an embodied knowledge.

By the results of experiments 1 and 2, we can conclude that if the zone is bigger, then higher acceptance rate is achieved, on the other hand, if the zone is smaller, then acceptance rate decreased. By the results of experiments 2 and 4, we can conclude that the shape of the character does not affect the acceptance rate and the acceptance rate is stable for any (simple) character. We are convinced that other characters which are written as one-stroke sketch other than "α" can be used in the proposed CAPTCHA as well. By the result of experiment 3, we can conclude that if we allow users to write slowly then the acceptance rate will increase but the transmission data gets larger, which is not desired for network congestions. By the results of experiments 4 and 5, we can conclude that it is necessary to permit width of the order of the inputted coordinates to some degree, that is, we must be tolerant to small errors data, possibly caused by an unintentional errors. More experiments and detailed analysis will be reported in the expanded version of the paper.

Table 3. Test Result

	Test1	Test2	Test3	Test4	Test5	Test6
Number of Subjects (People)	22	22	22	22	22	22
Acceptance Number (Time)	21	20	21	22	9	0
Acceptance Rate (%)	95.5	90.9	95.5	100	40.9	0

5 Future Work and Summary

We shall discuss several issues on the proposed CAPTCHA for future research. The response data to a challenge image of the proposed CAPTCHA consists of a series of coordinates. One coordinate consists of a pointer ID, x coordinate, and y coordinate and each data is 9 bytes, where, pointer ID indicates a human action on the touchscreen. The number of input data comprises about 150-200 coordinates in our experiment using the symbol "α". One coordinate is inputted per 0.01-0.02 seconds. Therefore, 1.35-1.8 kilobytes transmission is required for each response for the proposed CAPTCHA. The response data for the existing CAPTCHA is 6-10 characters, and the transmission data is several bytes. Thus, the transmission data is bigger than the existing CAPTCHA. The proposed CAPTCHA is required more computation to check whether or not a response can be accepted than the existing CAPTCHA. We will study how to reduce amount of transmission data and server's information processing.

In our experiment we made the challenge images and the authentication programs by hand. Automatic generation of the challenge image is necessary when we use it in a real system. Because one-stroke sketch is an embodied knowledge, it is important to devise a method to put embodied knowledge in challenge images and to apply continuous transformations to a character in order not to change the writing order. It should be note that there is no difference in the programs on Android OS but the adjustment of coordinates for platform smartphones is necessary. We will discuss these issues in the extended version of the paper.

In this paper, we propose a new CAPTCHA technique utilizing touchscreens to solve an inconvenience caused by the existing CAPTCHAs when using smartphones. We implement the proposed technique and carry out experiments to examine the usefulness and compare with the existing techniques. Using a touchscreen, one-stroke sketch is captured and represented as ordered series of coordinates. One-stroke sketch can be considered as one of embodied knowledges of human beings and so computer programs have difficulty to understand one-stroke sketch. Our technique is based on embodied knowledge of human beings and so computer programs cannot respond correctly to a challenge image. It is necessary to study more one stroke sketch as an embodied knowledge of human beings and validity and security of the proposed technique in the context of artificial intelligence and cognitive science.

References

1. Ahn, von, L., Blum, M., Hopper, N., Langford, J.: CAPTCHA: Using Hard AI Problems for Security. In: Biham, E. (ed.) Eurocrypt 2003. LNCS, vol. 2656, pp. 294-311. Springer, Heidelberg (2003)
2. Ahn, von, L., Blum, M., Langford, J.: Telling Humans and Computers Apart Automatically. Communications of the ACM, 47, pp. 56-60. (2004)
3. Ahn, von, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science, 321, pp. 1465-1468. (2008)
4. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In: ACM Conference on Computer and Communications Security, pp. 366-374. (2007)
5. Gossweiler, R., Kamvar, M., Baluja, S.: What's up CAPTCHA?: a CAPTCHA Based on Image Orientation. In: International Conference on World Wide Web, pp. 841-850. (2009)
6. Mori, G., Malik, J.: Recognizing Objects in Adversarial Clutter : Breaking a Visual CATCHA. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 134-141. (2003)
7. Turing, A.M.: Computing Machinery and Intelligence. Mind, vol. 59, 236, pp. 433-460. (1950)
8. `http://www.yamamoto-works.jp`