In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, April 2010, pp.430-437.

CAPTCHA Using Strangeness in Machine Translation

Takumi Yamamoto[1, 3] J. D. Tygar [2] Masakatsu Nishigaki[1, 4]
[1]Graduate School of Science and Technology Shizuoka University, 3-5-1 Johoku, Naka, Hamamatsu, Japan
[2] Computer Science Division, University of California, Berkeley, UCB-SIMS 102 South Hall #4600
Berkeley, CA 94720-4600 USA
[3]Research Fellow of the Japan Society for the Promotion of Science (DC)
[4]Japan Science Technology and Agency， CREST
E-mail: f5745037@ipc.shizuoka.ac.jp, tygar@cs.berkeley.edu, nisigaki@inf.shizuoka.ac.jp

Abstract— **CAPTCHA is a technique that is used to prevent automatic programs from being able to acquire free e-mail or online service accounts. However, as many researchers have already reported, conventional CAPTCHA could be overcome by state-of-the-art malware since the capabilities of computers are approaching those of humans. Therefore, CAPTCHA should be based on even more advanced human-cognitive-processing abilities. We propose using the human ability of recognizing "strangeness" to achieve a new CAPTCHA. This paper focuses on strangeness in machine-translated sentences as an example, and proposes CAPTCHA using Strangeness in Sentences (SS-CAPTCHA), which detects malware by checking if users can distinguish natural sentences created by humans from machine-translated sentences. We discuss possible threats to SS-CAPTCHA and countermeasures against these threats. We also carried out basic experiments to confirm its usability by human users.**

CAPTCHA, advanced human cognitive processing abilities, strangeness, machine translated sentences, SS-CAPTCHA (key words)

I. INTRODUCTION

With the expansion of Web services, denial of service (DoS) attacks by malicious automated programs (e.g., bots) are becoming a serious problem as masses of Web service accounts are being illicitly obtained, bulk spam e-mails are being sent, and mass spam blogs (splogs) are being created. Thus, the Turing test is becoming a necessary technique to discriminate humans from malicious automated programs and the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [1] developed by Carnegie Mellon University (CMU) has been widely used. The simplest CAPTCHA presents distorted or noise added text (Fig. 1) to users who visit various Web sites and want to use its services. We refer to this simple CAPTCHA as text recognition based-CAPTCHA. If they can read the given text, they are certified as human. If they cannot read the text, they are certified to be malicious automated programs (bots).

However, many researchers have recently pointed out security problems with conventional text recognition based-CAPTCHA [2]. Malicious automated programs that install a sophisticated Optical Character Reader (OCR) have been spreading and these have cracked conventional text recognition based-CAPTCHA [3,4].

Figure 1. Example CAPTCHA used for Google Accounts

It has become difficult for automated programs to pass tests (read texts) by increasing distortion or noise. However, it has also become difficult for humans to read texts. We therefore need to adopt even more advanced human cognitive processing abilities to enhance CAPTCHA to overcome this problem.

Image recognition-based CAPTCHA such as Asirra [7] is known as one of the effective solutions to enhancing CAPTCHA, because image recognition is much harder problem for machine than character recognition [1,5-7]. Labeled images are used in image recognition-based CAPTCHA to confirm that a user can recognize the meaning of the images. Several photos of animals are presented to a user in Asirra. The user is then asked to select a specific animal in a test. For example, suppose that the user is asked to select a "cat" and he/she can select all photos labeled as cat in the test, he/she is certified to be human. If not, he/she is certified to be an automated program.

However, a technique that has effectively been used to breach image recognition-based CAPTCHA has been reported and shocked many researchers [8]. Advancements made to cracking capabilities (CAPTCHA cracking algorithms and CPU processing speeds) will never end. No matter how advanced malicious automated programs are, a CAPTCHA that will not pass automated programs is required. Hence, we have to find another more advanced human cognitive processing ability to tackle this challenge.

This paper focuses on the human aptitude for recognizing "strangeness" as one of our more advanced human capabilities. When we encounter a situation that is a departure from our knowledge or common sense, we feel as though there is something wrong or that does not sit well with us. We define these feelings as "strangeness". If we gain more common sense or have more experience, our abilities to recognize strangeness become more sophisticated and we will be able to notice more subtle differences. Hence, recognizing strangeness is expected to be one of the most advanced mechanisms for recognition by humans, and we know it is quite difficult for automated programs (computers) to recognize strangeness the way humans do.

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.55

Therefore, we propose using the human ability of recognizing "strangeness". We have focused on strangeness in machine-translated sentences as an example in this paper. Thus, this paper proposes a new CAPTCHA that detects an automated program by checking if a user can distinguish natural sentences created by humans from machine-translated sentences. A human can recognize strangeness in machine-translated sentences and easily identify natural sentences while an automated program cannot notice subtle differences between natural sentences and machine-translated sentences. Therefore, the automated program cannot tell natural and machine-translated sentences apart.

In this paper, we refer to the proposed CAPTCHA as CAPTCHA using Strangeness in Sentences (SS-CAPTCHA). The next section introduces related works of image recognition-based CAPTCHA and the concept behind SS-CAPTCHA is described in Section III. We then discuss some possible attacks against SS-CAPTCHA and describe countermeasures against these attacks in Section IV. We carried out a basic experiment to test and confirm the usability of SS-CAPTCHA, which is explained in Section V. Finally, we discuss our future work and conclusions in Section VI.

II. RELATED WORKS

ESP-PIX, so called "naming images CAPTCHA" has been proposed in CMU [1]. In ESP-PIX, a user is presented several distinct images of same subject. Then the user has to correctly type the common term associated with the images ("elephant" in Fig.2).

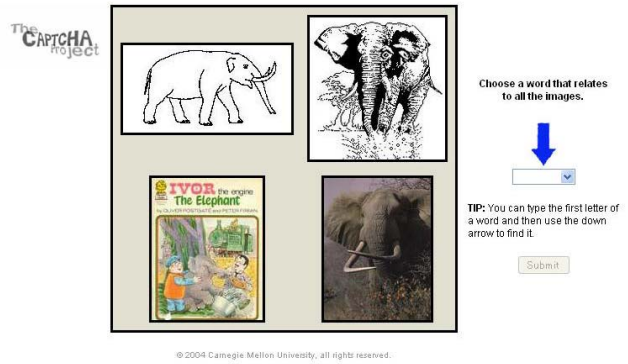

Figure 2. Example authentication window for ESP-PIX [1]

Chew and Tygar proposed two variations on the naming images CAPTCHA [5] (distinguishing images CAPTCHA and anomaly images CAPTCHA). The distinguishing images CAPTCHA presents two sets of images to a user (Fig.3). Each set has three images of the same subject. With equal probability, both sets either have the same subject or not. The user has to answer whether or not the sets have the same subject to pass the test.

Figure 3. Example authentication window for the distinguishing images CAPTCHA [5]

The anomaly images CAPTCHA presents five images of the same subject and one anomalous image to a user (Fig.4). The subject of the anomalous image is different from that of the other five images. The user has to identify the anomalous image to pass the test.

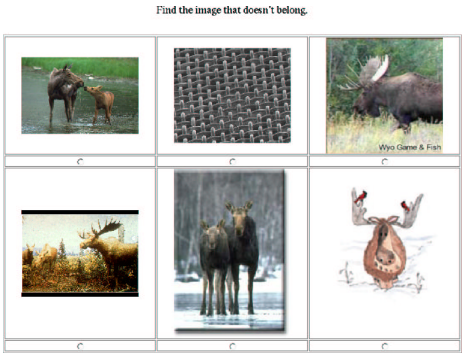

Figure 4. Example authentication window for the anomaly images CAPTCHA [5]

Oli proposed Kitten Auth in which several photos of animals are presented to a user (Fig.5). The user has to select a specific animal among them [6].

Figure 5. Example authentication window for the Kitten Auth [6]

Microsoft's Asirra [7] is similar to Kitten Auth in outline. A user has to select a specific animal among several images

(Fig.6). The notable feature of Asirra is that it works together with the world's largest site for homeless pets. The site provides extremely large number of labeled photos of animals. Thus the image database of Asirra is large enough to prevent attackers from reconstructing the database manually.

Figure 6. Example authentication window for Assira [4]

So far, image recognition-based CAPTCHAs have been considered as one of the best alternatives to text recognition based-CAPTCHAs (Fig. 1). However, a technique that has effectively been used to breach image recognition-based CAPTCHA has been reported and shocked many researchers [8]. Advancements made to cracking capabilities (CAPTCHA cracking algorithms and CPU processing speeds) will never end. No matter how advanced malicious automated programs are, a CAPTCHA that will not pass automated programs is required. Hence, we have to find another more advanced human cognitive processing ability to tackle this challenge.

III. SS-CAPTCHA

A. Concept

Although current machine-translation techniques have progressed a great deal, one of the most difficult problems seems to be, even for a state-of-the-art machine translator, to automatically generate perfectly natural sentences that will not make a human feel as though something is wrong. As most machine-translated sentences sometimes make humans feel as though something is wrong, a sentence translated from a non-mother tongue into a mother-tongue language with a machine translator usually sounds strange and does not pass inspection by a native speaker.

This means an automated program (computer) cannot accurately recognize the meaning of natural sentences. In other words, it is expected to be almost impossible for an automated program to notice subtle differences between natural sentences and machine-translated sentences. Therefore, it cannot tell natural sentences and machine-translated sentences apart. This is because if an automated program could feel as though something were wrong in machine-translated sentences as humans can, a machine translator would be able to generate more natural sentences than is presently possible by self-checking.

Thus, in this paper, we propose a new CAPTCHA called SS-CAPTCHA that detects an automated program by checking if a user can distinguish natural sentences created by humans from machine-translated sentences. There is an overview of SS-CAPTCHA in Fig. 7.

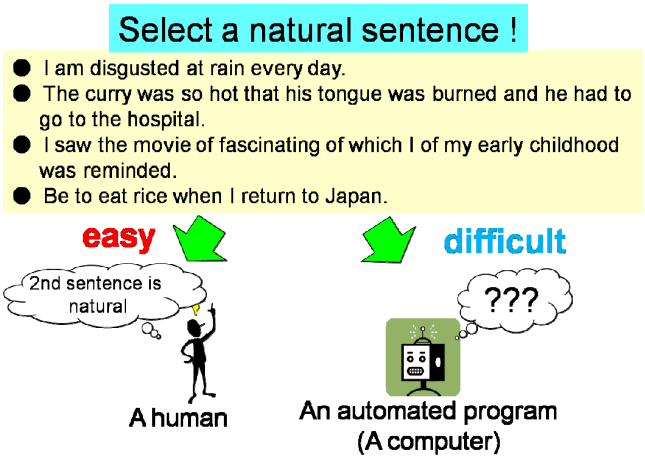

Figure 7. Overview of SS-CAPTCHA

To be more precise, a system simultaneously presents P natural sentences created by humans (NSs) and Q garbage sentences generated from a natural sentence by a machine translator (GSs) to a user. These (P+Q) sentences are placed in random order. The user is then required to select P NSs from (P+Q) sentences. If he/she can correctly select all P NSs, he/she is certified to be human. Otherwise, he/she is certified to be an automated program.

B. Sentence collection

CAPTCHA should have the ability to automatically and infinitely generate challenges. Therefore, SS-CAPTCHA should be able to automatically collect NSs and GSs, which are used for CAPTCHA challenges.

1) GS collection

A garbage sentence (GS) can be generated by translating a natural sentence (NS) created by humans from a non-mother-tongue into a mother-tongue language with a machine-translation program. A GS can also be generated by re-translating machine-translated sentences from a non-mother-tongue into a mother-tongue language after a NS is translated from a mother-tongue into a non-mother-tongue language (e.g., Japanese => English => Japanese).

Several different languages can be combined to generate a GS (e.g., Japanese => German => French => Chinese => English => Japanese). Moreover, a combination with several distinct machine-translation algorithms (programs) can also be used.

2) NS collection

We can see lots of natural sentences created by humans on the Internet. However, we cannot utilize these sentences as NSs for SS-CAPTCHA. If we use such sentences as NSs, NSs can always be searched while GSs cannot always be searched on the Internet. That is, malicious automated programs can easily

recognize that searchable sentences in a CAPTCHA test are likely to be NSs.

Thus, NSs used in SS-CAPTCHA should not appear often on the Internet. Such sentences can be extracted from paper media (e.g., newspapers, magazines, and books), which are periodically updated. Moreover, natural and formal sentences uttered by announcers in news shows are expected to be extracted with voice recognition software in the future. This paper does not explore reliable ways of collecting NSs but leaves this challenge for future work.

3) Sentence validation

A human sometimes creates strange sentences for other people. Likewise, a machine translator sometimes generates natural sentences for humans. That is, NSs created by humans are not always natural for us while GSs generated by a machine translator do not always make humans feel as though something is wrong. In this way, inappropriate NSs or GSs may be occasionally produced. If such NSs and GSs are used in a test of SS-CAPTCHA, a human user will be confused.

Therefore, newly created NSs and GSs are not used as CAPTCHA tests in SS-CAPTCHA immediately after acquisition. Such sentences (newly created NSs and GSs) are used as candidate sentences. We refer to these newly created NSs and GSs as Candidate NSs (CNSs) and Candidate GSs (CGSs). The CNS and the CGS should be checked whether the CNS and CGS are appropriate as an NS and a GS in SS-CAPTCHA. Sentences confirmed to be appropriate NSs or GSs for CAPTCHA tests are referred to as True NSs (TNSs) and True GSs (TGSs).

In SS-CAPTCHA, four types of sentences (TNS, TGS, CNS, and CGS) are displayed in a CAPTCHA test at a time. The TNS and TGS work as the Turing test to tell humans and computers apart, by checking whether users can distinguish a TNS from a TGS. If CNSs and CGSs are selected by a user who has been certified to be human according to the selection of TNS, these sentences are likely to be natural sentences that do not make humans feel as though something is wrong. Then, such CNSs and CGSs can be used as future TNSs. Likewise, CNSs and CGSs not selected by users who have been certified to be humans are likely to be garbage sentences that make them feel as though something is wrong. Then, such CNSs and CGSs can be used as future TGSs. The more human users verify each CNS or CGS, the more appropriate a TNS and a TGS will be as an NS and a GS.

IV. Possible Attacks against SS-CAPTCHA

In this section, we introduce two possible attacks against SS-CAPTCHA. We then discuss the seriousness of these attacks and countermeasures against them.

A. Convergence Analysis Attack

1) Convergence of results of machine translation

Let T be a function that translates an input sentence into another language (e.g., Japanese to English). Here, R is a function that re-translates the translated sentence into its original language (e.g., English to Japanese). Consequently, we have a function, F=R(T(S)), which is a composite function of the form.

Now, let S_1 be a sentence that is expressed in Japanese. Then, we can obtain $S_2=F(S_1)$, which is a machine-translated Japanese sentence with function F and S_2. Likewise, $S_3=F(S_2)$ can be obtained with function F and S_2. We can observe that $F(S_{i+1})$ equals $F(S_i)$ after C repetitions of this process (i=C). C is the number of repetitions that function F is applied until $F(S_{i+1})$ equals $F(S_i)$. Here, C is sentence dependent. We investigate the tendencies of C.

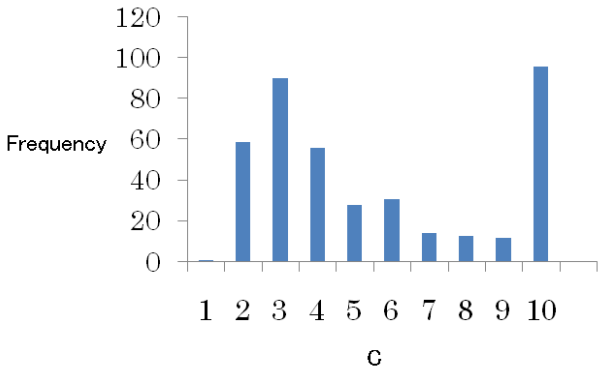

Figure 8. Histgram of C

Fig. 8 is a histogram of C that is obtained by applying an NS collected in advance to function F until $F(S_{i+1})$ equals $F(S_i)$. The S_1 used for this analysis were natural sentences which obtained by extracting sentences of adequate length (11–58 characters) from a Japanese newspaper and a book. Two hundred sentences each were extracted from the newspaper and book. We focused on Japanese as the language for pre-translated sentences and English as that for post-translated sentences and used a free translator, which is available on the Internet [9].

As C increases, the time needed for analysis increases. Therefore, we have ceilings on the number of repetition of translations. If $F(S_{i+1})$ does not equal $F(S_i)$ even when i equals 10, the translations are not repeated and C is set to 10.

According to Fig. 8, we can see that the translations from an NS (natural sentence) often converge (in other words, $F(S_{i+1})$ equals $F(S_i)$) after 2–4 repetitions of translation with function F, with the exception of the maximum limit case (C=10).

Considering that a GS is generated by applying an NS to function F once in SS-CAPTCHA (GS=F(NS)), the number of repetitions needed for converging the translation from a GS is obtained by subtracting one from that from an NS. Thus, we can easily understand that the translations from a GS often converge after 1–3 repetitions of translation with function F. This characteristic allows an automated program (computer) to have a strategy where a sentence that has a relatively smaller C than other sentences is likely to be a GS. We refer to this strategy as a convergence analysis attack and discuss countermeasures against this in the next section.

2) Countermeasures against Convergence Analysis Attack ~Alignment of Cs of NS and GS~

The difference in C (the number of repetitions where function F is applied until $F(S_{i+1})$ equals $F(S_i)$) between NSs and GSs makes a convergence analysis attack feasible. Hence, the GSs displayed in a window should be comparable in C to the NSs in the same window.

To make GSs comparable in C to NSs, the system discussed in this paper selects NSs and GSs in such a way that the mean value and standard deviation of Cs calculated from NSs and GSs displayed in a window become equivalent. By doing this, it becomes almost impossible to exploit the C of sentences in a window to identify GSs.

B. Search-engine-based Attack

1) Search-result differences between NS and GS

We can find numerous natural sentences created by humans on the Internet. On the other hand, in general machine-translated sentences (GSs) rarely get to be on the Internet. Hence, by comparing the Internet-search results, it may be possible to discriminate NSs from GSs in a CAPTCHA test. In this paper, we refer to this strategy as a search-engine-based attack.

As explained in Sec.III.B.2, NSs that can be searched by Google phrase searching are not used in SS-CAPTCHA. However, even if an NS cannot be searched by phrase searching (search result is 0), it may be possible to make a subtle difference between NSs and GSs by chopping sentences (NSs and GSs) into small parts (words or phrases) and comparing the search results of these small parts. Let us look at the two pairs of Japanese sentences.

NSa：私は今日中に提出しなければならない論文の執筆に追われています.

GSa：私は，今日の終わりまでに提出されるべきである論文を書くことによって，追いかけられます.

NSb：私はエアコンの風があまり好きではないので，窓を大きく開けて自然の風が通るようにしています.

GSb：私がエアコンの風があまり好きでないので，窓を大いに開けて，自然の風に通らされます.

None of the four sentences can be searched with Google phrase searching. However, by extracting characteristic small parts that make a human feel as though something were wrong from a GS and comparing the search results of the GS's small parts with those of the corresponding NS's small parts, these search results often contain subtle differences between an NS and a GS.

The two following small parts, GSa_p and GSb_p, are examples of small parts of GSs that make Japanese native speakers intuitively feel that something is wrong. The corresponding small parts of NSa_p and NSb_p have also been shown.

GSa_p :「提出されるべきである論文」

GSb_p:「窓を大いに開けて」

NSa_p:「提出しなければならない論文」

NSa_b:「窓を大きく開けて」

We can easily find that GSa_p and GSb_p cannot be searched by Google phrase searching while NSa_p and NSa_b can be. In general, there are more small parts that cannot be searched in GS than those in the corresponding NS. By doing this, it may be possible for a malicious automated program to discriminate NSs from GSs in a CAPTCHA test.

2) Possibility of Search-engine-based Attack

By extracting small strange parts from a GS and corresponding small parts from an NS and comparing the phrase search results of these small parts, it may be feasible to successfully carry out a search-engine-based attack. This means that, however, a search-engine-based attack needs both a GS and the corresponding NS. Therefore, in SS-CAPTCHA, a GS and the corresponding NS are not displayed simultaneously so that it is impossible for an automated program to find pairs of a GS and the corresponding NS on a given CAPTCHA test and to compare the phrase search results.

Moreover, since it is quite difficult for an automated program to implement (imitate) an intuitive sensory human process, an automated program cannot easily extract small strange parts from sentences. Table I lists the phrase-search results of small parts randomly extracted from NSa, NSb, Gsa, and GSb. The search results are widely varied regardless of whether the sentence is a GS or an NS. Hence, we expect that an automated program will find it quite difficult to deduce small strange parts according to the phrase-search results.

TABLE I. EXAMPLES OF PHRASE SEARCH RESULTS FOR SMALL PARTS

Small parts	Phrase search result	Source sentence of small parts
今日中に提出しなければ	15,300	NS_a
あまり好きでないので	99,100	GS_b
窓を大きく開けて	27,500	NS_b
論文を書くことによって	22,000	GS_a
自然の風が通るように	234	NS_b
今日の終わりまでに	73,000	GS_a

Furthermore, if an automated program tries to identify small strange parts from all sentences by brute force, it needs to send considerable numbers of search queries to an Internet search engine for each search-engine-based attack and cannot effectively carry out this attack. Thus, we consider that search-engine-based attacks do not pose practical threats on an SS-CAPTCHA and we have therefore omitted countermeasures against these in this paper.

V. USER STUDY

A. Experiment Objective

In this section, we discuss experiments we carried out to test and confirm the usability of SS-CAPTCHA. As we found that search-engine-based attacks did not pose practical threats to SS-CAPTCHA in the previous section, we only incorporated the countermeasure against convergence analysis attacks into SS-CAPTCHA.

B. Experimental Procedure

We used two types of NSs implemented in Sec.IV.A.1 in this experiment. The NSs were obtained by extracting sentences of adequate length (11–58 characters) from a Japanese newspaper (NS_1) and a book (NS_2). Two hundred sentences each were extracted from the newspaper and the book. Then, GSs were obtained with function F and NSs. We focused on Japanese as the language for pre-translated sentences and English as that for post-translated sentences and used a free translator, which is available on the Internet [9]. Thus, in this experiment we used two pairs of NSs and GSs; the first pair was NS_1 and GS_1 and the second pair was NS_2 and GS_2.

NS_1: NSs extracted from a newspaper

NS_2: NSs extracted from a book

GS_1: GSs generated from NS_1

GS_2: GSs generated from NS_2

As explained in Sec.III.B.3, SS-CAPTCA has a function to check CNSs (candidate NSs) and CGSs (candidate GSs) and eliminate inappropriate sentences. Therefore, in this experiment we eliminated inappropriate GSs in advance. We did not eliminate any NSs because we extracted NSs from a newspaper and a book, which are generally written grammatically and are not strange to humans. In this experiment, the three GSs were considered to be inappropriate.

GS-A: A GS that did not make a human feel as though something were wrong.

GS-B: A GS that included Roman letters not included in the corresponding original NS.

GS-C: A GS that included symbols not included in the corresponding original NS.

A GS-A is generated because the result of a machine translation is sometimes natural for humans. If GS-A is used as GS in SS-CAPTCHA, a human user will be confused about NS selection. GS-B and GS-C are generated because the corresponding original NS contains words or phrases that are not contained in the dictionary of the machine translator. Such words or phrases are usually replaced by Roman letters or symbols in machine translation. If GS-B and GS-C are used as GS in SS-CAPTCHA, it would become easier for an automated program to discriminate NSs from GSs. After these inappropriate GSs are eliminated, the number of GSs in GS_1 becomes 155 and that in GS_2 becomes 139.

The procedure for this experiment involved six steps:

1) The system randomly selected 5 NSs from NS_i (i=1, 2). Then, it calculated a mean value (M_{NS}) and standard deviation (σ_{NS}) for these 5 NS's C (the number of repetitions where function F was applied until $F(S_{i+1})$ equaled $F(S_i)$).

2) The system randomly selected 10 GSs from GS_i (i=1, 2). GSs were selected so that the mean (M_{GS}) and standard deviation (σ_{GS}) of these 10 GS's C were comparable to M_{NS} and σ_{NS} respectively ($|M_{GS}-M_{NS}| \leq \theta_M$, $|\sigma_{GS}-\sigma_{NS}| \leq \theta_\sigma$). In this experiment, θ_M and θ_σ were set to 0.5. A GS generated from an NS that had already been selected in step (1) was not selected for security reasons.

3) The system randomly presented all 15 sentences (5 NSs and 10 GSs) simultaneously in an authentication window.

4) An examinee was required to select 5 sentences that looked natural; in other words, that did not make him/her feel as though something were wrong.

5) If the examinee could select all 5 NSs correctly, he/she was certified to be human. If not, he/she was certified as not human.

6) Steps (1) to (5) were repeated 5 times for each NS_i (i=1, 2). That is, the examinee was required to repeat 10 tasks to select 5 sentences that looked natural in the 15 sentences. The system randomly selected i (i=1, 2) at each selection.

The examinees in this experiment were six volunteers who were college students. There is an example of the authentication window we used in this experiment in Fig. 9.

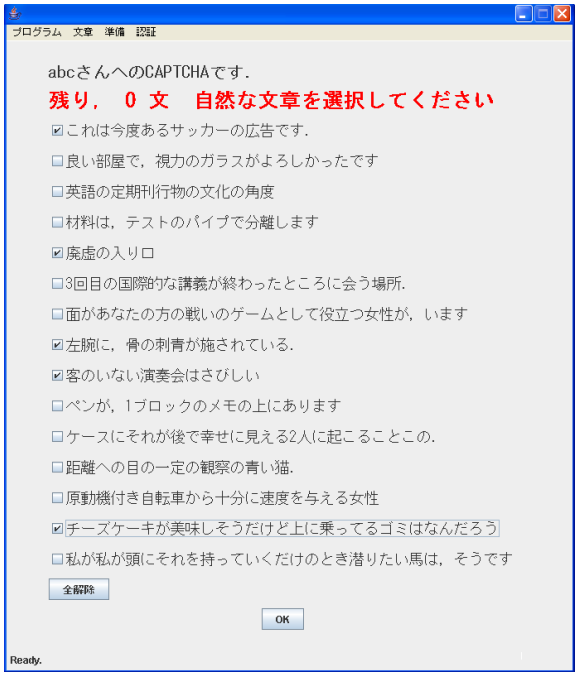

Figure 9. Example authentication window used in this experiment.

C. Results

Table II lists the results obtained from this experiment. Selection time means the average time the examinees required to finish choosing five sentences. Success rate means the rate of successful selection calculated according to each threshold, θ. Here, threshold θ is the number of acceptable missed selections. If θ increases, the usability by a human user will increase while the security of SS-CAPTCAH will decline. From the results, both NSs (NS_1 and NS_2) made examinees take a long time to select 5 sentences from 15. Moreover, the success rates were insufficient. We have to devise SS-CAPTCHA so that it will not confuse human users.

Some examinees commented to us that since the NSs in a newspaper (NS_1) were stiff and formal, it took them longer to read such sentences. We believe that people who often read newspapers or books written in stiff or formal style can easily read the sentences in them while those who often read magazines or cartoons written in casual or informal style are familiar with the sentences in these. Therefore, we should consider these human preferences and tastes to improve the usability of SS-CAPTCHA. The effectiveness of these extensions needs to be investigated in future work.

TABLE II. SELECTION TIME AND SUCCESS RATE

			NS_1	NS_2
Selection time [sec]			106.1	87.5
Success rate [%]	***Threshold θ***	***θ = 0***	63.3	56.7
		θ = 1	100.0	96.7
		θ = 2	100.0	100.0
		θ = 3	100.0	100.0

Examinees also commented that 15 sentences were too many to see in one view. Thus, we modified our system in a way where only one sentence (NS or GS) was presented on a screen. A user had to answer whether the presented sentences looked strange or natural (2-alternative selection). An additional experiment was carried out with the same examinees. Their selection times and success rates were measured. The procedure for this experiment involved six steps:

1) Five NSs were extracted from each NS_i (i=1, 2).
2) Five GSs were also extracted from each GS_i (i=1, 2). Therefore, the system had 20 sentences. The mean value and the standard deviation of the 10 NSs and 10 GSs were adjusted to equal each other as described Sec.V.B.
3) One sentence was randomly selected from the 20 sentences and presented to an examinee. A sentence that had already been selected was not re-selected.
4) The examinee was required to answer whether the presented sentence was strange or natural.
5) If the examinee could correctly answer strange to the presented GS or natural to the presented NS, he/she was certified to be human. If not, he/she was not certified to be human.
6) Steps (3) to (5) were repeated 20 times until all 20 sentences were used up.

TABLE III. SELECTION TIME AND SUCCESS RATE WITH 2-ALTERNATIVE SYSTEM.

	NS_1 - GS_1	NS_2 - GS_2
Selection time [sec]	7.4	6.7
Success rate [%]	90.0	100.0

Table III lists the results for this experiment, where the results are summarized according to (NS_1-GS_1) sentences and (NS_2-GS_2) sentences. Selection time means the average time examinees required to answer whether the presented sentence was strange or natural. Success rate means the rate of successful selection. We can also confirm from these results that sentences extracted from a newspaper (NS_1 and GS_1) are more difficult to read than those from a book.

In the previous experiment, the number of all possible combinations of selecting 5 from 15 sentences was 3003 ($_{15}C_5$) when θ equaled 0. For a comparison with the previous system, the user had to repeat 2-alternative selections at least 11 or 12 times. The number of all possible combinations of repeating 2-alternative selections 11 or 12 times corresponded to 2048 (2^{11}) or 4096 (2^{12}). Since all 2-alternative selections were almost independent of each other, a simple estimate of the selection time to repeat 2-alternative selections at least 11 or 12 times was that it would take 74–89 sec.

We do not believe these results were very reliable since this additional experiment was carried out informally, ignored the effect of order, and used the same sentences as in the previous experiment. However, most examinees commented that the 2-alternative system was easier to use even when they had to repeat the selections 11 or 12 times. Therefore it may be possible to further improve the usability of SS-CAPTCHA by adapting 2-alternative selection. We have left formal experiments for 2-alternative selection as future work and intend to explore further improvements to SS-CAPCTHA.

VI. CONCLUSION

We proposed a concept of a new CAPTCHA to utilize the human aptitude for recognizing "strangeness" as one of the more advanced human capabilities. As an example, we focused on strangeness in machine-translated sentences, and proposed a SS-CAPTCHA that could detect malware by checking if a user could distinguish natural sentences created by humans from machine-translated sentences. We discussed possible threats to the SS-CAPTCHA and countermeasures against these threats. Although we found that it was not very difficult for human users to solve the tests of SS-CAPTCHA, its usability was insufficient. We intend to explore further improvements to the SS-CAPCTHA in future work.

However, malicious users who attack today's CAPTCHAs are gradually changing from automated programs to human solvers. Such an attacker in the new threat attracts these human solvers by hosting his/her own porno sites or paying low wages. An automated program running on the attacker's sites is made

to access a victim Web server and obtains a CAPTCHA test from the victim. Then the automated program relays the CAPTCHA test to humans who visit the attacker's sites. The automated program obtains feedback from the human solvers, and sends it as a CAPTCHA response to the victim. By doing this, the automated program does not have to solve complicated Turing tests that need advanced human capabilities. These kinds of attacks are called relay attacks.

It is currently not easy for the SS-CAPTCHA to cope with these relay attacks. However, the need to overcome these attacks will surely increase in the near future. Therefore, we urgently have to tackle these kinds of problems in future work.

ACKNOWLEDGMENTS

We are deeply grateful to Dr. Keisuke Takemori of KDDI R&D Laboratories, Dr. Junji Nakazato of the National Institute of Information and Communications Technology, Dr. Satoru Torii of Fujitu Laboratories, and Mr. Kazuomi Oishi of Yokohama National University, who made valuable comments and suggestions to our study. This work was supported by a Grant-in-Aid for JSPS Fellows (No.20-6290) and by the Secom Science and Technology Foundation, Japan.

REFERENCES

[1] The Official CAPTCHA Site, http://www.captcha.net

[2] PWNtcha-Captcha Decoder, http://caca.zoy.org/wiki/PWNtcha

[3] K. Chellapilla, K. Larson, P. Y. Simard and M. Czerwinski, "Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs)," In Proceedings of the 2nd Conference on Email and Anti-Spam, 2005.

[4] J. Yan, A. S. E. Ahmad, "Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms," In Proceedings of the 2007 Computer Security Applications Conference, pp. 279–291, 2007.

[5] M. Chew and J. D. Tygar, M. Chew and J. D. Tygar, "Image recognition CAPTCHAs." In Proceedings of the 7th International Information Security Conference (ISC 2004), Springer, pp. 268-279, 2004.

[6] O. Warner, "Kittenauth," http://www.thepcspy.com/kittenauth.

[7] J. Elson, J. R. Douceur, J. Howell and J. Saul, "Asirra: a CAPTCHA that exploits interest-aligned manual image categorization," In Proceedings of the 2007 ACM CSS, pp. 366–374, 2007.

[8] P. Golle, "Machine Learning Attacks Against the ASIRRA CAPTCHA," In Proceedings of the 2008 ACM CSS, pp. 535–542, 2008.

[9] Excite translation, http://www.excite.co.jp/world/