2015 18th International Conference on Network-Based Information Systems

CAPTCHA System by Differentiating the Awkwardness of Objects

Michitomo Yamaguchi
Department of Mathematical Modeling, Analysis and Simulation, Graduate School of Advanced Mathematical Sciences, Meiji University, Tokyo, #164-8525 Japan
Email: yama3san@meiji.ac.jp

Takeshi Okamoto
Graduate School of Technology and Sciences, Tsukuba University of Technology, Ibaraki, #305-8521 Japan
Email: ken@cs.k.tsukuba-tech.ac.jp

Hiroaki Kikuchi
Department of Mathematical Modeling, Analysis and Simulation, Graduate School of Advanced Mathematical Sciences, Meiji University, Tokyo, #164-8525 Japan
Email: kikn@meiji.ac.jp

***Abstract*—The "Completely Automated Public Turing test to tell Computers and Humans Apart" (CAPTCHA) is a technique that prevents unauthorized access by bots. Most studies of CAPTCHA systems use human cognitive capacities as a countermeasure to facilitate recognition techniques. Differentiating between natural and awkward objects is an approach used to distinguish humans from bots. However, this approach is vulnerable to adversaries who exploit the differences in relative frequency between natural and awkward objects because of the difficulty in collecting natural objects. In this study, we propose a new scheme that does not require the utilization of natural objects, thereby addressing this shortcoming. Our proposed method requires that humans always distinguish awkward objects, which are generated by different parameters. We evaluated our scheme in several experiments.**

***Keywords*—CAPTCHA, Markov chain, Security analysis, Word salad**

I. Introduction

A. Background

The "Completely Automated Public Turing test to tell Computers and Humans Apart" (CAPTCHA) [1] is a technique for differentiating humans from software agents. Several online services utilize CAPTCHAs to combat malicious software agents, i.e., bots.

The most widely used method, visual-CAPTCHA (Figure 1), requires that users read distorted letters embedded in an image. Many online services also provide audio-CAPTCHA where users listen to distorted sounds. For public services, such as U.S. governmental sites [2], quiz-based CAPTCHA is used to avoid criticisms regarding accessibility. Thus, people with visual and auditory impairments can solve the quizzes because they only comprise text.

Fig. 1: Example of the Conventional CAPTCHA

The recognition of images and sounds and the completion of quizzes are considered to be difficult for bots; however, conventional visual-, audio-, and quiz-based CAPTCHAs may be insecure due to advances in recognition techniques [3], [4], [5], [6], [7]. A number of researchers have tried to overcome this issue by developing CAPTCHAs using new Artificial Intelligence (AI) problems [8], [9], [10], [11]. These CAPTCHA constructions require semantic interpretation because it is difficult for bots to solve questions based on the semantic interpretation of objects.

B. Difficulties: Problems of the Conventional Scheme

CAPTCHA systems require a large amount of objects (e.g., images, sounds, and sentences) to generate questions based on semantic interpretations. Several systems tend to utilize the objects directly when presenting questions to users; for example, the *KK*-scheme proposed by Kamoshida et al. [12] collects many natural sentences.

The *KK*-scheme is a CAPTCHA that exploits the awkwardness felt by humans when comparing a natural sentence and a "word salad" synthesized by a Markov chain. The scheme generates questions that comprise natural sentences and word salads, and users are required to distinguish them. Humans can interpret the semantic meaning and distinguish sentences with considering the difference in terms of "naturalness." By contrast, it is difficult for bots to interpret these differences because naturalness is ill-defined.

The *KK*-scheme requires source documents (i.e., a corpus) to generate questions, but this leads to two problems.

(1) Difficulty collecting natural sentences: The amount of natural sentences is limited because parts of sentences appear in public corpora. Suppose that we construct natural sentences and word salads from a corpus according to the *KK*-scheme. We could generate an excessive number of unique word salads because word salads comprise a combination of words that follows a Markov chain. On the other hand, depending on the size of the corpus, we might generate questions that have already been used as natural sentences. Thus, adversaries can differentiate natural sentences from word salads based on

978-1-4799-9942-2/15 $31.00 © 2015 IEEE
DOI 10.1109/NBiS.2015.114

whether the sentence in the question is found in a database of previous questions.

(2) Vulnerability to attacks using search engines: Adversaries can employ a large corpus such as online documents, and thus we must pay special attention to security against attacks using search engines. Adversaries can query the sentence given in the question using search engines and solve the CAPTCHA based on the results obtained.

Therefore, objects that are exactly the same as those found in the sources will have security issues. We must emphasize that these problems are common in schemes that use such objects in questions.

C. Our Objective and Approach

In this study, we propose a new CAPTCHA that overcomes the problems described in Section I-B.

Our approach has the following features.

- In our scheme, all of the questions are generated as awkward objects by programs. Our scheme does not use original objects in questions, thereby protecting against search engine attacks.
- Our scheme presents pairs of awkward objects, which are generated using different generation parameters. We expect that humans will be able to perceive the difference in terms of awkwardness by comparing the two objects. This is useful for improving the human accuracy rate, although no original objects are used.

D. Our Contributions

The main contributions of our study are as follows.

- *Proposal of an actual scheme:* We describe the construction of an actual new scheme.
- *Superiority of our scheme to the KK-scheme:* We demonstrate that our scheme performs better than the *KK*-scheme based on the following points.
  - Preservation of a high accuracy rate for humans.
  - Security against several attack types: 1) Automated proofreading attacks (MS Word) [13]; 2) Attacks based on different appearance rates in natural sentences and word salads; 3) Attacks using search engines.

  We illustrate the security weakness of the *KK*-scheme against attacks.
- *Optimal Markov chain model for alternative natural sentences:* Based on the experiments in (1)–(3) using different orders, we clarified the best parameters for the Markov chain model.

II. Related Work

Many CAPTCHAs have been proposed that use semantic interpretation.

For example, Asirra [14] employed semantic image interpretation, where the proposed method presents pictures of dogs and cats, and users are required to categorize them. The methods proposed by Holman et al. [15] and Shirali-Shahreza et al. [16] present several pictures and users need to select a related object from several choices. Similar to semantic image interpretation, several methods use semantic sound interpretation. Researchers also treat contextual cognition as a type of AI problem. Park et al. [17] showed that in a limited situation where it is necessary to distinguish phishing emails from legitimate emails, bots cannot identify certain types of emails whereas humans can identify them easily.

Ivey [18] proposed a method that presents common topic sentences except for one and users are required to answer with an odd one. Goto [19] proposed a CAPTCHA based on a phonemic restoration effect and similar pronunciation. Yamamoto et al. [20] and Kamoshida et al. [12] utilized the strangeness between natural sentences and machine-generated sentences as a CAPTCHA, where the former uses repeated machine translation to generate awkward sentences and the latter uses word salads.

We describe the algorithm for the *KK*-scheme as follows.

Algorithm for the *KK*-scheme [12]

1) A CAPTCHA system constructs a Markov chain model of order N from a corpus.
2) The system extracts h natural sentences from the corpus and synthesizes s word salads using the model.
3) The system sorts the order of z $(= h + s)$ sentences randomly and presents them to a user.
4) The user answers each sentence with *Ham* or *Spam* for z sentences.
5) The system checks the response and judges that the user is a human if $k \geq \theta$, where k is the number of correct answers and θ is a threshold.

III. Preliminary

A. Markov Chain

Let N be the order of a Markov chain. A Markov chain of order N is a process that satisfies

$$\begin{aligned} & P(X_{n+1} = x | X_n = x_n, \ldots, X_0 = x_0) \\ = \; & P(X_{n+1} = x | X_n = x_n, \ldots, X_{n-N+1} = x_{n-N+1}). \end{aligned}$$

We apply morphological analysis [21] to a corpus and obtain N-grams. We then construct a model that follows a Markov chain of order N. In the model, the future state $N+1$ depends on the previous N states.

B. Ham and Spam

We denote that a sentence synthesized by a Markov chain of order N is an order-N word salad. Our scheme uses word salads for both *Ham* and *Spam*. Let *Ham* and *Spam* be a semantic natural sentence and a semantic awkward sentence, respectively, which are order-N_{Ham} and order-N_{Spam} word salads, where $N_{Ham} > N_{Spam}$. It should be noted that in our scheme, the naturalness of sentences is a relative criterion based on a comparison between *Ham* and *Spam*.

C. Diversity of Synthesized Sentences

Suppose that we synthesize W_A word salads. Let W_U/W_A be the diversity of the synthesized sentences if W_U among W_A word salads are unique. We employ this as a criterion that determines how the scheme generates unique questions for the CAPTCHA.

D. Diversity of a Corpus

Let $\mathcal{D}_M$ be a set of M-grams that appear in a corpus. Let A be an M-gram word, where $A = (a_n a_{n-1} \dots a_{n-M+1}) \in \mathcal{D}_M$. Let $C_{(M,A)}$ be a candidate set of $(M+1)$-th words chained for an M-gram that follows a Markov chain, i.e.,

$$C_{(M,A)} = \{c \in \mathcal{D}_1 \mid P(X_{n+1} = c | X_n = a_n, \dots, X_{n-M+1} = a_{n-M+1}) > 0\}$$

Let C_M be the diversity of a corpus of M-grams, where $C_M = \sum_{A \in \mathcal{D}_M} |C_{(M,A)}|/|\mathcal{D}_M|$. We employ this as a feature of a corpus.

E. Evaluation Criteria

Let X and Y be random variables for a sentence in a query and the response, respectively, where the value of S denotes *Spam* and H denotes *Ham*. The probabilities of a query being *Ham* or *Spam* are $P(X = H) = h/z$ and $P(X = S) = s/z$, respectively. The CAPTCHA outcome has a joint probability $P(X, Y)$, which is computed by

$$\begin{aligned} P(Y = H, X = H) &= P(Y = H | X = H) P(X = H), \\ P(Y = S, X = H) &= P(Y = S | X = H) P(X = H), \\ P(Y = H, X = S) &= P(Y = H | X = S) P(X = S), \\ P(Y = S, X = S) &= P(Y = S | X = S) P(X = S). \end{aligned}$$

A failure case is represented either by a response of S to a given query of H, or a response of H to a given query of S. Let Y_h and Y_m be the responses by humans and bots, respectively. The failure probability for a CAPTCHA by a human (P_q) is defined as

$$P_q = P(Y_h = S, X = H) + P(Y_h = H, X = S).$$

Similarly, the success probability for a CAPTCHA by bots (P_m) is defined as

$$P_m = P(Y_m = S, X = S) + P(Y_m = H, X = H). \quad (1)$$

In Eq. (1), we consider that adversaries utilize several tools such as proofreading and search engines. Let W be a random variable for a process used by a tool, which can have values t or f. An event $W = t$ represents a process that detects certain information and $W = f$ represents an event that does not occur. The right-hand side of Eq. (1) is computed by

$$\begin{aligned} P(Y_m &= S, X = S) \\ &= P(Y_m = S | W = t) P(W = t | X = S) \\ &\quad + P(Y_m = S | W = f) P(W = f | X = S), \end{aligned}$$

$$\begin{aligned} P(Y_m &= H, X = H) \\ &= P(Y_m = H | W = t) P(W = t | X = H) \\ &\quad + P(Y_m = H | W = f) P(W = f | X = H). \end{aligned}$$

Questions: Choose more awkward sentence from 'A' and 'B'.

No. 1 ($N_{Ham} = 2, N_{Spam} = 1$):

A	There is mainly something used for exams of universities. Instruments for house-moving, furniture
B	I said outside the swimsuit of the eyebrows, the strangeness was done and returned to here

No. 2 ($N_{Ham} = 3, N_{Spam} = 1$):

A	I heard that his friend would come here. That's troubling and nonsense. Now
B	Because it becomes hard, I gain it for illness, look back, and Miwa has no interest in it before

No. 3 ($N_{Ham} = 4, N_{Spam} = 1$):

A	Due to the bad system, the guides are required to be kind, polite and shrewd. However
B	Sitting straight in the zoo, the lineage of Ogaki, Kuniko and Aoba, is serious

Answers: 'B' is a *Spam* for all questions.

Fig. 2: Sentences Synthesized by Our Proposed Method.

We define the false human rejection ratio (*FRR*) as the probability that a human correctly solves k CAPTCHA questions s.t. $k < \theta$. Similarly, the false machine acceptance rate (*FAR*) is defined as the probability that bots correctly solve k CAPTCHA questions s.t. $k \geq \theta$. *FRR* and *FAR* are given by the binominal distribution as follows.

$$FRR = \sum_{k=\theta}^{z} \binom{z}{k} P_q^k (1 - P_q)^{z-k}, \quad FAR = \sum_{k=\theta}^{z} \binom{z}{k} P_m^k (1 - P_m)^{z-k}$$

In this study, we employ the F-ratio as an evaluation criterion, which is given by

$$F = \frac{2 \cdot (1 - FAR) \cdot (1 - FRR)}{(1 - FAR) + (1 - FRR)}, \quad (2)$$

where *FRR* and *FAR* are computed s.t. $\theta = 1$, $z = 1$.

IV. Our Proposed Method

A. Outline

In the following, we demonstrate that there are two differences between our proposed method and the *KK*-scheme.

(1) How to Generate Ham: Our system synthesizes *Spam* using a Markov chain of order N_{Spam}. *Ham* is also synthesized by a Markov chain of order N_{Ham} s.t. $N_{Ham} > N_{Spam}$.

To ensure security, it is useful to employ word salads as *Ham* because of the following features of word salads compared with natural sentences.

- We can generate a large amount of sentences from a corpus.
- It is difficult for search engines to detect a corpus because the exact same sentences will not be present in the corpus.

TABLE I: Features of Our Corpus: (Numbers of Characters, Lines) = (80783, 5248).

N-gram	1	2	3	4	5	6	7
Number of Unique Words	7,893	34,469	60,790	73,632	77,532	77,526	76,395
Diversity of the Corpus (C_N)	4.403	1.785	1.231	1.075	1.023	1.008	1.002

(2) Answering Method: We are concerned with the lower accuracy rate obtained by humans because order-N_{Ham} word salads are more awkward than natural sentences. As a countermeasure, our scheme presents pairs of *Ham* and *Spam* for each question. Users then solve this problem by comparing two sentences in terms of their relative naturalness.

Figure 2 shows examples of our questions. In our opinion, humans cannot detect sufficient naturalness with a single *Ham* in examples 1 and 2. However, we consider that humans can detect differences in awkwardness based on a relative comparison of a *Ham* and *Spam* pair.

B. Our Scheme

The algorithm for our proposed scheme is as follows.

Algorithm for our Proposed Scheme

1) A CAPTCHA system chooses two orders s.t. $N_{Ham} > N_{Spam}$ and generates Markov chain models.
2) The system synthesizes z pairs of *Ham* and *Spam*.
3) The system exclusively randomly assigns *Ham* and *Spam* to choose 'A' and 'B' for z pairs.
4) The system presents z pairs as questions and a user must choose the most natural or awkward option.
5) The user answers each question with 'A' or 'B' for z pairs.
6) The system checks the response and judges that the user is a human if $k \geq \theta$, where k is the number of correct answers and θ is a threshold.

V. Evaluation

A. Evaluation Items

We conducted the following experiments to verify the effectiveness of our proposed method.

- Experiment 1: How much is the diversity of the synthesized sentences?
- Experiment 2: How often can search engines find the corpus of sentences?
- Experiment 3: How high is the accuracy at which subjects solve our proposed scheme?

B. Experimental Methods

Standard Settings: We describe the standard settings for our experiments as follows.

- We selected five types of Japanese documents from a public-domain book site called "Aozora-bunko" as a corpus. Table I shows the features of this corpus.
- We employed an order-N_{Spam} word salad as *Spam*, where $N_{Spam} = 1$. We employed an order-N_{Ham} word salad as *Ham*, where $N_{Ham} = 1, 2, 3, 4, 5$.
- In this study, we used an order-7 word salad as a natural sentence because Table I shows that the diversity of the corpus was 7-grams, i.e., $C_{N=7}$, which is almost equal to 1. Similarly, we assumed that the *KK*-scheme allowed users to distinguish two word salads synthesized with Markov order $N_{Ham} = 7$ and $N_{Spam} = 1$.
- A word salad comprised 30–40 characters.

Experiment 1: We synthesized $50,000$ word salads for each order and investigated the diversity of the synthesized sentences.

Experiment 2: We synthesized 100 word salads for each order. Each word salad was queried using the Yahoo! search engine. We assumed that the search engine had found the corpus if it occurred within the top 10 search results.

Experiment 3: We asked 16 Japanese subjects (13 men and three women) to answer our questions. For each $N_{Ham} = 2, 3, 4, 5$, the questions were generated as follows.

1) We randomly selected 10 order-N_{Ham} word salads from those synthesized in experiment 2. Similarly, we selected 10 order-N_{Spam} word salads from those synthesized in experiment 2. Note that $N_{Spam} = 1$.
2) We randomly selected a *Ham* and a *Spam*. We repeated this step 10 times and obtained 10 pairs.
3) Let 'A' and 'B' be choices. For each pair, we exclusively randomly assigned *Ham* and *Spam* to either 'A' or 'B', respectively.

C. Experimental Results

TABLE II: Diversity of the Sentences Synthesized by a Markov Chain.

Order N	1	2	3	4	5	6	7
Diversity	1.000	0.999	0.942	0.732	0.522	0.480	0.437

Experiment 1: Table II shows the diversity of the synthesized sentences, which demonstrates that the diversity of word salads was high compared with natural sentences (i.e., order-7 word salads), especially in the case where $N < 4$. A change in the value had a dramatic effect at $N = 4$, whereas the change was slow above this value. It is likely that $C_{N \geq 4} \approx 1$ was satisfied by the corpus in Table I.

TABLE III: Conditional Probability of a Sentence Being Detected.

Order N	1	2	3	4	5	7
$P(W = t\|X = x)^{\dagger}$	0.12	0.19	0.44	0.78	0.85	0.89

†: If $N = 1$, then $x = S$. Otherwise, $x = H$.

Experiment 2: Table III shows the detection rate of the corpus for each word salad using the Yahoo! search engine.

Table III shows that it was difficult for the search engine to find the corpus of word salads compared with the corpus of natural sentences, especially when $N = 2, 3$. Similar to the diversity of synthesized sentences, it is likely that the results were affected by the features of the corpus.

TABLE IV: Failure Rate by Humans.

Order $N^{\dagger}_{Ham}$	2	3	4	5	7
FAR	0.194	0.212	0.169	0.156	0.18

†: Order $N_{Spam} = 1$.

Experiment 3: The human failure rate results for each question (P_q) are presented in Table IV. It should be noted that the result for $N_{Ham} = 7$ was derived from [12].

Table IV shows that the human failure rate was 18% for the *KK*-scheme and 15.6–21.2% for our proposed method. These results suggest that the human failure rate does not depend greatly on the order of the Markov chain. Before this experiment, we considered that humans were affected more greatly by the difference in the order. However, these results indicate that the human ability to recognize the naturalness of sentences was higher than we expected.

VI. Considerations

A. Vulnerability of the KK-scheme

Kamoshida et al. [13] considered two types of attacks: a brute force attack and a word attack, where their results clearly demonstrated that word attack using the proofreading functions of MS Word was more powerful than the brute force attack. Thus, the results of their security analysis were based on word attack.

In this section, we consider two new attacks using the diversity of synthesized sentences and search engines, and we demonstrate that the *KK*-scheme is not secure against these attacks.

The success rate for adversaries depends on s and h because an attacker may be good at detecting *Ham* whereas another may be better at differentiating *Spam* rather than *Ham*. We present calculation methods based on security analysis with $s = 5$, $h = 15$ ($z = h + s = 20$) as examples.

First, we review the word attack calculation method. Second, we analyze new attacks in the same manner.

Word Attack: We provide a method for calculating P_{mw}, which is the success rate of word attacks. Let W_w be a random variable, with values of t_w or f_w. An event $W_w = t_w$ denotes that MS Word detects incorrect phrases in sentences X for a question and $W = f$ denotes that the event does not occur. We have $P(W_w = t_w|X = S) = 0.24$, $P(W_w = t_w|X = H) = 0$, which are derived from the results described in [13]. We have $P(W_w = t_w) = 0.06$, $P(W_w = f_w) = 0.94$ because $P(X = S) = 0.25$, $P(X = H) = 0.75$. In this attack, adversaries always answer the question with *Spam* if MS Word finds the wrong phrases in the sentences; otherwise, they answer with *Ham* at a probability of $P(X = H|W_w = f_w) = 0.798$ or with *Spam* at a probability of $P(X = S|W_w = f_w) = 0.202$. Thus, we obtain $P(Y_w = H, X = H) = 0.798$, $P(Y_w = S, X = S) = 0.394$ using these values. Consequently, we obtain $P_{mw} = 0.697$ using Eq. (1).

Attack Based on the Diversity of Synthesized Sentences: In this attack, adversaries check the sentences in a question to determine whether they are the same as those in previous questions. It is useful for adversaries to employ this type of attack if there is a clear difference in the diversity of the synthesized sentences between *Ham* and *Spam*, especially when the corpus is small.

We provide a method for calculating P_{md}, which is the success rate for an attack based on the diversity of synthesized sentences. Let W_d be a random variable, with values t_d or f_d. An event $W_d = t_d$ denotes that adversaries find sentences X in a question based on previous questions and $W_d = f_d$ denotes that the event does not occur. We obtain $P(W_d = t_d|X = S) = 0$, $P(W_d = t_d|X = H) = 0.563$ from the results of experiment 1 based on $N = 1, 7$. In this attack, adversaries always answer the question with *Ham* when $W_d = t_d$; otherwise, they answer with *Ham* at a probability of $P(X = H|W_d = f_d) = 0.567$ or with *Spam* at a probability of $P(X = S|W_d = f_d) = 0.433$. Consequently, we obtain $P_{md} = 0.716$ in the same manner as the word attack calculation.

Attack Using Search Engines: In this attack, adversaries query the sentences in a question using search engines and compare the differences in the results obtained for *Ham* and *Spam*. It is useful for adversaries to employ this type of attack if *Ham* is the same in parts of public documents.

We provide a method for calculating P_{ms}, which is the success rate for an attack using search engines. Let W_s be a random variable, with values of t_s or f_s. An event $W_s = t_s$ denotes that a search engine detects a source for the sentences X in a question and $W_s = f_s$ denotes that the event does not occur. Thus, we obtain $P(W_s = t_s|X = S) = 0.12$, $P(W_s = t_s|X = H) = 0.89$ from the results of experiment 2 based on $N = 1, 7$. In this attack, when $W_s = t_s$, adversaries answer with *Ham* at a probability of $P(X = H|W_s = f_s) = 0.957$ or with *Spam* at a probability of $P(X = S|W_s = f_s) = 0.043$; otherwise, they answer with *Ham* at a probability of $P(X = H|W_s = f_s) = 0.273$ or with *Spam* at a probability of $P(X = S|W_s = f_s) = 0.727$. Consequently, we obtain $P_{ms} = 0.823$ in the same manner as the word attack calculation.

Comparison between Several Attacks: For all the pairs of (s, h), we calculated the success rate of each attack as described in this section. The results are shown in Figure 3 and demonstrate that the minimum values of the rate are (P_{mw}, P_{md}, P_{ms}) = (0.567, 0.683, 0.796). Hence, attacking with search engines is the most powerful strategy against the *KK*-scheme.

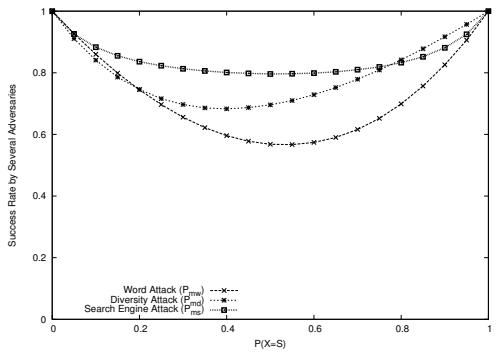

Fig. 3: Attack Success Rate Given Known Probability of *Spam* $P(X = S)$ in *KK* scheme.

TABLE V: Comparison between the *KK*-Scheme and Our Proposed Method for each Question.

Scheme	N_{Ham}	*FRR*	*FAR*	F-ratio
KK	7	0.180	0.796	0.33
Our Proposal	2	0.194	0.505	0.61
	3	0.212	0.563	0.56
	4	0.169	0.720	0.42
	5	0.156	0.767	0.37

B. Optimal Markov Chain Model and Comparison with KK-Scheme

We compared our proposed method with the *KK*-scheme based on the F-ratio calculated using Eq. (2). We utilized *FRR* from the results in Table IV. The success rate of the attack using search engines employed *FAR* based on the results in Section VI-A.

Table V shows that our proposed method had the best F-ratio when $N_{Ham} = 2$ and it performed better than the *KK*-scheme.

It should be noted that Table V shows the results obtained for one question. $FAR > FRR$ was satisfied, so we applied the gap amplification technique [1] to our scheme, where users must solve multiple questions to improve the security of the scheme.

VII. Discussions

A. Response Time

The response time of the *KK*-scheme is relatively high compared with other visual CAPTCHAs [12]. However, our scheme requires that users read two sentences for each question, and thus the response time is longer compared with the *KK*-scheme. We will address this shortcoming in future research. For a simple example, it may be more efficient to reduce the number of characters in the word salad.

B. Difference in the Orders between N_{Spam} and N_{Ham}

In this study, we fixed N_{Spam} as 1. We then modified N_{Ham} to investigate its effects in several experiments. However, it is possible that our scheme will also work with $N_{Spam} \neq 1$ s.t. $N_{Ham} - N_{Spam} > 0$.

In general, humans are not good at reading awkward sentences, which yields a longer response time and a reduced accuracy rate for humans. This might be improved by using two types of word salads synthesized with relatively large orders.

C. Adaptation of Our Approach to Other AI Problems

Our approach requiring users to distinguish different awkward objects could be applied to other AI problems.

It is known that cognitive bias often affects human decision making. In methods that require users to identify one object, e.g., the *KK*-scheme, human answers are affected by previous questions due to biases, such as anchoring and adjustment, confirmation bias, conservation, and confirmation bias. Factors such as logical fallacy, priming, and fake familiarity [22] also cause human errors.

Thus, we could prepare the same questions but sort the order in which they are presented to users. It is reasonable to expect that there will be differences in the answers provided even though the questions are the same. For example, humans might identify an awkward object as a natural object if an extreme awkward object is presented in the previous question.

However, although we only used awkward objects, our proposed method prevented a decline in *FRR*, which may be explained by the relative comparison restricting these different types of bias within one question.

If we consider *FAR*, then due to the benefits in terms of *FRR*, our proposed method can employ more indistinct objects as questions. This would be useful for avoiding pattern matching and search attacks, thereby making it difficult for adversaries to break our scheme.

We think that these beneficial features are widely accepted in other CAPTCHAs, which can control the awkwardness of the objects generated. During 3D object extraction, speech synthesis, and machine translation, we could control their precision in these processes. It may be useful to consider applying our approach to these processes and employing them as CAPTCHA questions.

VIII. Conclusion

In this study, we proposed a new CAPTCHA system based on differences in the awkwardness of objects. Our approach involves a relative comparison of objects in terms of their awkwardness. We showed that this feature was effective in improving security against attacks using databases and search engines.

First, we analyzed the security of the *KK*-scheme and highlighted its problems. Second, we constructed the actual scheme and confirmed its performance in several experiments. Thus, the comparisons between our scheme and the *KK*-scheme demonstrated the advantages of our method in terms of the F-ratio.

References

[1] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *Proceedings of EUROCRYPT*, vol. 2656. Springer-Verlag, 2003, pp. 294–311.

[2] T. W. House. (2014) We the people: Your voice in our government. [Online]. Available: https://petitions.whitehouse.gov/

[3] E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell, "The end is nigh: Generic solving of text-based CAPTCHAs," in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/woot14/workshop-program/presentation/bursztein

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

[5] E. Bursztein, R. Beauxis, H. S. Paskov, D. Perito, C. Fabry, and J. C. Mitchell, "The failure of noise-based non-continuous audio CAPTCHAs," in *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22–25 May 2011, Berkeley, California, USA*, 2011, pp. 19–31.

[6] J. Tam, J. Simsa, S. Hyde, and L. von Ahn, "Breaking audio CAPTCHAs," Advances in Neural Information Processing Systems 21. MIT Press, 2008, pp. 1625–1632.

[7] Y. Michitomo, N. Toru, W. Hajime, O. Takeshi, and K. Hiroaki, "Vulnerability of the conventional accessible Captcha used by the White House and an alternative approach for visually impaired people," in *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2014, Paper–ID 1729.

[8] S. A. Ross, J. A. Halderman, and A. Finkelstein, "Sketcha: A captcha based on line drawings of 3D models," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 821–830.

[9] M. Imsamai and S. Phimoltares, "3D CAPTCHA: A next generation of the CAPTCHA," in *Proceedings of Information Science and Applications 2010*, ser. ICISA '10. IEEE, 2010, pp. 1–8.

[10] P. Qvarfordt, E. Rieffel, and D. Hilbert, "Motion and interaction based CAPTCHA," Dec 2013, US Patent 8,601,538. [Online]. Available: http://www.google.com/patents/US8601538

[11] J. Kani, T. Suzuki, A. Uehara, T. Yamamoto, and M. Nishigaki, "Four-panel cartoon CAPTCHA," *IPSJ Journal*, vol. 54, no. 9, pp. 2232–2243, Sep 2013.

[12] Y. Kamoshida and H. Kikuchi, "Word salad CAPTCHA - application and evaluation of synthesized sentences," *15th International Conference on Network-Based Information Systems*, vol. 0, pp. 799–804, 2012.

[13] Y. Kamoshida and H. Kikuchi, "Proposal of CAPTCHA using artificial synthesis sentences and its security evaluation (Japanese)," *IPSJ Journal*, vol. 54, no. 9, pp. 2156–2166, 2013.

[14] J. Elson, J. R. Douceur, J. Howell, and J. Saul, "Asirra: a CAPTCHA that exploits interest-aligned manual image categorization," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. ACM, 2007, pp. 366–374.

[15] J. Holman, J. Lazar, J. H. Feng, and J. D'Arcy, "Developing usable CAPTCHAs for blind users," in *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2007, pp. 245–246.

[16] S. Shirali-Shahreza, G. Penn, R. Balakrishnan, and Y. Ganjali, "Seesay and hearsay CAPTCHA for mobile interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. ACM, 2013, pp. 2147–2156.

[17] G. Park, L. M. Stuart, M. Taylor, and V. Raskin, "Comparing machine and human ability to detect phishing emails," in *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2014, Paper–ID 1956.

[18] C. Liam, "System and method for delivering a human interactive proof to the visually impaired by means of semantic association of objects," US Patent Application 20120232907, 2012.

[19] M. Goto, T. Shirato, and R. Uda, "Text-based CAPTCHA using phonemic restoration effect and similar sounds," in *IEEE 38th Annual Computer Software and Applications Conference, COMPSAC Workshops 2014*, 2014, pp. 270–275.

[20] T. Yamamoto, J. Tygar, and M. Nishigaki, "CAPTCHA using strangeness in machine translation," *2013 IEEE 27th International Conference on Advanced Information Networking and Applications*, vol. 0, pp. 430–437, 2010.

[21] T. Kudo, "MeCab : Yet another part-of-speech and morphological analyzer," *http://mecab.sourceforge.net/*. [Online]. Available: http://ci.nii.ac.jp/naid/10019716933/

[22] T. Stafford and M. Webb, *Mind Hacks*. O'Reilly, 2004.