マウストラッキングを用いた CAPTCHA 方式の検討

立田　怜平 a)・山場　久昭 b)・久保田　真一郎 c)・岡崎　直宣 d)

A Study on the CAPTCHA Using Mouse Tracking

Ryohei TATSUDA, Hisaaki YAMABA, Shin-Ichiro KUBOTA, Naonobu OKAZAKI

Abstract

CAPTCHAs, which are reverse Turing tests, are used in many websites in order to guard them from bots attacks. However, there are many methods for breaking CAPTCHAs. Relay attack is one of such methods solving CAPTCHA using human solvers. We propose a CAPTCHA using mouse tracking to resist relay attack. We used delay time that is caused by communications needed in relay attack. We constructed an experimental enviroment that can simulate relay attack. A series of experiments was carried out to evaluate the performance of the proposed method.

Keywords: CAPTCHA, relay attack, mouse tracking

1. はじめに

近年、無料メールサービスなどの Web サービスに対し、ボットと呼ばれる自動プログラムを用いたアカウントの大量取得や、それらを用いた SPAM メール送信などの不正行為が問題視されている。

このような問題の防止を目的に、人間とボットを識別するための CAPTCHA と呼ばれる方式が開発された [1]。これは、人間には容易に解答できるがコンピュータには判別が困難である問題をユーザーに出題し、正解できたユーザーを人間と判断する技術である。

例えば、代表的な CAPTCHA に文字列 CAPTCHA がある。文字列 CAPTCHA はノイズや歪曲を付加した文字列の画像をユーザーに提示し、その画像（以下、CAPTCHA の問題画像と呼ぶ。）に表示されている文字列を正しく答えられるかで人間かボットか判断する。しかし近年では、CAPTCHA を突破する攻撃手法が登場してきている。既に、文字認識技術やパターン分類技術の発達によって文字列 CAPTCHA や画像型 CAPTCHA などの代表的な CAPTCHA は容易に突破されるようになってきており、その脆弱性が多くの研究者に指摘されている。

a)工学専攻大学院生
b)情報システム工学助教
c)情報システム工学准教授
d)情報システム工学教授

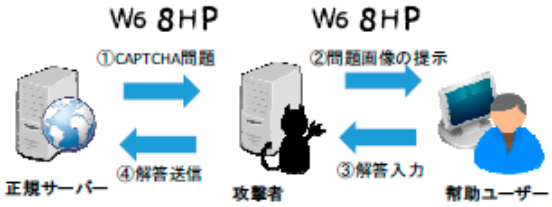

図 1.　リレーアタックの一例.

特にリレーアタックと呼ばれる攻撃手法は、インターネット上の一般ユーザーや低賃金労働者に CAPTCHA の問題画像を中継してそれを解読させる攻撃手法であり、人間が CAPTCHA の解読を行うため、プログラムを想定した対策では効果がなく、新たな対策が求められている。

そこで本論文では、リレーアタックを行った時に生じる通信の遅延時間に着目し、リレーアタックに耐性を持たせた CAPTCHA 方式を提案する。提案方式は、動的な CAPTCHA であり、連続的に移動してその位置を変化させる移動オブジェクトをランダムな位置に出現する複数の妨害オブジェクトの中から認識し、それを、マウスカーソルで追跡できるか否かによってアクセスしてきている者が人間か自動プログラムかを判別する。本手法は、この動的な性質に加え、リレーアタックで生じる CAPTCHA の問題画像を中継した際の遅延を利用して、リレーアタックの防止を図っている。

2. リレーアタック

2.1 リレーアタックとは

リレーアタックは、攻撃者が正規サイトからCAPTCHAの問題画像を取得し、第三者の人間にそれを中継して解答してもらい、その解答を正規サイトに送ることCAPTCHAを突破する手法である。CAPTCHAの問題画像の取得や第三者への中継などは、攻撃者の作成したプログラムで自動的に行われる。リレーアタックの具体的な手法には、攻撃者が運営するサイト（以下、リレーサイトと呼ぶ。）にインターネット上の一般ユーザーが訪問して来たら、正規サイトから取得してきたCAPTCHAの問題画像を提示し、リレーサイトのコンテンツを閲覧することと引き換え解読させる手法や、攻撃者が低賃金労働者を雇うCAPTCHAの問題画像を労働者に転送し、報酬を与えて解読させる手法などがある。以降の説明では、リレーアタックに際して、CAPTCHAの解答を提供する者（インターネット上の一般ユーザーや低賃金労働者）を幇助ユーザーと呼ぶことにする。

2.2 リレーアタック対策

この節では、既存のリレーアタック対策やリレーアタックに耐性を持つCAPTCHAについて述べる。

2.2.1 IPアドレスの違いを利用した対策

鈴木等はリレーアタックの特徴、すなわち、正規サイトにアクセスするPCとリレーサイトで中継されるCAPTCHAを解くPCとが異なっていることを利用し、リレーアタックが行われていることを検知する手法を提案している[2]。しかしこの手法ではリレーアタックを検知するために必要なIPアドレスを正規サーバーに通知する必要がある。そのため、そのプログラムを一般ユーザーのPCにインストールしなければならない。よって、低賃金労働者を利用したリレーアタックのように、不正であることを知った上でリレーサイトにアクセスしてくるユーザーがいる場合には、このプログラムをインストールしないことによって対策の回避が可能となってしまう。

2.2.2 DCG-CAPTCHA

DCG-CAPTCHA[3,4]は簡単なミニゲーム形式のCAPTCHAである。ユーザーが与えられた指示に適するオブジェクトをマウスなどで選択し、その選択が正しければ、人間とみなすものである。DCG-CAPTCHAのオブジェクトは常に移動しているので、ユーザーが解答を行う際のリアルタイム性に着目し、ユーザーとCAPTCHAとの間のインタラクションのタイミングを検査することでリレーアタックの検出を実現している。しかし、DCG-CAPTCHAは自動プログラムに対する耐性が低いという欠点がある。

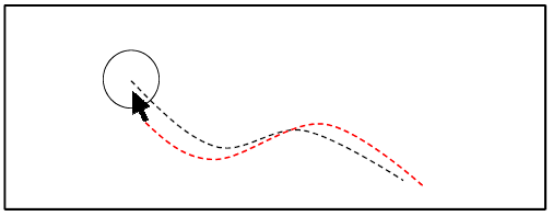

図2. 提案CAPTCHAの例.

3. 提案手法

3.1 目的と提案CAPTCHA

本研究では、リレーアタックに対する耐性を持ちながら、一般ユーザーの協力的な行為も必要とせず、プログラムによる自動化攻撃にも耐性を持つCAPTCHAを作成することを目的とする。2.2で述べたように、リレーアタックに対する対策はいくつか考えられているが、それらの手法は、一般ユーザーの協力的な行為を必要としたり、自動プログラムへの耐性が弱くなったりしている。

提案するCAPTCHAは、動的なCAPTCHA方式であり、図2に示すように移動する円形のオブジェクト（以下、移動オブジェクトと呼ぶ。）をマウスカーソルで追跡できるか否かで解答者が人間かボットかを判断するものである。次節にて、提案CAPTCHAのリレーアタックとプログラムによる攻撃への対応について示す。

3.2 想定される攻撃に対する耐性

3.2.1 リレーアタックへの対応

図3にCAPTCHAに対してリレーアタックを行ったときの通信についてのシーケンス図を示す。図3で用いている記号の意味を以下に示す。

Ox_t , Oy_t ：時間 t の移動オブジェクトの座標
M^lx , M^ly ：正規ユーザのマウスカーソルの座標
M^ax , M^ay ：幇助ユーザのマウスカーソルの座標
$\Delta t1$：中継PCから幇助ユーザにCAPTCHAのフレーム画像が送信されてくるまでの時間
$\Delta t2$：幇助ユーザから中継PCにマウスカーソルの座標が送信されてくるまでの時間

まず、正規ユーザが提案するCAPTCHAのサーバーにアクセスしているときの振る舞いを以下に示す。

(1) 時刻t0の移動オブジェクトの座標(Ox_{t0} , Oy_{t0})に移動オブジェクトが表示されている。
(2) 中継PCは、(Ox_{t0} , Oy_{t0})に移動オブジェクトが表示されている。
(3) 中継PCは、(Ox_{t0} , Oy_{t0})に移動オブジェクトが表示されてりうフレーム画像を取得し、幇助ユーザに送信する。

(4) Δt1経った時に幇助ユーザには、(Ox_{t0}, Oy_{t0})に移動オブジェクトがあるように見える。
(5) 幇助ユーザは、移動オブジェクト上にマウスカーソルを置く。このとき、マウスカーソルの座標を(M^ax, M^ay)とする。この座標は、②で中継PCに送信される。
(6) マウスカーソルの座標(M^ax, M^ay)は、（５）からΔt2経った時に、中継PCに到着する。この時、中継PC上の移動オブジェクトの位置は座標(Ox_{t1}, Oy_{t1})まで移動している。
(7) マウスカーソルの座標(M^ax, M^ay)と(Ox_{t1}, Oy_{t1})のずれがと比べると、(M^ax, M^ay)と(Ox_{t1}, Oy_{t1})のずれが大きくなる。

以上より、正規ユーザーがアクセスした時のマウスの座標のずれより幇助ユーザーがアクセスした時のずれがΔt1 + Δt2の分だけ大きくなっている。

正規アクセスと比較すると、リレーアタックでは、CAPTCHAのフレーム画像を幇助ユーザーに送信する処理と幇助ユーザーの解答を攻撃者の中継PCに送信する処理が追加されている。この追加された通信によって、CAPTCHAサーバーに直接アクセスしている中継PCで表示されているフレーム画像と幇助ユーザーのPC上で表示されているフレーム画像には、時間のずれが生じる。このずれを生み出している遅延時間を利用してリレーアタックによるCAPTCHAの突破を防ごうというのが提案手法の基本的な考え方である。

3.2.2 物体追跡技術への対処

提案手法のCAPTCHAでは、基本的には、移動する円形オブジェクトをマウスカーソルで追跡する解答方法をとっている。その点に着目すると、物体追跡技術を用いて、移動オブジェクトをプログラムで自動的に追跡する攻撃が考えられる。

そこで提案手法では、移動オブジェクトと同じ形、大きさ、色の妨害オブジェクトを用いて、
プログラムによる追跡が困難になるように工夫を加えた。
提案するCAPTCHAは、移動オブジェクトのフレーム画像と妨害オブジェクトのフレーム画像を重ねて表示する。
また、移動オブジェクトが1フレームごとに位置を更新するのと同時に複数の妨害オブジェクトがランダムに位置を更新する。
移動オブジェクトを追跡するためには、CAPTCHAのフレーム画像を解析して、移動オブジェクトを特定する必要がある。ところが、攻撃者が移動オブジェクトを自動追跡しようとフレーム画像を解析しようとしても、各フレーム画像は、同じ形状、色のオブジェクトがランダムに配置されているようにしか見えない。そのため、移動オブジェクトを検出し追跡することは困難になると考えられる。

3.3 認証手順

提案するCAPTCHAを用いた認証手順を図4に示す。
CAPTCHAのプログラムが動作を始めると、移動オブジェクトが動き出す。その上にマウスカーソルを乗せたら追跡開始とし、解答時間の間、追跡を行う。移動オブジェクトの中心座標とマウスカーソルとの距離が移動オブジェクトの半径以下であるときを追跡が成功していると呼ぶ。
その期間の長さの総和を追跡成功時間とし、それが設定した閾値よりも長ければ、解答者を正規ユーザーと判断する。
追跡してもらう解答時間の長さは、現在最も広く利用されている文字列CAPTCHAの解読にかかる平均所要時間は10秒程度あるため [5]、追跡開始から10秒間とした。

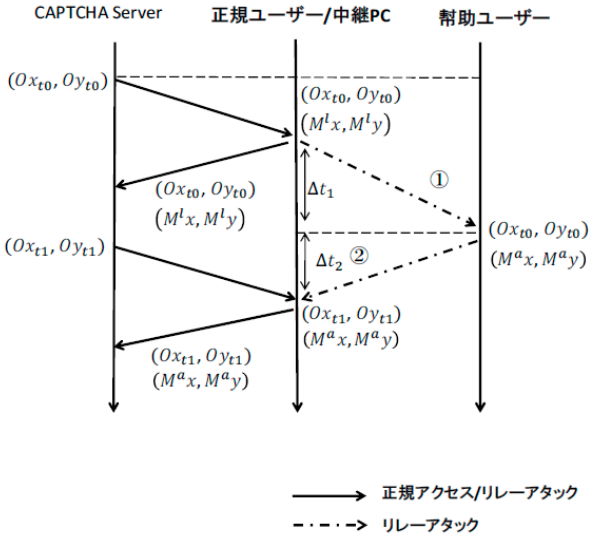

図 3. 提案 CAPTCHA に対するリレーアタックのシーケンス図.

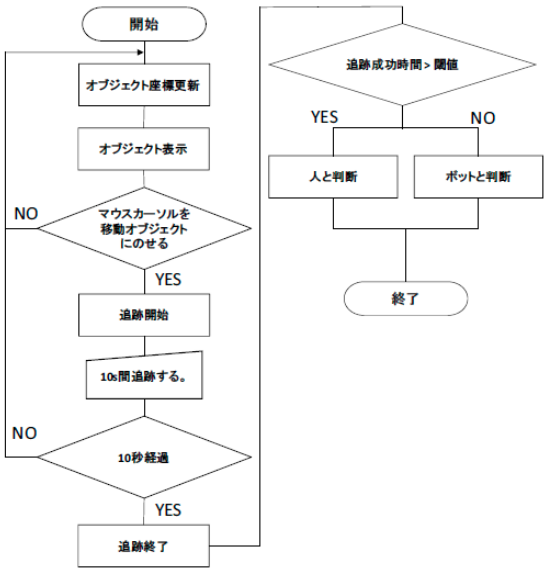

図 4. 認証手順のフローチャート.

4. 実装

4.1 開発環境

本CAPTCHAの開発言語にはJavaScript、CAPTCHAのサーバーはNode.js を用いることにより、CentOS6.6上で実装した。

4.2 CAPTCHA システムの実装

図5 は、作成したCAPTCHAの1 フレームを抜き取ったものである。図中のスタート位置とは、CAPTCHAが開始されたときの移動オブジェクトの位置であり、現在位置とは、抜き取ったフレームでの移動オブジェクトの位置である。白線は、抜き取ったフレームまでに移動オブジェクトが移動した軌跡を表したものである。移動オブジェクトは、白線で示されているような不規則な動きをする。

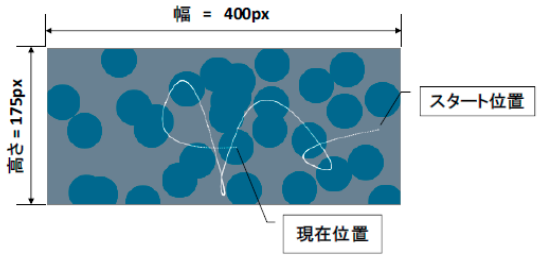

図 5. 実装した CAPTCHA の 1 フレーム.

5. 実験と考察

今回は、提案CAPTCHA がリレーアタックに耐性を持つことの検証とそのユーザビリティ評価を通して、CAPTCHA としての実用性について調査する。

5.1.1 実験目的

実装した提案CAPTCHA に対して実際にリレーアタックを行い、リレーアタックへの耐性が与えられてるかを確認する。

5.1.2 実験方法

実験は、宮崎大学工学部生の被験者8 名に、構築した実験用サイトに、正規アクセス及びリレーアタックでそれぞれ5 回ずつアクセスした上で提案CAPTCHAを解いてもらい、移動オブジェクトの追跡成功時間の計測を行った。

今回は、リレーアタックを再現するためのソフトウェアとしてVNC(Virtual Network Computing) を使用した。VNCはネットワークを通じて接続された他のコンピューターの画面を遠隔操作するソフトウェアである。

正規アクセスでは、被験者にCAPTCHAサーバーに直接アクセスしてもらい、表示された提案CAPTCHAを解いてもらった。リレーアタックでは、中継PC でVNCサーバーを起動しておき、その中継PCでCAPTCHAサーバーにアクセスして提案CAPTCHA を表示させる。次に、幇助ユーザー用のPC のweb ブラウザから中継PC のVNC サーバーに接続する。すると、中継PC のCAPTCHA が表示された画面が幇助ユーザー用のPC に表示されるので、被験者に幇助ユーザー用のPC を操作して中継されたCAPTCHAを解いてもらった。なお、正規アクセスのときに、CAPTCHAサーバーにアクセスするPC とリレーアタックのときにVNC サーバーを起動して、CAPTCHA サーバーにアクセスする中継PC は、同じものを利用した。

5.1.3 実験結果

正規アクセスとリレーアタックのそれぞれの環境で被験者8 人に5 回ずつ提案CAPTCHA を解いてもらったときの40 個のデータから得られた、移動オブジェクトの最長追跡成功時間、最短追跡成功時間、平均追跡成功時間を表1 に示す。

表1 より、リレーアタックの最長追跡成功時間は2.3秒であり、正規アクセスでの最短追跡成功時間である4.1 秒よりも短い。このことから、リレーアタックか否かを判断する追跡成功時間の閾値を4 秒に設定（4秒以上の追跡成功時間であれば、CAPTCHA を正しく解いたとする。）すると、実験環境の場合は、提案CAPTCHA はリレーアタックに耐性を持つといえる。

しかし、リレーアタックの遅延時間は通信環境に依存するため、今後は、適切な閾値を効率良く発見できる手法を検討する必要がある。

表 1. 実験結果

	正規アクセス	リレーアタック
最長追跡成功時間［秒］	8.8	2.3
最短追跡成功時間［秒］	4.1	0.1
平均追跡成功時間［秒］	6.5	0.6

表 2. 成功率と所要時間

成功率[%]	平均所要時間[秒]
96.6%	13.0

5.2 ユーザビリティ評価

ユーザビリティ評価では、提案CAPTCHA の正答率や所要時間の測定と被験者に対するアンケート調査を行い、提案CAPTCHA の実用性を確認する。ユーザビリティ評価は、宮崎大学工学部生10 名を対象に行った。追跡成功時間の閾値は、リレーアタック耐性の評価実験で、正規アクセスした場合の最短追跡成功時間が4.1 秒であったことから追跡成功時間が4 秒以上であれば追跡できているとみなし成功とすることとした。各被験者には練習を行った後、提案CAPTCHAを3 回ずつ解いてもらい、各解答の成否と解答にかかった所要時間を記録した。また、被験者

に1 点～5点の評価でアンケート回答してもらった。アンケートの質問項目を以下に示す。

1. 簡単に解けたか（簡単であれば5 点）
2. 面倒だと感じたか（面倒でないなら5 点）
3. CAPTCHA は、使いやすかったか（使いやすいなら5点）
4. web サービス上で使いたいか(使いたいなら5点)
5. 実際のweb サービスの場面でCAPTCHA を解くことが要求されたときに、文字列CAPTCHAと提案CAPTCHAのいずれかを選ぶことができた場合どちらを選ぶか。

被験者10 人に3 回ずつ解いてもらった提案CAPTCHAの成功率と平均所要時間をまとめた結果を表2 に、アンケート結果について表3 に示す。全ユーザーの平均正答率は、96.6% であり、平均所要時間は、13.0 秒である。一般的な文字列型CAPTCHA の平均所要時間は10 秒程度であるため、提案CAPTCHA は文字列型CAPTCHA と同程度の時間で解けるCAPTCHA であると考えられる。また、アンケート結果から、全体的に1 点、2 点の評価をした回答がないため、提案方式に大きな負担を感じた被験者はいないと考えられる。

質問(5) では、すべての被験者が文字列CAPTCHAよりも提案CAPTCHA を選択していた。これらのことから、提案CAPTCHA は実用的であるといえる。

表3. アンケート結果

	平均得点
簡単に解けたか？	4.6
解くのは面倒だったか？	4.7
CAPTCHAは使いやすかったか？	4.7
Webサービス上で使いたいか？	4.4

6. まとめと今後の課題

本研究では、リレーアタックを行った時に生じる、通信の中継による遅延時間に着目し、リレーアタックに耐性を持たせたCAPTCHA を提案した。また、提案方式のCAPTCHA を実装し、リレーアタックを模擬的に実現する実験環境を構築して、リレーアタックへの耐性の検証実験を行った。実験の結果、提案方式が実験環境の条件の下ではリレーアタックに対して耐性を持つことが可能であることを示した。また、ユーザビリティ調査を行い、提案CAPTCHA が実用的であることを確認した。

今後は、今回確認することができなかった、自動プログラムによる攻撃への耐性について検証実験を行い、提案方式が自動プログラムへ十分な耐性をもつかどうか検討していきたい。

参考文献

1) L. von Ahn, M. Blum, N. Hopper, and J. Langford: *"CAPTCHA: Telling humans and computers apart*, Advances in Cryptology, Enrocrypt'03, vol.2656 of Lect. Notes Comput. Sci.,pp.294-311,2003.
2) 鈴木徳一朗、山本匠、西垣正勝: リレーアタックに耐性をもつ *CAPTCHA* の提案, 情報処理学会研究報告. CSEC,[コンピュータセキュリティ],2010(21),1-8
3) Mohamed, Manar, et al.: *A three-way investgation of a game-CAPTCHA: automated attack, relay attacks and usability.*, Proceedings of the 9th ACM symposium on Information, computer and communications srcurity. ACM, 2014.
4) Mohamed, Manar, et al.: *Dynamic cognitive game captcha usability and detection of streaming-based farming*, the Workshop on Usable Security (USEC), co-located with NDSS. 2014.
5) 可児潤也, 鈴木徳一郎, 上原章敬, 山本匠, 西垣正勝: *4* コマ漫画 *CAPTCHA*, 情報処理学会論文誌, 54(9), 2232-2243..
6) 藤田, 真浩, 池谷, 勇樹, 米山, 可児, 西垣正勝: *SNOW NOISE CAPTCHA:* 無意味な情報を利用した動画 *CAPTCHA* の提案, 研究報告コンピュータセキュリティ(CSEC), 2014(29), 1-7..