Vol.2017-CSEC-78 No.11
Vol.2017-SPT-24 No.11
2017/7/14

リレーアタックに耐性を持つインタラクティブな動画CAPTCHA方式の検討

立田 怜平[1]　山場 久昭[1]　油田 健太郎[1]　朴 美娘[2]　岡崎 直宣[1]

概要：ボットによるアカウントの大量取得や，それに伴う不正行為への対策として CAPTCHA と呼ばれる反転チューリングテストが用いられている．これまで、画像認識や文字認識等の技術の向上によるボットの攻撃手法の巧妙化に対して，CAPTCHA のボットへの攻撃耐性を強化する研究が多く行われてきた．しかし，CAPTCHA を回避する手法として，リレーアタックと呼ばれる攻撃手法が用いられることがある．これは，人間が CAPTCHA の解読を行うため，ボットを想定したこれまでの対策では効果がない．そこで，本研究では，リレーアタックを行った場合に生じる通信遅延に着目し，リレーアタックでの CAPTCHA の解答を困難にすることを目指した CAPTCHA 方式の検討とその有効性を実験により確認した．

キーワード：CAPTCHA，ボット，リレーアタック，通信遅延

A Study on the Interactive Movie CAPTCHA Resistant to Relay Attack

Ryohei Tatsuda[1]　Hisaaki Yamaba[1]　Kentaro Aburada[1]　Mirang Park[2]　Naonobu Okazaki[1]

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

Keywords: CAPTCHA，bot，relay attack，delay time

1. はじめに

Web サービスの普及により，誰でも様々なサービスを利用することが可能となっている．それらの Web サービスに対して，ボットと呼ばれる自動プログラムを用いた不正行為が行われている．例えば，メールサービスのアカウントをボットを用いて自動的に大量取得し，スパムメールの送信に利用するなどの事例が挙げられる．このような，不正行為を防止するために，CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) と呼ばれる反転チューリングテストによる判別手法が広く利用されている [1].

CAPTCHA は，チャレンジ/レスポンス型テストの一種であり，人間には容易に解答できるがコンピューターには困難な問題を出題し，正しい解答をした者を人間と判断するシステムである．一般的に利用されている手法には，Web ページ上に歪みやノイズを加えた文字列画像を提示し，Web サイトの閲覧者がその文字列を判読できるか否かを試す文字列 CAPTCHA（図 1）がある．

しかし，OCR 技術の進歩や，解読アルゴリズムの向上により，文字列 CAPTCHA は容易に突破されるようになってきている．そのため，動物や物などの画像を識別する人

[1] 宮崎大学
University of Miyazaki
[2] 神奈川工科大学
Kanagawa Institute of Technology

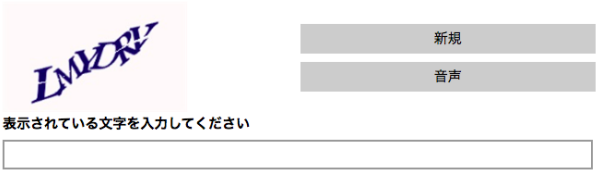

図 1 Microsoft 社のサイトで利用されている CAPTCHA（文字列 CAPTCHA）[6]

図 2 リレーアタックの一例

間の高度な能力を利用する画像 CAPTCHA[2][3] や文字列 CAPTCHA を動画へ応用した動画 CAPTCHA[4][5] など数多くの方式が提案されてきた．

このように，人間には判読しやすく，かつ，ボットには解読が難しい CAPTCHA を実現するために数多くの研究が行われてきたが，CAPTCHA を回避する手法として，リレーアタックと呼ばれる攻撃手法が用いられることがある．リレーアタックは，インターネット上の一般ユーザーや報酬に誘引された人間（以下，幇助ユーザーと呼ぶ．）を利用して CAPTCHA を解読させ，その解答を利用する手法である．リレーアタックでは，人間が CAPTCHA の解読を行うのでコンピューター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，本稿ではリレーアタックを行った際に生じる遅延時間に着目し，リレーアタックでの CAPTCHA の解答を困難にすることを目指した CAPTCHA 方式を提案する．提案方式は，表示される動画に対してマウスカーソルを移動させるアクションを行う動画型 CAPTCHA である．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認識し，マウスカーソルで追跡できるか否かで人間かボットかを判別する．

リレーアタックでは，攻撃者が幇助ユーザーに CAPTCHA の出題画像を転送する通信の遅延時間が発生するため，提案方式 CAPTCHA の場合，攻撃者に提示されている動画と幇助ユーザーに中継されている動画には，ずれが生じ，リレーアタックによる移動オブジェクトの追跡が困難になると考えた．本稿では，リレーアタックを再現し，CAPTCHA の転送で生じる遅延時間で提案方式 CAPTCHA の解答が困難になるかを検証した．

2. リレーアタック

2.1 リレーアタックの仕組み

典型的なリレーアタックは，攻撃者が CAPTCHA を提示する Web ページから CAPTCHA の出題画像を取得し，幇助ユーザーに転送する．そして，報酬を与えることと引き換えに CAPTCHA の解読を行ってもらい，その解答を利用することで CAPTCHA を突破する手法である（図 2）．問題画像の取得や幇助ユーザーへの問題の転送は，攻撃者の作成したプログラムで自動的に行われる．リレーアタックでは，CAPTCHA の解読を行うのが人間であるため，この攻撃手法を実行されると，高い確率で CAPTCHA を突破されてしまう．

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手法 [8]

リレーアタックでは，CAPTCHA を出題するサイトにアクセスする PC の IP アドレスと転送された CAPTCHA を解く PC の IP アドレスが異なっている．この特徴を利用し，リレーアタックが行われていることを検知する．

この手法では，文字列 CAPTCHA を用いている．CAPTCHA を出題するサイトごとに，ランダムな文字列（以下，キーワードと呼ぶ．）を決定し，出題する CAPTCHA の文字列内に，必ずキーワードを含める．この対策では，キーワードが含まれるデータが送信された際に，CAPTCHA を提示しているサイトのサーバーに通知する機能を PC に追加しなければならない．この仕組みを適用すると，リレーアタックで中継された CAPTCHA を解く PC と CAPTCHA を提示するサイトにアクセスしている PC から，CAPTCHA のサーバーに通知が届くことになり，異なる IP アドレスから CAPTCHA の解答が行われたことが分かり，リレーアタックを検知することができる．

しかし，キーワードが送信されたことを通知する機能は，PC にインストールする専用プログラムとして実現されるため，プログラムのインストールを行わないことで，対策の回避が可能となってしまう．

2.2.2 リレーアタックのパフォーマンス低減手法 [9]

リレーアタックで，CAPTCHA を解読する幇助ユーザーには，1000 個の CAPTCHA の解読につき，0.5 ドル〜3 ドルの報酬が与えられるという報告がある [7]．この対策は，1 つの CAPTCHA の解読に掛かる時間を長くすることで，幇助ユーザーが 1 日に解読できる CAPTCHA の数を減らし，金銭的な面からリレーアタックを抑制しようという手法である．

しかし，CAPTCHA の解読に掛かる時間を長くしてしまうと，リレーアタックとは無関係なユーザーの解読の負担が大きくなり，ユーザービリティの低下に繋がってしまう．

2.2.3 DCG-CAPTCHA[10][11][12]

DCG-CAPTCHA は，簡単なミニゲーム形式の

CAPTCHA である（図 3）．ユーザーは，図 3 の青い領域にあるオブジェクトの形状に一致するオブジェクトを白い領域にある複数のオブジェクトの中から選択し、青い領域の同形状のオブジェクトの位置に，ドラッグ&ドロップで移動させる操作を行う．正しいオブジェクトを選択できていれば，ユーザーを人間とみなす．

DCG-CAPTCHA のオブジェクトは常に移動しているので，リレーアタックの攻撃者は，フレーム画像を幇助ユーザーに送信し続けなければならない．この通信で生じる遅延によって，攻撃者の PC に提示されている CAPTCHA と幇助ユーザーに提示される転送された CAPTCHA には，ずれが生じることになる．そのため，幇助ユーザーの解答を利用して CAPTCHA を突破することが困難になる．

しかし，画像処理技術によって同形状のオブジェクトを認識することや移動するオブジェクトを追跡することは，プログラムによって自動化可能であるためボットへの耐性は低いと言える．

図 3 DCG-CAPTCHA[10]

3. 提案手法

既存のリレーアタック対策で示した手法のうち，DCG-CAPTCHA のようなミニゲーム形式の CAPTCHA では，ユーザーはオブジェクトの移動などの変化にリアルタイムで対応しなければならない．リレーアタックのように攻撃者が幇助ユーザーに DCG-CAPTCHA のフレーム画像を転送する通信と幇助ユーザーの DCG-CAPTCHA の解答を攻撃者に送信する通信が生じる攻撃手法では，この 2 つの通信による遅延時間が発生するので，幇助ユーザーの解答を利用しても，攻撃者に提示されている DCG-CAPTCHA にリアルタイムで対応することが難しく，CAPTCHA の解答が困難になるため，リレーアタックに対抗する方式として実用性が示唆されている．

しかし，2.2.3 節で述べたように，DCG-CAPTCHA はボットに対して脆弱であるため，CAPTCHA 本来の機能である，ボットと人間を区別することが難しくなってしまう．そこで，リレーアタックでの解答を困難にしつつ，ボットへの耐性も確保できる方式を考える必要がある．

3.1 提案する CAPTCHA 方式

文献 [11] から、リレーアタックに耐性を持たせるためには，DCG-CAPTCHA のオブジェクトが移動するような動画型で，ユーザーがリアルタイムに動画の変化に対応しながら解答を行う方式が有効であると考えられる．

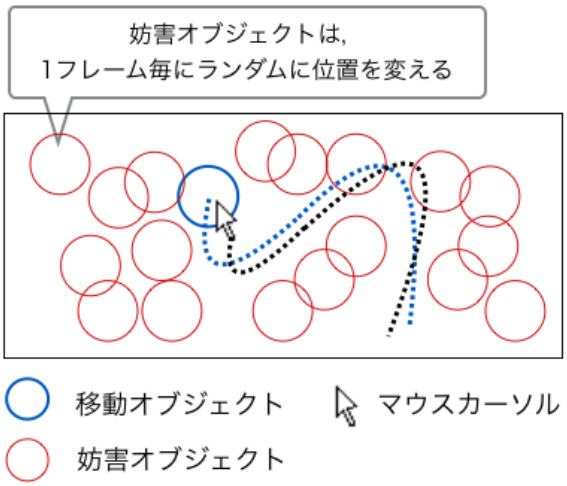

図 4 提案方式 CAPTCHA のコンセプト図

提案方式の CAPTCHA では，移動するオブジェクト（以下，移動オブジェクトとする．）をマウスカーソルで追跡する解答方式とした．これはリレーアタックで生じる通信の遅延時間によって，幇助ユーザーの解答を利用した移動オブジェクトの追跡が困難になると考えたからである．

また，物体追跡技術などを用いて自動的に移動オブジェクトの追跡することを困難にするために，1 フレームごとにランダムに位置を変える妨害オブジェクトを複数追加し，ボットに耐性を持たせた．ユーザーに求められるタスクは，妨害オブジェクトの中から移動オブジェクトを認識し，マウスカーソルで追跡することである．この追跡ができるか否かで解答者が人間かボットかを判断する．以上を踏まえた，提案方式のコンセプト図を図 4 に示す．

次節にて，提案方式のリレーアタック耐性とボットによる攻撃への対処について考察する．

3.2 想定される攻撃に対する耐性

3.2.1 リレーアタック耐性

図 5 に CAPTCHA に対してリレーアタックを行った時の通信についてのシーケンス図を示す．なお，図 5 の提案方式 CAPTCHA の妨害オブジェクトは，省略しているものとする．

図 5 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

Mx_t, My_t
: 幇助ユーザーが転送されたフレーム画像 t に対応した時のマウスカーソル座標

$\Delta t1$
: 中継 PC から幇助ユーザーに CAPTCHA のフレーム

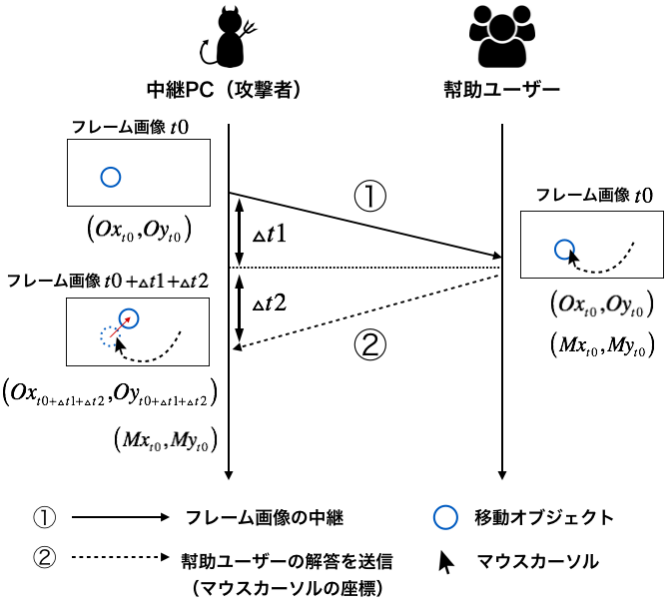

図 5 提案 CAPTCHA に対するリレーアタックのシーケンス図

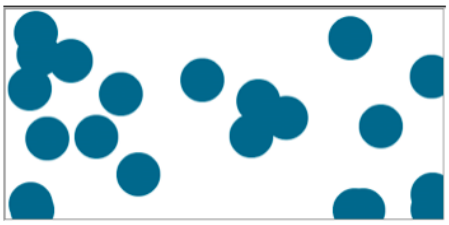

図 6 初期設計の提案方式のフレーム画像

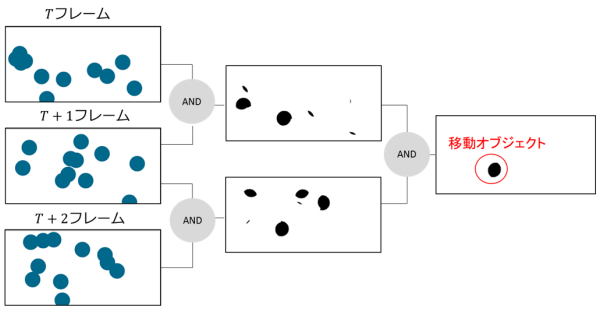

図 7 初期設計の提案方式への差分攻撃

画像が送信されてくるまでの時間
$\Delta t2$
幇助ユーザーから中継 PC に解答に用いるマウスカーソルの座標が送信されてくるまでの時間

次に，提案方式の CAPTCHA にリレーアタックを行った時の振る舞いを以下に示す．

(1) 時刻 $t0$，移動オブジェクトの位置 (Ox_{t0}, Oy_{t0}) のフレーム画像を取得し，幇助ユーザーに送信する．(図 5 の ①)
(2) (1) から $\Delta t1$ 経った時に幇助ユーザーには，(Ox_{t0}, Oy_{t0}) に移動オブジェクトがあるように見える．
(3) 幇助ユーザーは，移動オブジェクト上にマウスカーソルを移動させる．この時のマウスカーソルの座標を (Mx_t0, My_t0) とする．この座標は，中継 PC に送信される．(図 5 の ②)
(4) マウスカーソルの座標 (Mx_t0, My_t0) は，(3) から $\Delta t2$ 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 $(Ox_{t0+\Delta t1+\Delta t2}, Oy_{t0+\Delta t1+\Delta t2})$ まで移動している．

以上より，CAPTCHA のフレーム画像の中継により生じる通信時間と幇助ユーザーの解答（マウスカーソルの座標）の送信で生じる通信時間によって，幇助ユーザーの解答を利用した移動オブジェクトの追跡が困難になると考えられる．このように，リレーアタックで生じる通信の遅延時間を利用し、CAPTCHA の解答を困難にすることが提案方式の基本的な考え方である．

3.2.2 ボットへの対処

提案方式の CAPTCHA は，マウスカーソルで移動オブジェクトを追跡する解答方法をとっているのでプログラムで自動的に CAPTCHA を解こうとする場合，物体追跡技術が用いられると考えられる．移動オブジェクトを自動的に追跡するためには，フレーム画像中からリアルタイムで移動オブジェクトを検出する必要がある．物体追跡において，対象物体の検出には，カラーヒストグラムや形状などの視覚的特徴が用いられる．そこで，妨害オブジェクトは，移動オブジェクトと同色、同形状，同じ大きさに設定する．この難読化により，攻撃者が移動オブジェクトを自動的に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，視覚的特徴が同じ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ため，移動オブジェクトを検出することは，困難になると考えられる．図 6 にフレーム画像を示す．

しかし，移動オブジェクトと妨害オブジェクトの視覚的特徴を同じにするだけでは，ボットへの耐性は十分ではない．動画中から数フレームを取得し，取得したフレームの差分を利用して移動オブジェクトの位置を把握する攻撃が考えられる．移動オブジェクトは，連続的に移動するため連続したフレーム間での位置の変化は小さいが，妨害オブジェクトは，1 フレームごとにランダムな位置に出現するため，連続したフレーム間での位置の変化が激しいという特徴がある．連続したフレームの AND 演算をとると，フレーム間でオブジェクトが重なる領域だけが残るため、位置の変化が激しい妨害オブジェクトを排除していくことができ，位置の変化が小さい移動オブジェクトだけが残り，位置が分かってしまう．図 7 に差分を利用した移動オブジェクトの位置推測の例を示す．

差分攻撃での移動オブジェクトの位置を推定するのを防ぐためには，AND 演算をとると，妨害オブジェクトのように除外されてしまうことが望ましい．そのため，連続したフレーム間で移動オブジェクトが重なる領域を無くす必

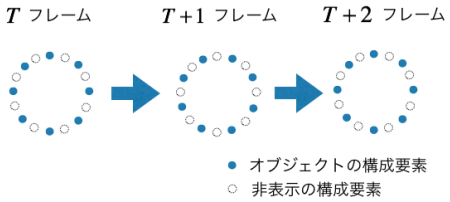

図 8 差分攻撃に耐性を持たせるためのオブジェクトの設計

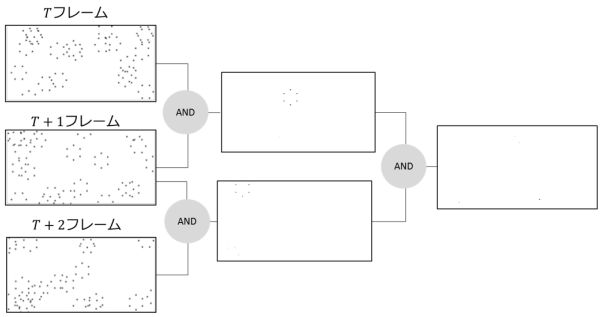

図 9 図 8 のオブジェクト設計に対する差分攻撃

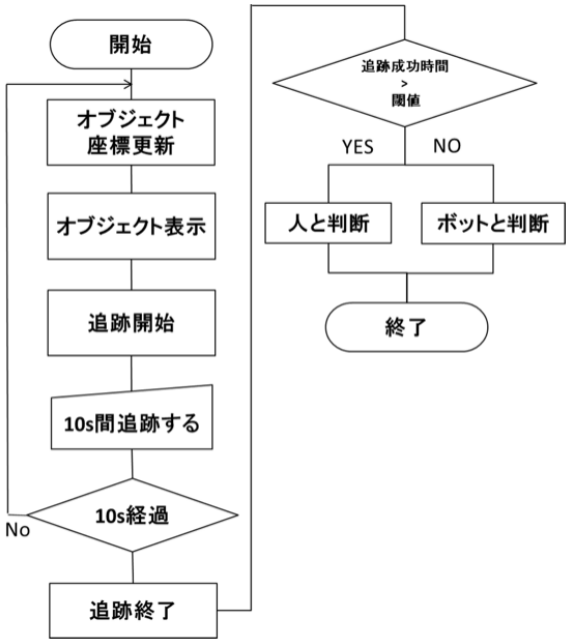

図 10 提案手法のフローチャート

要がある．図 6 のような円形オブジェクトを塗りつぶしたものは，連続したフレーム間で移動オブジェクトの重なる領域が大きいため，差分攻撃に対して脆弱である．

提案方式では，差分攻撃に耐性を持たせるためにオブジェクトを図 8 のように設計した．このオブジェクトは，1 フレームごとにオブジェクトの構成要素がフレーム間で重なる領域がないように位置を変えるので，移動オブジェクトであっても連続したフレーム間でオブジェクトの重なる領域を無くすことができる．そのため，AND 演算で移動オブジェクトの位置を推測することが困難になると考えられる．図 8 のオブジェクトを適用した CAPTCHA に差分攻撃を行うと，図 9 のような結果が得られた．連続したフレーム画像の AND 演算の結果からは，図 7 のように移動オブジェクトの位置を特定できていないことが分かる．

3.3 認証手順

提案方式 CAPTCHA の認証手順を図 10 に示す．CAPTCHA を開始して、移動オブジェクト上にマウスカーソルを乗せたら追跡開始とし，追跡開始から 10 秒の間，移動オブジェクトをマウスカーソルで追跡してもらう．この 10 秒間で，移動オブジェクト上にマウスカーソルを乗せていられた時間（以下，追跡成功時間とする．）が設定した閾値よりも長い時間であれば，ユーザーを人間だと判断する．また，文献 [3] によると，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 秒程度であるため，追跡してもらう解答時間は追跡開始から 10 秒間とした．

4. 実装

4.1 運用方式

提案方式は，動画型の CAPTCHA であり，あらかじめサーバーからクライアントへ出題動画を送信した後，クライアント上で実行する方式では，攻撃者に動画を解析する十分な時間を与えることにつながる．また，リレーアタックにおいても，出題動画そのものを転送された場合，リレーアタックで生じる遅延時間で CAPTCHA を解くことが困難になるという効果は，期待できない．したがって，サーバーとクライアントが動画やユーザーの解答をリアルタイムに送受信可能な環境を構築する必要がある．

4.2 CAPTCHA の実装

本来ならば，4.1 節に示した運用方式を実装することが望ましいが，今回は，提案方式 CAPTCHA の有用性を検討する段階であるため，サーバー・クライアント間の実装は行わず，スタンドアロン PC 上で動作するように実装した．提案方式 CAPTCHA の実装に用いた各パラメーターの詳細を以下に示す．

更新頻度: 60fps で動作する．移動オブジェクトと妨害オブジェクトの位置座標は，1 フレームごとに更新する．

移動オブジェクト: 移動オブジェクトは，12 つの構成要素が円形に配置されて作られる．そのうち，6 つの構成要素を表示し，残りを非表示にする．表示されていた構成要素は，次のフレーム画像では非表示になり，逆に非表示だった構成要素が表示される．このように，構成要素の表示・非表示の切り替えは，1 フレームごとに行われる．

妨害オブジェクト: 妨害オブジェクトは，移動オブジェクトと同じ大きさ，同形状，同色で表示される．

追跡成功時間: 移動オブジェクトは半径 25px の円形なので，マウスカーソルの座標と移動オブジェクトの中心座標との距離が 25px 以下であった時の時間の総和を追跡成功時間と呼ぶ．

ウィンドウサイズ: 高さ 250px，幅 500px

5. 実験と考察

5.1 リレーアタック耐性の検証実験

5.1.1 実験目的

提案方式 CAPTCHA に対して，リレーアタックを行い，リレーアタックで生じる遅延時間によって，提案方式 CAPTCHA の解答が困難になるかを検証する．

5.1.2 実験方法

実験は，宮崎大学工学部生 10 名に，正規アクセスで提示された CAPTCHA とリレーアタックで提示された CAPTCHA をそれぞれ 5 回ずつ解いてもらい，移動オブジェクトの追跡成功時間を計測した．なお，幇助ユーザーと中継 PC 間の CAPTCHA のフレーム画像の転送と幇助ユーザーの解答の送信に掛かる時間は，両者間の地理的距離や通信環境に依存するため，CAPTCHA を中継する中継 PC とリレーアタックで CAPTCHA の解答を行う幇助ユーザー PC の通信に異なる遅延時間を発生させてリレーアタックを行い，それぞれの遅延時間が発生した時の幇助ユーザーの追跡成功時間を計測した．その結果から，幇助ユーザーにとって提案方式 CAPTCHA が難しいものになり得るか検証する．

5.1.3 実験環境

リレーアタックは VirtualBox を利用し，仮想環境上で再現した．中継 PC と幇助ユーザー用の PC をゲスト OS として用意し、2 つのゲスト OS 間で CAPTCHA の中継を行った．また，CAPTCHA の中継には文献 [11] でリレーアタックを再現するためのソフトウェアとして利用されていた VNC（Virtual Network Computing）を用いた．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．

遅延時間の発生には VyOS を利用し，中継 PC と幇助ユーザー間の RTT（Round - Trip Time）が約 50ms，100ms，200ms になるように設定した．よって，計 3 パターンの通信環境でのリレーアタックを行い，追跡成功時間のデータを収集した．

実験環境の詳細は，以下のとおりである．

中継 PC，幇助ユーザー PC（OS）: Ubuntu 16.04 LTS

VNC サーバー: Vino

VNC クライアント: Remmina (色数は「256 色」，品質は「最高」に設定)

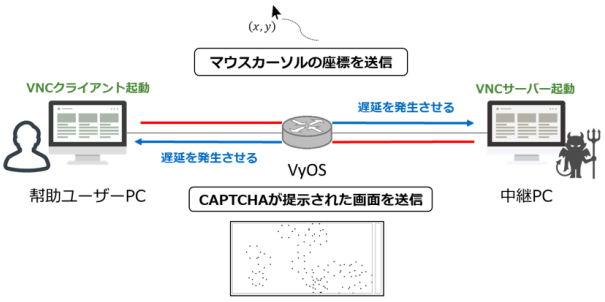

図 11 実験環境

5.1.4 実験結果と考察

実験参加者 10 名に正規アクセスで提案方式 CAPTCHA を解いた時の追跡成功時間と約 50ms，100ms，200ms の遅延時間が発生する条件下でのリレーアタックで提案方式 CAPTCHA を解いた時の追跡成功時間のデータのヒストグラムを図 12，図 13，図 14 に示す．この図から，遅延時間が大きくなるほど，追跡成功時間が短くなっており，移動オブジェクトを十分に追跡できていないことが分かる．

得られた追跡成功時間のデータから，幇助ユーザーにとって達成が難しい追跡成功時間の閾値を検討する．まず，追跡成功時間のデータが正規分布に従うか確認するために，シャピロ・ウィルク検定を行った．シャピロ・ウィルク検定は，正規性検定の一つで，データが正規分布に従うか判断するために用いられる．検定の結果，正規アクセスの追跡成功時間と遅延時間 50ms のリレーアタックの追跡成功時間は，正規分布に従うことが分かった．

今回は，正規アクセスと遅延時間 50ms のリレーアタックの追跡成功時間のデータを正規分布で近似し，幇助ユーザー受容率 (FAR : False Accept Rate) と正規ユーザー拒否率 (FRR:False Reject Rate) から，閾値を検討する．正規アクセスと遅延時間 50ms のリレーアタックの追跡成功時間の正規分布は，図 15 に示す．FAR と FRR については，図 16 のような結果が得られた．この結果によると，FAR と FRR が等しくなる等価エラー率（EER : Equal Error Rate）は約 4%であり，その時の追跡成功時間の閾値は，5.63 秒であった．

ここで，CAPTCHA として実用的な閾値を考える．文献 [14] によれば，一般的に利用されている google の re-CAPTCHA の平均成功率は，97%である．今回の提案方式 CAPTCHA に同程度の正規ユーザーの成功率を持たせたい場合は，FRR が 3%になるように閾値を設定する．このときの追跡成功時間の閾値は約 5.5 秒であり，FAR は約 5%になる．2.2.2 節でも述べたように，幇助ユーザーは，CAPTCHA1000 個の解読につき報酬を得ているが，FAR が 5%であれば，1000 個解いたとしても，このうち成功するのは，50 個程度となるため，経済的に成り立たなくなる．

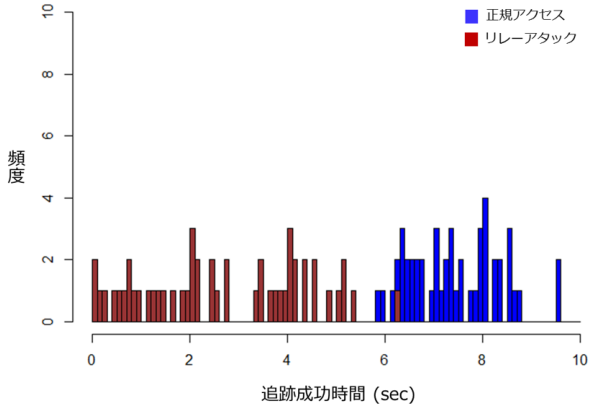

図 12 正規アクセスと遅延時間 50ms のリレーアタックの追跡成功時間

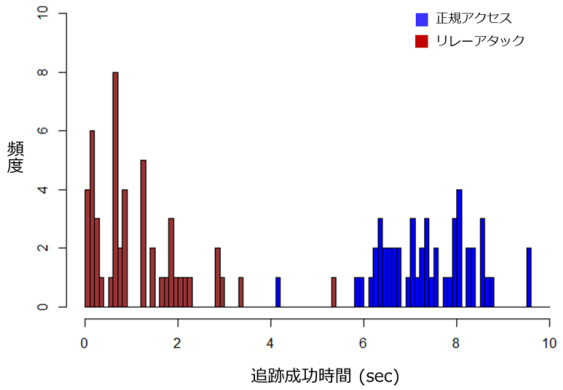

図 13 正規アクセスと遅延時間 100ms のリレーアタックでの追跡成功時間

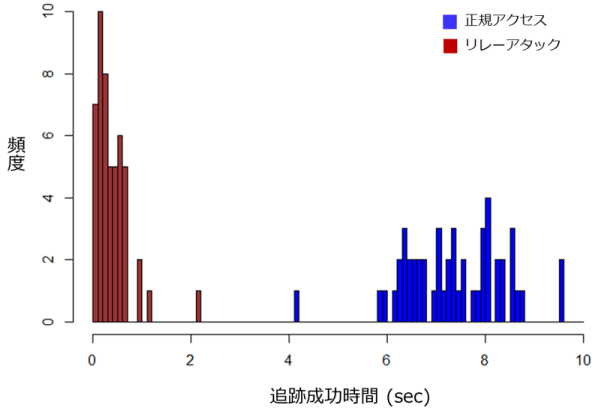

図 14 正規アクセスと遅延時間 200ms のリレーアタックでの追跡成功時間

6. まとめと今後の課題

CAPTCHA を回避する手法として，リレーアタックと呼ばれる攻撃手法が用いられることがあり，ボット想定したこれまでの対策では効果がないことを述べた．これに対して，本稿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，リレーアタックでの解答を困難にする

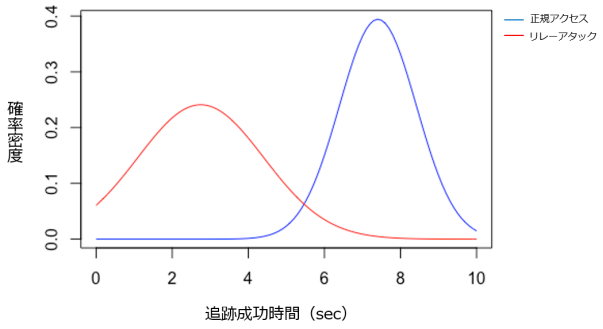

図 15 正規アクセスと遅延時間 50ms のリレーアタックでの追跡成功時間の正規分布

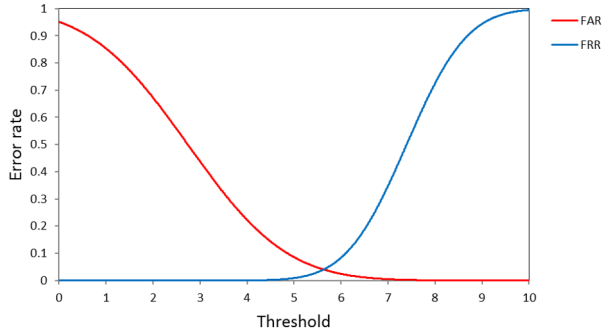

図 16 幇助ユーザー受容率（FAR:False Accept Rate）と 正規ユーザー拒否率（FRR:False Reject Rate）

CAPTCHA 方式を検討した．また，実装した提案方式の CAPTCHA に対して，異なる通信環境でリレーアタックを実際に行い，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，リレーアタックで生じる通信の遅延時間が 50ms のときに，提案方式 CAPTCHA が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，遅延時間が大きくなるほど，提案方式 CAPTCHA を解くことが難しくなることが分かった．

今後は，移動オブジェクトのスピードや大きさなど提案方式の各種パラメーターやリレーアタック実験のパラメーターを変更して，繰り返し評価実験を行い，リレーアタックへの耐性について引き続き調査していく．これと並行して，ボットによる攻撃が差分攻撃以外にあるか検討し，提案方式 CAPTCHA のボットへの耐性についても調査していきたい．

参考文献

[1] L. von Ahn, M. Blum, N. Hopper, and J. Langford, “CAPTCHA: Telling humans and computers apart,” Advances in Cryptology, Eurocrypt’03, vol.2656 of Lect. Notes Comput. Sci.,pp.294-311, 2003.

[2] Elson, J., Douceur, J.R., Howell, J. and Saul, J. Asirra: A CAPTCHA that exploits interest-aligned manual image categorization, Proc. 14th ACM Conference on Computer and Communications Security, pp.366-374 (2007).

[3] 可児潤也, 鈴木徳一郎, 上原章敬, 山本匠, & 西垣正勝. (2013). 4 コマ漫画 CAPTCHA. 情報処理学会論文誌,

54(9), 2232-2243.
[4] NuCAPTCHA: NuCAPTCHA (online), available from (http://www.nucaptcha.com) (accessed 2017-05-11).
[5] 森 拓真, 宇田隆哉, 菊池眞之:アモーダル補完を利用した動画 CAPTCHA の提案, マルチメディア, 分散協調とモバイルシンポジウム 2011 論文集,pp.1518-1525 (2011).
[6] Microsoft: Microsoft アカウント,Microsoft(オンライン), 入手先 (https://signup.live.com)(参照 2017-05-14).
[7] Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, M. G. and Savage, S., "Re:CAPTCHAs-Understanding CAPTCHA-Solving Services in an Economic Context", USENIX Security Symposium, Washington, pp.1-18 (2010).
[8] 鈴木徳一郎, 山本匠, & 西垣正勝. (2010). 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.
[9] 小宮山哲俊, 梅澤猛, & 大澤範高. (2014). L-017 CAPTCHA リレーアタックのパフォーマンス低減手法の提案 (L 分野: ネットワーク・セキュリティ, 一般論文). 情報科学技術フォーラム講演論文集, 13(4), 171-172.
[10] Mohamed, Manar, et al. "A three-way investigation of a game-CAPTCHA: automated attacks, relay attacks and usability." Proceedings of the 9th ACM symposium on Information, computer and communications security. ACM, 2014.
[11] 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.
[12] Gao, Song, et al. "Gaming the game: Defeating a game captcha with efficient and robust hybrid attacks." Multimedia and Expo (ICME), 2014 IEEE International Conference on. IEEE, 2014.
[13] 藤田, 真浩, 池谷, 勇樹, 米山, 可児, ... & 西垣正勝. (2014). 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.
[14] YAN, Jeff; ELAHMAD, Ahmad Salah. Usability of CAPTCHAs or usability isuues in CAPTCHA design. In: Proceedings of the 4th sympousium on Usable privacy and security. ACM, 2008. p. 44-52.