Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

リレーアタックに耐性をもつ CAPTCHA の提案

鈴木 徳一郎† 山本 匠†† 西垣 正勝††

近年，CAPTCHA を攻撃する不正者は，CAPTCHA の解読に自動プログラムを使うのではなく，ネット上の一般ユーザを労力として活用するようになってきている．この攻撃は，リレーアタックと呼ばれ，不正者が正規サイトの CAPTCHA 画像をコピーし，不正者自身が運営するサイトに転載することによって，不正者のサイトを訪問する閲覧者に CAPTCHA を解かせている．リレーアタックにおいては，CAPTCHA を不正に（不正だと知らずに）解読するのは人間であるため，自動プログラムに対するいかなる難読化技術も役に立たない．そこで，本稿は正規サイトへのアクセスを行っている不正者サイトの IP アドレスと CAPTCHA の解答を行っている一般ユーザが操作する PC の IP アドレスの差異を用いたリレーアタック検知方式を提案する．

Proposal of CAPTCHA resistant to relay attack

Tokuichiro SUZUKI† Takumi YAMAMOTO††
Masakatsu NISHIGAKI††

It has been recently reported that malicious users who attack CAPTCHAs are gradually changing their strategy from using automated programs to using human solvers. Such malicious users try to bring net-surfers around the world together by hosting some attractive web site. The malicious web site accesses a victim web site to obtain a CAPTCHA test on the victim site. Then, the malicious site relays the CAPTCHA test to net-surfers who are visiting the malicious web site. The CAPTCHA test will be solved by the net-surfers, and thus the malicious web site can send the CAPTCHA response to the victim site. This is how the malicious web site can use those net-surfers as human resources to solve CPTCHA test on victim web sites. These kinds of attacks are called relay attacks. So far, many researches have studied to improve CAPTCHAs' tolerability against a various attacks conducted by automated programs (malwares). Those countermeasures will, however, not work at all, since the attackers are human beings in the relay attacks. Therefore, we urgently have to tackle the relay attacks. This paper focuses on the difference in PC between the entity who accesses the victim site and the entity who solves the CAPTCHA test under the circumstance relay attacks are conducted. Based on the observation, we propose a relay attack detecting scheme by comparing the IP addresses of the accessing entity with that of the solving entity.

1. はじめに

WEB サービスの発展にともなって，人間と機械を識別するチューリングテストの有用性が益々高まっている．無料 WEB メールやブログなどのインターネットにおける WEB サービス提供サイトに対し，自動プログラム（マルウェア）を使って，大量にアカウントを不正取得する，多数のブログサイトにスパム記事を不正投稿する，大量に不正なサービス利用要求を行うなどのいわゆる DoS（Denial of Service：サービス不能）攻撃が定常的に頻発しているためである．チューリングテストはこのようなマルウェア（悪意のある自動プログラム）と正規のユーザ（人間）を識別するために必須の技術であり，現在，CMU の研究者によって開発された CAPTCHA [1]と呼ばれる方式が広く利用されている．

CAPTCHA の基本形態は，歪曲やノイズが付加された文字列画像を WEB ページに提示し，訪問者がその文字を判読できるか否かを試すものである．この方式の CAPTCHA の例を図 1 に示す．また，画像に限らず，音声などを利用した CAPTCHA も利用されている．

図 1．Google で使用されている CAPTCHA 画像[3]
Figure 1: An example of a CAPTCHA used for Google Accounts

しかし，近年，CAPTCHA を攻撃する不正者は，CAPTCHA の解読に自動プログラムを使うのではなく，ネット上の一般ユーザを労力として活用するようになってきている．不正者は，自身が運営するサイトを用意し，そのサイト上で，正規サイトの CAPTCHA 画像をコピーして，不正者サイトへの閲覧者に対してその CAPTCHA 画像を表示する自動プログラムを実装しておく．不正者サイトにアクセスしてきた一般ユーザは，表示される CAPTCHA を，それが不正な CAPTCHA であると知らずに解読する．不正者サイトは，この CAPTCHA レスポンスを正規サイトに転送してやれば，正規サイトの CAPTCHA を通過することができる．この攻撃は，CAPTCHA 画像および

† 静岡大学大学院情報学研究科 〒432-8011 浜松市中区城北 3-5-1
Graduate School of Informatics, Shizuoka University
†† 静岡大学創造科学技術大学院 〒432-8011 浜松市中区城北 3-5-1
Graduate School of Science and Technology, Shizuoka University

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

そのレスポンスが不正者サイトを経由してやりとりされることから「リレーアタック（relay attack）」と呼ばれている[2].

現在までに，人間には判読しやすく，かつ，自動プログラムには解読が難しいCAPTCHA を実現するために数多くの技術が研究されている [5][11] [16,17]. しかしリレーアタックにおいては，CAPTCHA を不正に（不正だと知らずに）解読するのは人間であるため，これまでに培われてきた自動プログラムをターゲットとした対策技術は一切役に立たない．また，そもそも CAPTCHA は機械と人間を区別するためのチューリングテストであるため，人間による不正な CAPTCHA 解読（リレーアタック）と人間による正規の CAPTCHA 解読（正規アクセス）を切り分けることは本質的に不可能である．すなわちリレーアタックは，CAPTCHA の存在意義さえ揺るがす大きな問題であり，その対策は急務となっている[4,5].

そこで本稿では, 正規サイトへのアクセスを行っている不正サイトの IP アドレスと CAPTCHA の解答を行っている一般ユーザの PC の IP アドレスの差異を用いて，正規サイト側がリレーアタックが行われていることを検知し，正規にサービスを受けようとしているユーザ以外の解答を受け付けない CAPTCHA を提案する．

以下, 2 章でリレーアタックについて紹介し, 3 章で提案方式について述べる. 4 章, 5 章で実装・評価を行い，6 章で考察する．7 章で本稿をまとめる．

2. リレーアタック

2.1 リレーアタックの定義

リレーアタックとは，不正者が正規サイトの CAPTCHA 画像をコピーし，不正者自身が運営するサイトや不正者が作成したアプリケーションソフト（トロイの木馬）等にその CAPTCHA 画像を転載することによって，不正者のサイトを訪問する人間やアプリケーションソフトを使用する人間に CAPTCHA を解かせる攻撃である．また，自動プログラムを使って CAPTCHA を解読するのではなく，ネット上の一般ユーザを労力として活用して CAPTCHA を解読する攻撃をリレーアタックと捉えることも可能であろう．

2.2 リレーアタックの仕組み

一般的なリレーアタックの仕組みを説明する（図２）

Step１．不正者は，リレーアタックを行うサイト（以下，リレーサイト）を開設する．リレーサイトは，何らかの報酬（ポルノ画像が閲覧できる，賃金が貰える）によって一般ユーザを誘引する．

Step２．一般ユーザがリレーサイトを訪問した瞬間に，リレーサイトは自動的に正規サイトにアクセスするように作られている．正規サイトはリレーサイトからのアクセスに対し，CAPTCHA を表示する．

図２．リレーアタックの仕組み
Figure 2: An overview of Relay attack

Step３．リレーサイトは，正規サイトの CAPTCHA 画像をコピー（図２①）して，リレーサイト上にその画像を貼り付けた Web ページを生成する．

Step４．リレーサイトを訪れた一般ユーザに，Step３で生成した Web ページを提示（図２②）し，「問題を解いたら報酬を与える」ことを説明する．ただし，不正者が CAPTCHA を不正に解こうとしていること，一般ユーザの解答が不正行為の幇助になることは説明しない．

Step５．一般ユーザは，報酬を得るために，この Web ページの CAPTCHA を解き，リレーサイトに解答を入力する（図２③）．

Step６．CAPTCHA の解答を得たリレーサイトは，正規サイトにそれを転送する．一般ユーザの解答が正解だった場合は, リレーサイトは, 正規サイトの CAPTCHA を通過することができ，目的に応じた不正を働くことに成功する（図２④）．一般ユーザには報酬を与える．一般ユーザの解答が不正解だった場合は，一般ユーザに再入力を促す．

Step７．Step２～６をリレーサイトへの訪問者である一般ユーザを利用して繰り返す．ここで，リレーサイトの一連の動作はすべてプログラムによって自動的に行われる．

2.3 リレーアタックの種類

リレーアタックは大別して以下のようなものがある．

・ポルノサイト（以下，ポルノアタック）[1,6]
不正者が運営するポルノサイトに正規サイトの CAPTCHA を掲示し，ポルノサイト訪問者にポルノ画像を見せる代わりに，解読したい CAPTCHA に解答してもら

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

う．ポルノサイト訪問者はリレーアタックだと知らずにCAPTCHAを解いている．

・労働者募集サイト[7]
不正者は，低賃金労働者を対象に労働者を募集するサイトを運営する．サイトに応募してきた労働者を非常に安い賃金で雇い，解読したい大量のCAPTCHAを労働者に送り，解答させる．低賃金労働者は，お金欲しさのため，リレーアタックだと知っていてCAPTCHAを解く場合もあると考えら，ブラックマーケット化しているとの懸念もある．

・トロイの木馬
表向きは無害のアプリケーションプログラムを装っている．ユーザがこのプログラムを実行すると，リレーサイトへのアクセスを行い，「プログラムの実行のためにはCAPTCHAを解くことが必要だ」というメッセージを表示する．上記ポルノアタックのように，ポルノ画像閲覧ソフトを装ってユーザにCAPTCHAを解かせるもの[6]だけでなく，CAPTCHAを解かないとPCを強制シャットダウンする等，ユーザを脅迫して強制的にリレーアタックを実行させるものも存在する[8]．

・ボット
文献[15]では，ボットを用いてリレーアタックを行う方法が検討されている．ユーザのPCに感染したボットは，リレーサイトへのアクセスを行い，Webブラウザが送信するリクエスト等を遮断して，正規サイトのCAPTCHAを割り込ませて提示する．ユーザが正しくCAPTCHAを解くことができたら，ボットはリクエストを再送信し，その後のWebアクセスを継続する．このリレーアタックでは，ユーザには閲覧先のWebサイトのCAPTCHAが表示されたように見えるため，ユーザにリレーアタックと気づかれることはなく，ユーザがネットサーフィンをする度にCAPTCHAを解かせることが可能であると述べている．

3. 提案方式

3.1 コンセプト

正規サイトに直接アクセスしてWebサービスを受けようとするユーザと，正規サイトにリレーアタックを行おうとしているリレーサイトの様子を図３に示した．正規サイトに直接アクセスしているユーザも，リレーサイトにアクセスしているユーザも，どちらも一般のユーザであるが，ここでは両者を区別するために，前者を「正規ユーザ」，後者を「幇助ユーザ」と呼び分ける．図３から，正規ユーザがCAPTCHAを解く場合は「正規サイトにアクセスしているPC」と「CAPTCHAを解答しているユーザが操作しているPC」が同じなのに対し，リレーアタックでは「正規サイトにアクセスしているサーバ」と「CAPTCHAを解答している幇助ユーザが操作するPC」が異なっていることが分かる．そこで本稿では，各エンティティのIPアドレスを正規サイト側で検査することによって，リレーアタックを検知する方法を提案する．

正規サイトは，HTTPリクエストのFromアドレスから，今アクセスをしてきたエンティティ（正規ユーザのPC，または，リレーサイト）のIPアドレスを知ることができる．ここで，HTTPはハンドシェイクプロトコルであるため，リレーサイトが自らのIPアドレスを詐称した場合には通信が成立しないことに注意されたい．一方，CAPTCHAの解答を行っているエンティティ（正規ユーザのPC，または，幇助ユーザのPC）のIPアドレスを正規サイトに通知する仕組みについては工夫が必要となる．本稿では，第三者機関と一般ユーザのPCにインストールする専用プログラムによって，これを実現することとした．

以下，「正規サイトにアクセスしているエンティティのIPアドレス」と「CAPTCHAを解答しているユーザが操作しているエンティティのIPアドレス」の同異によって正規サイト側でリレーアタックを検知する方式を具体的に説明し，その実現例を示す．

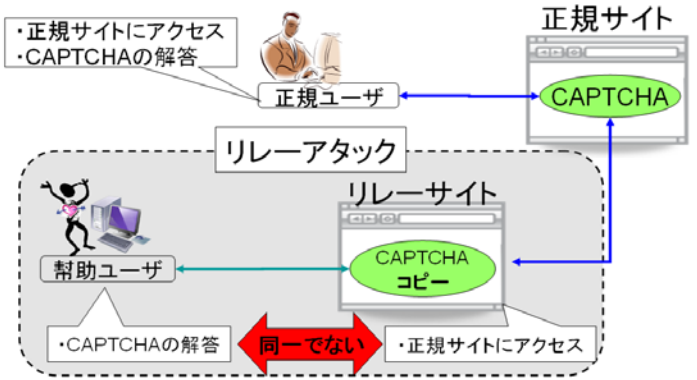

図３．正規ユーザとリレーアタックの違い
Figure 3: A difference between legitimate user and relay attack

3.2 提案方式の実現方法

本方式では，まず正規サイト毎に「CAPTCHAに含めるランダムな文字列（以下，キーワード）」を決定する．キーワードは，全世界でユニークであり，他の正規サイトと重複がないものとする．正規サイトは，自サイトにアクセスしてきたエンティティにCAPTCHAを提示する際に，乱数等を使ってその都度CAPTCHAチャレンジを生成するが，CAPTCHAレスポンス中のどこか一部に登録したキーワードが必ず含まれるようにCAPTCHAチャレンジを生成することが義務付けられる．

正規サイトは，自サイトの名称，URL，キーワードを第三者機関に登録する．第三

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

者機関は，全正規サイトの登録情報（正規サイト毎の名称，URL，キーワード）を一覧にまとめ，「サービスネームテーブル」として管理する．サービスネームテーブルは第三者機関によって常に最新の状態に維持されており，インターネット上に公開される．

また，ユーザの PC 上の Web ブラウザには，以下の機能を追加する．

イ）定期的にサービスネームテーブルを第三者機関から受信し，常に最新版のサービスネームテーブルを所持する．

ロ）ブラウザが外部へデータを送信する時点で，サービスネームテーブル中のいずれかのキーワードが送信データの中に含まれているか検索する．

ハ）送信データの中にキーワードが含まれていた場合は，当該キーワードのサイトの URL にキーワードを検出したことを知らせる「CAPTCHA コンファーム」を送信する．

CAPTCHA コンファームの内容は，ブラウザから送信される当該送信データ（キーワードを含んだ文字列）である．この CAPTCHA コンファームを送信する PC が，3.2 節での実際に CAPTCHA の解答を行っているユーザが操作するエンティティ（正規ユーザの PC，または，幇助ユーザの PC）である．なお，IP アドレスの詐称ができないよう，CAPTCHA コンファームはハンドシェイクプロトコルで送信することが望ましいと考える．

正規サイトは，CAPTCHA レスポンスが正解であることを確認した上で，正規サイトにアクセスしてきているエンティティの IP アドレスと CAPTCHA コンファームを送信してきたエンティティの IP アドレスが一致した場合に，正規ユーザからの Web アクセスであると判定する．CAPTCHA が解けても，CAPTCHA コンファームとの IP アドレスが一致しないエンティティからのアクセスはリレーアタックと判定し，接続を遮断する．

3.3 提案方式の例

本節では，（1）正規ユーザが正規サイトで CAPTCHA を解いた場合と，（2）不正者がリレーアタックを行い，幇助ユーザを利用して正規サイトの CAPTCHA を解いた場合の提案方式の流れを説明する．

3.3.1 正規サイトで CAPTCHA を解く場合（図４）

1）正規ユーザ（IP アドレス : IP_A）が，正規サイト B に①アクセスする（図４①）．

2）当該ページから，正規ユーザのブラウザに Web ページデータが送られる．この Web ページデータには提案方式の CAPTCHA チャレンジが含まれている（図４②）．すなわち，CAPTCHA レスポンスの一部に正規サイト B のキーワードが含まれるように，CAPTCHA チャレンジが生成されている．

3）正規ユーザが CAPTCHA の解答を入力し，「OK」をクリックする．

4）正規ユーザの PC から正規サイト宛に CAPTCHA レスポンスが送信される（図４

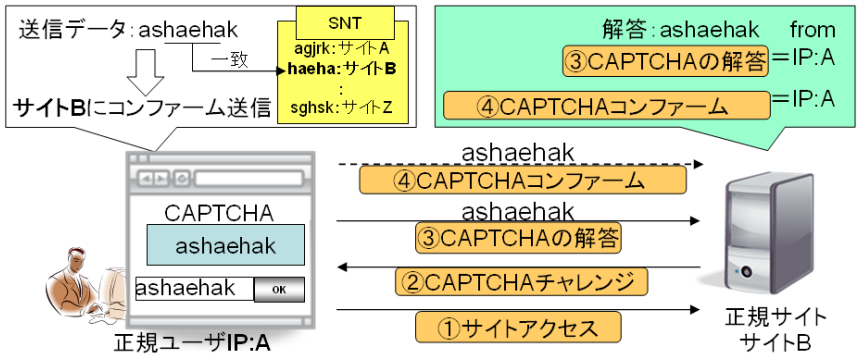

図４．正規ユーザが CAPTCHA を解答する例
Figure 4: An example of Web access from legitimate user

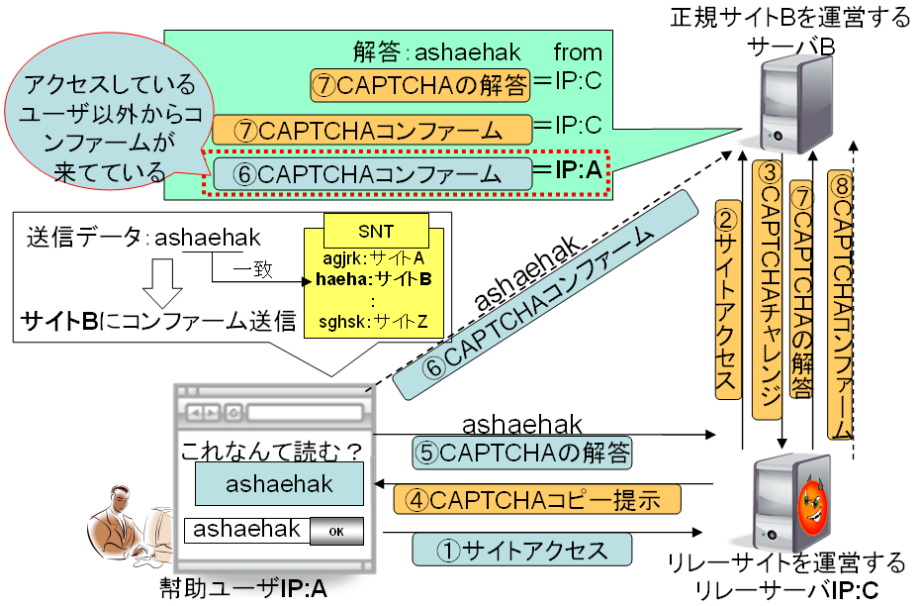

図５．リレーアタックによる解答の例
Figure 5: An example of relay attack detection

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

③)．この CAPTCHA レスポンスにはキーワードが含まれるため，3.2 節イ）～ハ）の機能によって，正規ユーザの PC からサービスネームテーブルに記されている正規サイト B の URL に CAPTCHA コンファームが発信される（図４④)．

５）正規サイト B で，CAPTCHA レスポンス③と CAPTCHA コンファーム④の送信元 IP アドレスの一致を確認する．今回の例では，③と④はどちらも IP_A から発信されているため，リレーアタックでは無いと判定する．

3.3.2 ・リレーアタックの場合（図５）

１）幇助ユーザ（IP アドレス：IP_A）が，リレーサイト（IP アドレス：IP_C）にアクセスする（図５①)．

２）リレーサイトが，リレーアタックのターゲットとなる正規サイト B の Web ページにアクセスする（図５②)．

３）正規サイト B からリレーサーバに Web ページデータが送られる．この Web ページデータには提案方式の CAPTCHA チャレンジが含まれている（図５③)．

４）リレーサーバから幇助ユーザにリレーサイトの Web ページデータが送られる（図５④)．この Web ページデータには３）の CAPTCHA チャレンジのコピーが含まれている．

５）幇助ユーザが CAPTCHA の解答を入力し，「OK」をクリックする．

６）幇助ユーザの PC からリレーサイト宛に CAPTCHA レスポンスが送信される（図５⑤)．この CAPTCHA レスポンスにはキーワードが含まれるため，3.2 節イ）～ハ）の機能によって，幇助ユーザの PC からサービスネームテーブルに記されている正規サイト B の URL に CAPTCHA コンファームが発信される（図５⑥)．ここで，CAPTCHA レスポンス⑤はリレーサイトに，CAPTCHA コンファーム⑥は正規サイト B に送られることに注意されたい．

７）リレーサーバは，「６）にて受信した CAPTCHA レスポンス」を「３）にて受けとった CAPTCHA チャレンジに対する解答」として，正規サイト B に CAPTCHA レスポンスを返送する（図５⑦)．

８）正規サイト B で，CAPTCHA レスポンス⑦と CAPTCHA コンファーム⑥の送信元 IP アドレスの一致を確認する．今回の例では，⑦は IP_C から発信されているのに対し，⑥は IP_A から発信されているため，CAPTCHA のリレーがあったと判定する．なお，リレーサイトは CAPTCHA レスポンス⑦を正規サイトに送る際に，CAPTCHA コンファームについても偽造し，CAPTCHA レスポンスといっしょに正規サーバに送ることも可能である（図５⑧)．しかし，この場合は，正規サーバ B に，CAPTCHA コンファームが 2 通（「IP_A からのコンファーム⑥」と「IP_B からのコンファーム⑧」）届くため，不正を検知できる．

4. 実装

本章では，（A）ユーザ PC に常駐し，Web ブラウザの送信情報を監視し，キーワードが含まれている場合に CAPTCHA コンファームを送信するプログラムと，（B）キーワード込みの CAPTCHA を生成するとともに，CAPTCHA レスポンスと CAPTCHA コンファームの送信元 IP アドレスの同異によってリレーアタックを検知する機能を組み込んだ Web サービスサイトの実装を行い，提案方式の実現性を確認する．

4.1 環境

開発環境は以下のとおりである．

OS：Microsoft Windows XP Professional Version 2002 Service Pack 3
Web ブラウザ：Firefox　version3.5.7[9]
Web サーバソフトウェア：Apache2.2[10]

ユーザ PC の Firefox ブラウザに組み込むプログラム（上記の A）は C 言語で作成し，Web サービスサイトのプログラム（上記の B）は Perl で作成した．

4.2 提案方式の処理の流れ

実装プログラムの処理の流れを図 6 に示す．以下で，ブラウザとサーバの処理をそれぞれ説明する．なお今回は，CAPTCHA チャレンジと CAPTCHA レスポンスの検査についてはセッション DB で管理するシステムで構成した．また，CAPTCHA レスポンスと CAPTCHA コンファームの検査を行うために，コンファーム DB を追加した．

●ユーザ側（ブラウザ機能）

Step U0. サービスネームテーブル（SNT）を定期的に受信する．サービスネームテーブルにアップデートがあった場合には，データを更新する．

Step U1. Web サイトに Web ページを要求する．

Step U2. CAPTCHA の解答が入力フォーム内に入力された状態で，送信ボタンが押されると入力された解答が CAPTCHA レスポンスとして送信される．

Step U3. Firefox ブラウザが送信する任意のデータの中にサービスネームテーブルに記述されているキーワードが含まれてないか常時監視する．キーワード検索には Aho-Corasick 法[18]を用いた．

Step U4. 送信データ中に，サービスネームテーブルに登録されているキーワードが含まれていた場合，サービスネームテーブルを参照し，当該キーワードに紐付いている URL に対し CAPTCHA コンファームを送信する．

●Web サイト側（CAPTCHA 認証）

Step S0. CAPTCHA コンファームを常時受信し，送られてきたコンファームと送信元 IP アドレスをコンファーム DB に保存する．

Step S1. 登録したキーワードを解答に含む CAPTCHA チャレンジを生成し，同時に，セッション ID とその CAPTCHA の正答をセッション DB に保存する．

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

Step S2. Web サイトにアクセスしてきたユーザに Web ページを提示する．
Step S3. ユーザから CAPTCHA レスポンスを取得する．
Step S4. セッション DB から CAPTCHA の正答を取得し，ユーザから送られてきた CAPTCHA レスポンスが正しいか判定する．
Step S5. コンファーム DB から CAPTCHA コンファームを取得し，CAPTCHA レスポンスを送信したエンティティの IP アドレスを検査する．
Step S6. CAPTCHA レスポンスの正否，および，CAPTCHA レスポンスと CAPTCHA コンファームの送信元 IP アドレスの同異から，人間による正規の Web アクセスか否かを判定する．
Step S7. ユーザに判定結果を提示する．
提案方式の導入に際しての追加処理は，Step U0,U2,U4 と Step S0,S4,S5 である．

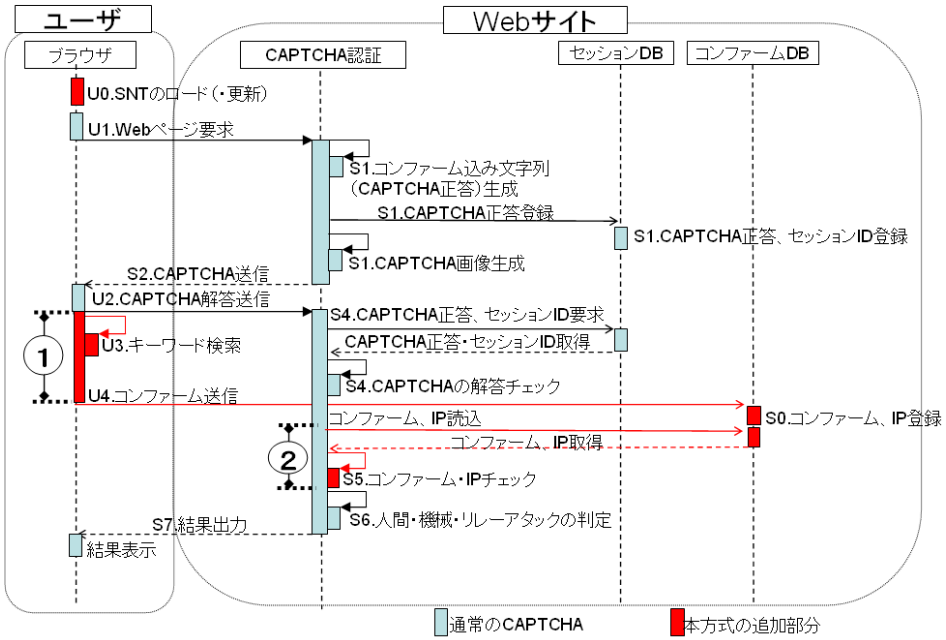

図６．実装プログラムの流れ
Figure 6: System implementation and its flow

5. 検証実験

キーワードをランダムに生成して登録したサービスネームテーブルを用いて，実装した提案方式のリレーアタック検知と CAPTCHA 以外のものを CAPTCHA コンファームとして送信する誤検知率，処理時間について実験を行った．

5.1 実験方法

検知実験は，3.3.2 節で述べたリレーアタックを行うリレーサイトを用意し，提案方式によって実際にリレーアタックが検知可能であるか実験を行った．

誤検知実験では，
・キーワードの文字数： 9 文字
・サービスネームテーブルの登録件数： 1,000,000 件
とし，ユーザが日常のネットサーフィンでブラウザを使用した場合に，提案方式によって無用な CAPTCHA コンファームが送信されることはないか実験を行った．ここで，CAPTCHA コンファームはブラウザからの送信データに対して発せられるので，なるべく多くのデータがブラウザから送信される環境で実験を行うべきである．そこで今回は，ブロガーが自身のブログサイトに記事を書き込むという状況を想定することとした．ブログに掲載されている記事はブロガーがその都度，自身のブラウザから送信したものである．このため，実在するブログサイトから 10 件のブログをランダムに選択し，それぞれ約１ヶ月分のブログデータ（html タグ等を含むページソース）を収集することによって，擬似的に約１ヶ月分のブラウザからの送信データを用意した．そして，その中にサービスネームテーブルに登録されている 1,000,000 件のキーワードが含まれているか計数した．なお今回は大小文字の区別はしていない．

また，提案方式のオーバヘッドを測定した．具体的には，ユーザ PC 内でブラウザから送信されるデータにキーワードが含まれるか否を検索し CAPTCHA コンファームを送信する処理に要する時間（図６①）と，Web サイト内で CAPTCHA コンファームの IP アドレスをチェックする処理に要する時間（図６②）を計測した．ここで，前者については，Firefox を普段から使用している大学生１名に３日間，本方式を実装したブラウザを普段通りに使用してもらったときの図６①の処理に要する時間を計測し，その平均を求めた．実験に使用した PC の CPU は，AMD Phenom™ 9350e （2.0GH z ×４）である．後者は，4 章にて実装した Web サイト上で図６②の処理を 100 回施行し，それに要する時間の平均を求めた．実験に使用したサーバ PC の CPU は，AMD Phenom™ 9350e （2.0GH z ×４）である．

5.2 結果

検知実験から，提案方式がリレーアタックを検知可能であることを確認した．誤検知実験においては，今回は，CAPTCHA コンファームの誤送信は見られなかった．

図６①のオーバヘッドを表１に示す．今回の検索は Aho-Corasick 法[18]により行っ

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

ているため，この処理時間は主にサービスネームテーブルに登録されているキーワードの最長文字数によって決まり，登録サイト数には大きく依存しない．キーワード長が長くなれば処理時間が増加するが，9 文字の場合の一回の検査あたりオーバヘッドが平均 5.32 [μs]であれば，利便性の低下は軽微と考えてよいのではないかと思われる．また，図６②のオーバヘッドは 5 [ms]であった．

表１．誤検知件数とブラウザにおけるオーバヘッド
Table1: Number of false positives & overhead on browser

SNT 登録件数	誤検知件数[件]	平均処理時間[μ s]
1,000,000 件	0	5.32

6. 考察

6.1 各種のリレーアタックに対する効果

2.3 節で述べた各種のリレーアタックに対する提案方式の効果を考察する．

・ポルノサイトを利用したリレーアタック
幇助ユーザはそのサイトがリレーサイトであることに気づかずにリレーアタックを行っていることが一般であるため，ユーザが自身の PC に CAPTCHA コンファーム送信機能をインストールしておけば，Web サーバ側で提案方式を運用することによってリレーアタックを検知できると考えられる．

・労働者募集サイトを利用したリレーアタック
低賃金労働者は，お金儲け目当てで，当該サイトがリレーサーバであると知っていたとしてもサイトにアクセスしてくる可能性がある．この場合，リレーサイト（不正者）は，リレーアタックを検知されないように，幇助ユーザに PC の CAPTCHA コンファーム送信機能をオフにするよう指示をするだろう．これを防ぐためには，本機能を PC の必備要件とし，機能をオフにできないような作り込みが必要だと考える．

・トロイの木馬およびボットを利用したリレーアタック
本稿では，CAPTCHA コンファーム送信機能を Web ブラウザの機能として実装したが，ユーザの PC から発信されるすべての送信データに対してキーワードが含まれるか否かの検査をして，CAPTCHA コンファームを送信するようにすれば，トロイの木馬やボットを利用したリレーアタックも検知可能だと考えられる．

6.2 適用可能な CAPTCHA について

本方式を適用可能な CAPTCHA の条件を以下に記す．

（i）キーワードを含む CAPTCHA レスポンスが Web サーバに送信される．

提案方式では，ブラウザが送信するデータの中に含まれるキーワードを検知して，CAPTCHA コンファームを発信することでリレーアタックを検知する．そのため，CAPTCHA レスポンスの中にキーワードを含み得る CAPTCHA である必要がある．

（ii）CAPTCHA チャレンジからキーワードの位置が推測できない．

CAPTCHA チャレンジから，CAPTCHA レスポンスのどこにキーワードが含まれるかが類推できてしまうような CAPTCHA は不適格である．そのような CAPTCHA の場合，リレーサイトは，正規サイトから取得した CAPTCHA チャレンジから「キーワードの文字列を問うているチャレンジの部分」を機械的に削除し，CAPTCHA チャレンジ残りの部分だけを幇助ユーザに提示することによって，ユーザの PC から CAPTCHA コンファームを発信させずに，機械には解読できない部分の CAPTCHA レスポンスを幇助ユーザから得ることができる．

（iii）CAPTCHA の「解き方」を答えることは不可能である．

例えば，Asirra [16,17] にて表示される各画像に文字列が割り当てられており，猫の画像に割り当てられている文字列を連結したものが CAPTCHA レスポンスの文字列となるような CAPTCHA は不適格である．そのような CAPTCHA の場合，リレーサイトは，正規サイトから取得した CAPTCHA チャレンジの画像に用い，幇助ユーザに「猫の画像の画図番号を答えよ」という CAPTCHA チャレンジを提示することによって，ユーザの PC から CAPTCHA コンファームを発信させずに，CAPTCHA レスポンスの生成法を幇助ユーザから得ることができる．

6.3 プライバシの問題

提案方式は，サービスネームテーブルに登録されているキーワードを含む文字列情報を正規 Web サービスサイトに自動送信しているため，ユーザのプライバシ情報の漏洩に関する問題があると考えられる．

キーワードはランダムな文字列であるため，日常の PC の操作の中で PC から外部に送信されるデータの中で乱数性がある情報と誤一致が起こる可能性がある．そのような情報には，登録ページへのログインの際のパスワード，ネット購入の際のクレジットカード番号，HTTP プロトコルにおけるセッション ID，暗号化された各種データ，等が挙げられる．この内，深刻な問題として懸念されるのは，パスワードやカート番号とキーワードが誤一致してしまい，正規 Web サーバにこれらの情報が CAPTCHA コンファームとして送信されるという事態である．

パスワードとの誤一致を回避する方法としては，キーワード長をパスワード長よりも長くするという対策が考えられる．今回の実装においても，パスワードの平均長が 8 文字である[12,13]ことから，キーワード長を 9 文字としている．しかし，9 文字以上のパスワードを使用しているユーザも少なくはなく，また，クレジットカード番号は 16 桁であるため，根本的な解決策とはならない．更に，キーワード長を長くするほど CAPTCHA レスポンスの文字数も多くなるため，正規ユーザが CAPTCHA の解答を入

Vol.2010-DPS-142 No.16
Vol.2010-CSEC-48 No.16
2010/3/4

力する手間が増え，利便性も低下するという問題もはらむ．

プライバシ漏洩の問題に対する有効な解決策の一つは，CAPTCHA コンファームの Web サーバに発信するにあたって，その内容をハッシュ化するという方法であろう．正しい CAPTCHA コンファームの場合，その内容は CAPTCHA レスポンスのハッシュ値となる．よって，Web サーバ側で CAPTCHA レスポンスと CAPTCHA コンファームの同異を検査する際には，CAPTCHA レスポンスのハッシュ値と CAPTCHA コンファームが一致しているか否かを確認すればよい．

7. まとめと今後の課題

正規サイトにアクセスするユーザ PC と CAPTCHA の解答を行っているユーザ PC の IP アドレスの違いからリレーアタックを検知する方式を提案した．本方式を実装し，実験を通じてその実現可能性を検証した．

提案方式においては，CAPTCHA コンファームの発信が正規ユーザのプライバシの漏洩を招く危険性がある．早急に CAPTCHA コンファームのハッシュ化を図りたい．また，第三者機関によってサービスネームテーブルをどのように管理するかに関して，詳細な検討が必要である．

提案方式は，低賃金労働者を利用したリレーアタックのように幇助ユーザが不正を知った上でリレーサイトにアクセスしてくる場合には，幇助ユーザ自身が PC における本方式の機能をオフすることによって，検知を回避することが可能である．今後は，CAPTCHA の問題形式を工夫し，正規ユーザと幇助ユーザの知識や経験の差等を用いることによって，幇助ユーザには難解な CAPTCHA を実現することができないか検討したい．

謝辞 本研究は一部，（財）セコム科学技術振興財団の研究助成を受けている．

参考文献

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

[2] CNET Japan：今度はポルノ画像をエサに―スパマー対フリーメールサービスの「イタチごっこ」，http://japan.cnet.com/news/sec/story/0,2000056024,20065869,00.htm，2004 年 5 月 10 日

[3] Google，http://www.google.co.jp/

[4] ZDNet Japan：Google の CAPTCHA 実験が的外れな理由，
http://japan.zdnet.com/sp/feature/07zeroday/story/0,3800083088,20392346,00.htm，2009 年 4 月 27 日

[5] 山本匠，J.D.Tyger，西垣正勝：機械翻訳の違和感を用いた CAPTCHA の提案，情報処理学会研究報告 CSEC-46 No.37 2009

[6] Symantec : Torojan.Captchar.A,
http://www.symantec.com/business/security_response/writeup.jsp?docid=2007-103012-0328-99&tabid=2

[7] K Chellapilla, K Larson, P Simard, M Czerwinski : Computers beat humans at single character recognition in reading-based Human Interaction Proofs (HIPs), 2nd Conference on Email and Anti-Spam (CEAS), 2005

[8] Dr.WEB : Harmless virus and other malicious trends of august 2009,
http://news.drweb.com/show/?i=441&c=5&lng=en

[9] Mozilla Japan, http://mozilla.jp/

[10] apache, http://www.apache.org/

[11] 鈴木徳一郎，山本匠，西垣正勝：４コマ漫画 CAPTCHA の提案 2009 年暗号と情報セキュリティシンポジウム予稿集，3D3-3（CD-ROM），2009

[12] Acunetix Web Spplication Security Blog : Statixtics from 10,000 leaked Hotmail passwords,
http://www.acunetix.com/blog/websecuritynews/statistics-from-10000-leaked-hotmail-passwords/

[13] Microsoft Malware Protection Center Threat Research & Response Blog,
http://blogs.technet.com/mmpc/

[14] 安健司，赤羽康彦，尾崎将巳，瀬本浩二，佐々木良一：暗号メールにおける個人情報不正送出チェックシステムの評価，情報処理学会論文誌，Vol.46，No.8，2008

[15] Manuel Egele, Leyla Bilge, Engin Kirda, Christopher Kruegel : CAPTCHA Smuggling: Hijacking Web Browsing Sessions to Create CAPTCHA Farms, 25th Symposium On Applied Computing (SAC), Track on Information Security Research and Applications, Lusanne, Switzerland, March 2010

[16] J.Elson,J.Douceur,J.Howell,J.Saul:Asirra:
a CAPTCHA that exploit interest-aligned manual image categorization. 2007 ACM CSS, pp.366-374, 2007

[17] MSR Asirra Project,http://research.microsoft.com/asirra/

[18]Aho-Corasick algorithm：Wikipedia,
http://ja.wikipedia.org/wiki/%E3%82%A8%E3%82%A4%E3%83%9B-%E3%82%B3%E3%83%A9%E3%82%B7%E3%83%83%E3%82%AF%E6%B3%95