第1章　はじめに

インターネットの急速な普及に伴って，情報のアクセスが容易になり，我々の生活が豊かになった一方で，インターネットを悪用したサイバー犯罪が横行している．

サイバー犯罪は，サーバやアプリケーションの脆弱性を利用して，アクセス権限を要する情報を不正に入手したり，サーバに莫大な負荷をかけて，サービスに支障をきたしたりする行為のことである．サイバー犯罪の被害として，顧客の個人情報が流出したり，サービスが長期間，利用できなくなることなどが挙げられる．これらの被害は，サービスを利用する顧客にとっても，サービスを運営する企業にとっても大きな損失となるため，サイバー犯罪の被害を未然に防ぐことはもちろん，被害に遭った際に犯人を特定するための対策を講じることも重要である．

サイバー犯罪の犯人を特定する情報として，多くの場合，IP アドレスが用いられる．サイバー犯罪の被害を受けたサーバのログから，サイバー犯罪に該当するアクセスの IP アドレスを捜査し，インターネットサービスプロバイダに照会を要請することで，犯人の身元を特定することができる．

しかし，技術の進歩により，近年では，IP アドレスのみによる個人の特定は困難になってきている．VPN や Tor を用いることで，インターネットサービスプロバイダから割り振られている本来の IP アドレスを秘匿化してインターネットに接続することができる．また，キャリアグレード NAT の普及により，通常より IP アドレスでの個人の特定が難しくなっていることも事実である．これらの技術についての詳細は，次章で後述する．

そこで，本論文では，サイバー犯罪が発生した際に，IP アドレス以外の方法で犯人を特定する手法について提案する．インターネット利用者が，サーバに送信する情報は複数挙げられるが，その中でも Cookie に着目した特定システムを提案する．Cookie は，Web サーバが Web ブラウザに対して，任意の文字列を記憶させるための仕組みである．サーバから Cookie を保存するように指示されたブラウザは，以降，その Web サイトに対して，記憶した Cookie を送信するようになる．

Cookie を利用することで，サービスのログイン状態を記憶したり，EC サイトにおいて，ショッピングカートの中身を記憶させたりすることができる．一方で，アクセスしてきたユーザに，ユニークな文字列を発行し，Cookie として保存させることで，過去にそのユーザがアクセスしてきたかどうかがわか

るため，個人を特定するために用いられることもある．このように，Cookie はユーザを特定する十分な情報となり得る．

第2章　研究背景

2.1　IP アドレスによる特定の難しさ

発信元の IP アドレスを特定することで，発信者の利用しているインターネットサービスプロバイダやキャリア，地域レベルの位置情報を取得できることが多い．サイバー犯罪が発生した際に，IP アドレスからインターネットサービスプロバイダを特定し，インターネットサービスプロバイダに照会を要請することで犯人を特定することができる．

しかし，技術の進歩により，IP アドレスによる個人の特定は困難であるケースも珍しくない．IP アドレスによる個人の特定が困難となる技術の代表例を以下に示す．

2.1.1　キャリアグレード NAT

キャリアグレード NAT とは，インターネットサービスプロバイダなどが，ネットワークアドレス変換 (NAT) を行う仕組みである．ネットワークアドレス変換とは，同一 LAN 内の各端末に割り振られたプライベート IP アドレスと，インターネットサービスプロバイダから割り振られたグローバル IP アドレスを変換するための仕組みである．IPv4 アドレスの枯渇問題により，インターネットに接続する各端末すべてにグローバル IP アドレスを割り振ることは困難である．そこで，各端末には，グローバル IP アドレスの代わりに，プライベート IP アドレスを割り振り，インターネットに接続する際に，プライベート IP アドレスをグローバル IP アドレスに変換することで，LAN 内の複数の端末を同じグローバル IP アドレスでインターネットに接続させることができる．ネットワークアドレス変換は，各家庭の LAN 内で行われるが，これをインターネットサービスプロバイダ単位で行ったものがキャリアグレード NAT である．

IPv4 アドレスの枯渇問題を解消する手段として，キャリアグレード NAT がしばしば利用されているが，同一のグローバル IP アドレスが，複数のインターネット利用者によって使用されるため，キャリアグレード NAT を使用している場合は，IP アドレスによる個人の特定が困難となる．インターネットサービスプロバイダでは，キャリアグレード NAT を利用した際の IP アドレスの変換テーブルを保持しているが，変換テーブルは，最新の約 2 ヶ月分

しか記録されていないため，保存期間を過ぎると，IP アドレスによる特定ができなくなる．

2.1.2 VPN

2.1.3 Tor

2.2 IP アドレスによる誤認逮捕

2.3 Cookie による個人の特定

IP アドレスによる個人の特定は困難であったり，誤認逮捕を招く可能性があるため，サイバー犯罪において犯人を特定する情報としては，十分とはいえない．そこで，本論文では，Cookie による個人の特定可能性に着目した．

Cookie とは，Web サーバが，Web ブラウザに対して，任意の情報 (文字列) を記憶させるための仕組みである．サーバから Cookie を保存するように指示されたブラウザは，以降，その Web サイトに対して，記憶した Cookie を送信するようになる．

Cookie を利用することで，サービスのログイン状態を記憶したり，EC サイトにおいて，ショッピングカートの中身を記憶させたりすることができる．一方で，アクセスしてきたユーザに，ユニークな文字列を発行し，Cookie として保存させることで，過去にそのユーザがアクセスしてきたかどうかを調べることができる．このように，Cookie はユーザを特定する十分な情報となり得る．

ブラウザがサーバに送信する情報として，使用している言語，ブラウザや OS の種類，バージョンなどが挙げられるが，これらは他のユーザと似たような情報となることが多い上，本来とは異なる情報を送信することも可能である．それに対して，Cookie は，サーバがブラウザに保存するように指示する値であるため，サーバがユニークな値を発行し，発行した値を記録しておくことができる．そのため，他のユーザと区別することができる上，ユーザが，サーバから指示された値とは異なる Cookie を送信しても，詐称したことがわかる．よって，Cookie は，ブラウザがサーバに送信する情報の中でも，個人を特定できる可能性が高いことがわかる．

2.4 SNS による個人の特定

近年，多くのインターネット利用者が SNS を利用している．Twitter 社が公開した”Q2 2017 Letter to Shareholders”によれば，2017 年における Twitter の月間利用者数は 3 億 2800 万人である．また，Facebook 社が公開した”Facebook Q2 2017 Results”によれば，2017 年における Facebook の月間利用者数は約

20 億人である．SNS が発展した現代においては，世界中の人々が，SNS を通じてインターネットに自分の情報を発信しているといえる．

Twitter や Facebook などの SNS には，多くの個人情報が保存されている．そのため，サイバー犯罪の犯人が利用している SNS のアカウントを特定し，SNS を運営する企業に協力を依頼することで，犯人の身元を特定することは十分に可能であると考えられる．

最近の多くの Web サイト，とりわけニュースサイトやブログサイトでは，自身もしくは自社のサイトを読者に広めてもらうために，SNS のシェアボタンを設置していることが多い．SNS のシェアボタンのリソース (画像やスクリプトなど) を読み込むために，ブラウザは SNS のサイトに対してリクエストを送信するが，その際に，ユーザがその SNS にログインした状態であれば，SNS のサーバにセッション情報を含む Cookie を送信しているはずである．

したがって，SNS のシェアボタンを設置したサイトでサイバー犯罪が発生した際，犯人がその SNS にログインした状態であれば，犯人の SNS のアカウントを特定することができる可能性が高いと考えられる．Cookie が個人を特定する情報となり得ることについては前述したが，SNS の Cookie はサイバー犯罪の犯人を特定する上で非常に重要な情報であるといえる．

2.5　犯人の不注意による逮捕事例

SNS による特定は，犯人が SNS にログインしていることが前提である．用心深い犯人であれば，サイバー犯罪を行う前に SNS からログアウトしているか，別のブラウザを使用しているであろう．

しかし，すべてのサイバー犯罪の犯人が，完全に証拠を残さずに犯罪を行うわけではない．実際に，Tor を利用して児童ポルノを投稿した犯人が逮捕された事例がある．Tor を使用することで発信元を隠蔽することができるが，逮捕された犯人は，Tor の利用中に，不注意により通常のブラウザを使用したために発信元を特定されてしまった．このように，犯人の不注意によって逮捕されるケースもある．

SNS において，利用を中断する度にログアウトするユーザは少数であると考えられる．ログインした状態でも通常のインターネットの使用に支障がないためである．よって，SNS にログインした状態で，その SNS のシェアボタンが設置されたサイトにアクセスすることは十分に考えられる．犯人の不注意により SNS のアカウントを特定することは，犯人の特定に大きく寄与するといえる．

第3章　研究方法

3.1　研究目的

第 2 章で述べたように，IP アドレスによる特定は困難である．そのため，サイバー犯罪の犯人を特定する際は，IP アドレスだけでなく，それ以外の情報を元に特定する手法が必要である．本論文では，Cookie が個人を特定できる可能性が高いことに着目した．また，近年において SNS を利用するユーザが多く存在し，ログアウトする機会が少ないことから，サイバー犯罪において，犯人の SNS のアカウントを発見することが，犯人の特定に大きく寄与するのではないかと考えられる．最近では，SNS のシェアボタンを設置する Web サイトが多く見受けられる．SNS のシェアボタンを読み込む際に送信された Cookie を利用することで，サイバー犯罪の犯人を特定する手法を提案することを目的とする．

3.2　提案手法

SNS のシェアボタンを読み込む際に送信された Cookie から，個人をどの程度特定できるか実験するために，本研究では，匿名掲示板サイトと SNS サイトを用意した．

第4章　実装

4.1　匿名掲示板

本研究で使用した匿名掲示板は，インターネット上にフリーソフトウェアとして公開されている KENT-WEB の LIGHT BOARD(以下，LIGHT BOARD)である．この掲示板は，アカウント登録が不要で利用できる掲示板サイトである．名前の入力が必須となっているが，実名を入力する必要はなく，投稿毎に別の名前を入力することも可能であるため，匿名掲示板として利用することができる．

本研究では，LIGHT BOARD のソースプログラムをダウンロードし，改変したものを，自身のサーバで公開し実験を行った．実験で使用した匿名掲示板サイトを図 4.1 に示す．

図 4.1: 匿名掲示板サイト

掲示板への書込みの際に入力する項目は複数あるが，必須項目は書込みを行った人の名前，書込みのタイトル，本文，画像認証の 4 項目である．画像認証の項目には，入力欄の右側に画像として表示された数字を入力する．こ

れはスパム投稿を防止するためのものである．先述の通り，名前は実名を入力する必要はないため，個人情報を一切入力せずに投稿できるため，利用者から見ると匿名掲示板のように利用することができる．必須項目を入力して投稿ボタンを押下すると投稿内容がサーバに送信され，書込み一覧に投稿内容が表示される．

4.1.1 シェアボタンの設置

LIGHT BOARD にはシェアボタンは設置されていない．本研究では，SNS のシェアボタンを利用して実験を行うため，シェアボタンを設置した．設置した SNS のシェアボタンは，Twitter，Facebook，Mastodon の 3 種類である．

Mastodon のシステムは，匿名掲示板サイトと同様に，本研究の実験用に用意したサーバ上に置かれている．そのため，Mastodon のシェアボタンを読み込む際に送信された Cookie は，保存，解析することができる．送信された Cookie を保存し，その Cookie から Mastodon のユーザを特定するプログラムを作成した．詳細は後述する．

4.1.2 同一ユーザ特定機能の実装

LIGHT BOARD は，書込み時に名前の入力が必須であるが，同一ユーザが複数の書込みを行う際に，それぞれ異なる名前をつけることが可能である．そのため，複数の書込みが同一人物によるものかどうかを判断することは難しい．そこで，本研究の実験で使用する匿名掲示板には，ユーザを追跡するための Cookie を付与する機能を実装した．

初めて匿名掲示板にアクセスしてきたユーザには，サーバ側で生成したランダムな文字列を Cookie として付与する．2 回目以降にアクセスしてきたユーザ (ブラウザ) は，初めてアクセスした際に付与された Cookie をサーバに送信するため，以前にアクセスしたことがあることがサーバ側でわかる．したがって，複数の書込みを行ったユーザは，書込みを行う際に送信している Cookie が同じであるため，同一人物による書込みであることがわかる．

ユーザを追跡するための Cookie を付与する機能に該当する箇所をソースコード 4.1 に示す．

ソースコード 4.1: 同一ユーザ特定機能

```
1  use String::Random;
2
3  sub track {
4      my $host = $ENV{REMOTE_HOST};
5      my $addr = $ENV{REMOTE_ADDR};
6      my $time = time;
7      my ($min,$hour,$mday,$mon,$year,$wday) = (localtime($time))[1..6];
8      my @wk = ('Sun','Mon','Tue','Wed','Thu','Fri','Sat');
9      my $date = sprintf("%04d/%02d/%02d(%s) %02d:%02d",
```

```
10          $year+1900,$mon+1,$mday,$wk[$wday],$hour,$min);
11    my $cookie = $ENV{HTTP_COOKIE};
12
13    my %cookie;
14    foreach (split(/;/, $cookie)) {
15      my ($key, $val) = split(/=/);
16      $key =~ s/\s//g;
17      $cookie{$key} = $val;
18    }
19
20    # もしトラッキング用のクッキーがセットされていればファイルに保存する
21    # 保存されていなければ新しく発行してセットさせる
22    if ($cookie{tracking_id}) {
23      open(DAT, ">> $cf{trackingfile}") or error("open err: $cf{
            trackingfile}");
24      my $new = "$date<>$cookie{tracking_id}<>$host<>$addr";
25      print DAT "$new\n";
26      close(DAT);
27    }
28    else {
29      my $random = String::Random->new();
30      my $tracking_id = $random->randregex("[a-zA-Z0-9]{64}");
31      print "Set-Cookie: $cf{tracking_id}=$tracking_id\n";
32    }
33  }
```

サブルーチン track は，トラッキング (特定) 用の Cookie が送信されなければ新たに Cookie を発行し，送信されていればその Cookie をファイルに保存する．発行する Cookie は，String::Random を使用して生成した乱数を値とする．String::Random を使用して生成する乱数は，大文字，小文字を区別した 64 文字の英数字であるため，他のユーザと重複する可能性は極めて低いと考えられる．よって，匿名掲示板への書込みとトラッキング用の Cookie を照らし合わせ，複数の書込みに対して同じ Cookie が送信されている場合，同一人物による書込みであると判断できる．

4.2 SNS

本研究では，SNS のアカウントを特定する実験を行うためのシステムとして，Mastodon を利用した．本来のサイバー犯罪においては，Twitter や Facebook などの，世界的に利用されている SNS の運営と協力して犯人のアカウントを特定することを想定しているが，本研究の実験では，Twitter や Facebook のサーバ内の情報を閲覧することはできない．そこで，個人で運営することができる，Twitter に似た SNS である Mastodon を利用することで，シェアボタンを読み込む際に送信される Cookie から Mastodon のアカウントを特定する実験を行った．送信される Cookie からアカウントを特定することは，Twitter や Facebook においても技術的に可能であるため，実際のサイバー犯罪では Twitter や Facebook などの運営と協力することで，犯人のアカウントが特定できるといえる．

Mastodon

Mastodon とは，ドイツ人の Eugen Rochko 氏が，2016 年 2 月に開発を開始した，分散型 SNS のことである．Twitter に似た機能を持つ SNS として日本でも注目を集めている．オープンソースソフトウェアとして GitHub 上に公開されており，誰でも自由にプログラムを利用することができる．

Twitter や Facebook とは異なり，Mastodon はプログラムを改変することができ，また改変したプログラムをサーバ上に置いて運営することもできる．GitHub からダウンロードした Mastodon のプログラムに，本研究の実験に必要な機能を実装して利用した．

4.2.1　シェアボタンの実装

匿名掲示板に設置するシェアボタンの実装を行った．/button という URL にアクセスすると，Mastodon のシェアボタンの画像だけが表示されるページを作成した．匿名掲示板では，HTML の iframe タグを利用し，Mastodon のシェアボタンのページを読み込んでいる．

ユーザ (ブラウザ) は，匿名掲示板にアクセスした際に，Mastodon のシェアボタンを読み込むために Mastodon のサーバにアクセスする．Mastodon のサーバにアクセスする際に，もしユーザが Mastodon にログインしている状態であれば，そのログイン状態を保持する Cookie を Mastodon のサーバに送信するため，匿名掲示板にアクセスしたユーザと Mastodon にログインしているユーザを結びつけることができる．

4.2.2　送信された Cookie を保存する機能の実装

匿名掲示板に設置されたシェアボタンを通して Mastodon のサーバに送信された Cookie をデータベースに保存する機能を実装した．Mastodon のシェアボタンが読み込まれた際に，送信元の情報を取得し，データベースに保存している．保存する送信元の情報を表 4.1 に示す．

表 4.1: 保存した送信元のデータ

データ	データ型	データ例
IP アドレス	string	133.19.169.6

Cookie	text	remember_user_token=ABCD1234;
		_session_id=EFGH5678;
		_mastodon_session=IJKL9012

リファラ	string	https://www.google.co.jp/
ユーザエージェント	string	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36

取得したデータは，IP アドレス，Cookie，リファラ，ユーザエージェントの 4 つである．データ型は，データベースに保存する際の型を意味する．データ例は，実際に保存されるデータの例を示している．

データベースには，Cookie の他に送信元の IP アドレスやリファラ，ユーザエージェントを保存している．本研究においては Cookie のみの保存で十分であるが，Cookie が取得できなかった場合や，より詳細な情報を得るために，Cookie 以外の情報も保存することにした．

リファラ

リファラとは，どのページからアクセスしたかを示す情報である．例えば，Google の検索結果からリンクをクリックしてサイト A にアクセスした場合，ブラウザがサイト A に対して送信するリファラは https://www.google.co.jp/ である．リンクをクリックしてページを遷移した場合に限らず，iframe を使用して別のサイトを読み込んだ場合にもリファラは送信される．匿名掲示板の HTML 内に含まれる iframe タグから Mastodon のページを読み込んだ場合，Mastodon のサーバに送信されるリファラは，匿名掲示板の URL である．このように，リファラはどのページを経由してアクセスしてきたかがサーバ側にわかるため，Web サイトの解析ツールなどでしばしば用いられる．

ユーザエージェント

ユーザエージェントとは，ブラウザがサーバに対して送信する情報であり，使用しているブラウザや OS，またその種類などの情報が含まれている．例えば表 4.1 のデータ例の場合，アクセスしてきたユーザは，macOS 10.12.6

を使用しており，Google Chrome 62.0.3202.94 でアクセスしていることがわかる．使用しているブラウザごとに Web ページの表示を変えるなど，ユーザビリティの向上を目的に使用されることもあるが，サイバー犯罪においては，犯人の使用している OS やブラウザがわかるといった利点がある．ただし，ユーザエージェントは詐称 (本来とは異なる情報をサーバ側に送信) することも可能であるため，必ずしもユーザエージェントと同じ OS やブラウザを使用しているとは限らない．

4.2.3 ブラウザフィンガープリンティングの実装

Cookie やリファラ，ユーザエージェントなどはブラウザからサーバに送信される情報であるが，JavaScript を使用することで，通常はサーバに送信されない情報を取得することができる．

JavaScript はブラウザ上で実行されるスクリプト言語であり，Web ページの遷移を行うことなく HTML の構成要素を書き換えたり，非同期でサーバと通信したりすることができる．近年ではほとんどの Web サイトに JavaScript が使用されている．また，JavaScript を使用して，ブラウザやデバイスの情報を取得することができる．このように，Cookie の代わりに JavaScript を利用して，ユーザが使用するデバイスを調べて同一ユーザを特定する仕組みをブラウザフィンガープリンティングという．

ブラウザフィンガープリンティングには，4.2.2 項で説明したリファラやユーザエージェントなども含まれる．本研究の実験では，通常はサーバに送信されない情報である画面サイズを，JavaScript を使用して取得することにした．JavaScript を使用しないと取得できない情報を収集することで，より精度の高い特定手法を提案できると考えられるためである．

サーバ側で，ブラウザから送信された画面サイズの情報を受け取るための API エンドポイントを作成した．API エンドポイントとは，API にアクセスする際の URL のことである．本研究では，/api/v1/fingerprint というエンドポイントを作成し，このエンドポイントに対して HTTP の POST リクエストが送信された場合に，送信された情報をデータベースに保存する機能を実装した．

JavaScript では，ユーザが使用するデバイスの画面サイズを取得し，サーバに送信するプログラムを実装した．該当する JavaScript のプログラムをソースコード 4.2 に示す．

ソースコード 4.2: 画面サイズの取得とサーバへの送信

```
1 var xhr = new XMLHttpRequest();
2 xhr.open('POST', '/api/v1/fingerprint', true);
3 xhr.setRequestHeader('content-type', 'application/x-www-form-
      urlencoded');
4 xhr.send('screen_size=' + screen.width + 'x' + screen.height);
```

JavaScript でサーバにリクエストを送信する際には，XMLHttpRequest() を使用する．新しく生成した XMLHttpRequest() のオブジェクトを変数 xhr に格納する．xhr.open() を使用して，HTTP メソッド (GET や POST など) やリクエストを送信する URL(エンドポイント) を指定する．必要であれば xhr.setRequestHeader() を使用してリクエストヘッダを設定することもできる．最後に xhr.send() を使用して，サーバに送信するデータを指定し，送信する．JavaScript でデバイスの画面サイズを取得するには screen.width(画面横幅) と screen.height(画面縦幅) を使用する．例えば screen.width で取得した値が 1440, screen.height で取得した値が 900 であった場合, 'screen_size=1440x900' という文字列をサーバに送信する．この情報を受け取ったサーバは，データベースにこのデータを文字列として保存する．