Computer Security Symposium 2018
22 - 25 October 2018

Torネットワークへの攻撃に対する サーバ発ランダム画像挿入による防御手法の検討

高橋 元春[1]　成田 匡輝[1]　猪股 俊光[1]　杉野 栄二[1]

概要：近年、ユーザのアクセス先 Web サイトが特定される事を防ぐ匿名通信システム Tor が普及している。しかし、その匿名性に対する攻撃手法も徐々に進化している。攻撃手法の 1 つとして、流れるパケットの特徴から利用者のアクセスする Web サイトを特定する「指紋攻撃」がある。そこで本稿では、Tor に対する指紋攻撃への防御手法を検討する。我々は Tor 秘匿サービスのパケットを分析し、バナー広告のような画像によって、 Tor 秘匿サービスのサーバ側のパケット量をアクセス毎にランダムに変化させる手法を考案した。この Tor サーバ発ランダム画像ノイズで指紋攻撃の効果を低減できる事を示す。

キーワード：匿名化通信、Onion Router、Tor、Tor 秘匿サービス、指紋攻撃

Defence against the Attack to Tor Network by Inserts of Random Graphics from Hidden Server

Motoharu Takahashi[1]　Masaki Narita[1]　Toshimitsu Inomata[1]　Eiji Sugino[1]

Abstract: In recent years, an anonymous communication system "Tor" that prevents attacker from identifying web sites accessed by users is becoming widely used. However, network attack techniques against the anonymity are gradually evolving. As one of the attack techniques, Web Fingerprinting Attack that identifies user's accessing web sites by observation of packets which pass through in the Tor network is becoming a threat. In this paper, we discuss the defense techniques against Web Fingerprinting Attack in Tor Network. We observed packets from Tor Hidden Service and invented the technique that enable Tor Hidden Service to vary packet quantity from Tor HTTP server each time the users access. The defense technique varying the packet's quantity uses graphics like banner ad. We attempt to decrease the effect of Web Fingerprinting Attack by the random graphical noise from Tor server.

Keywords: Anonymity System, Onion Router, Tor, Tor Hidden Service, Web Fingerprinting Attack

1. はじめに

近年、急速なインターネットの普及に伴い、情報をやり取りする上でのプライバシー保護問題が騒がれている。特に、ユーザのアクセス先の Web サイトを特定するプライバシー侵害が発生している。このような状況において、ユーザが身元を隠して Web ブラウジングが可能な匿名化通信技術が開発された。その 1 つに Onion Router（Tor）がある。Tor は P2P 技術を利用した SOCKS プロキシとして動作する。そして Tor は HTTP サーバの IP アドレスを隠す、Tor 秘匿サービスと呼ばれるシステムも提供している。一方、この技術に対しての匿名性を脅かす攻撃手法も登場している。その 1 つに「指紋攻撃」がある。この攻撃は、ユーザのアクセス先 Web サイトである、Tor 秘匿サービスの HTTP サーバから流れるパケットの特徴量（指紋）から、ユーザのアクセス先サイトやサイトのコンテンツを特定する手法である。本研究では、Tor ユーザのアクセス先 Web サイトである、Tor 秘匿サービスの HTTP サーバから画像挿入によるノイズを発生させて指紋攻撃の精度を低

[1] 岩手県立大学 ソフトウェア情報学部
Iwate Prefectural University
Faculty of Software and Information Science

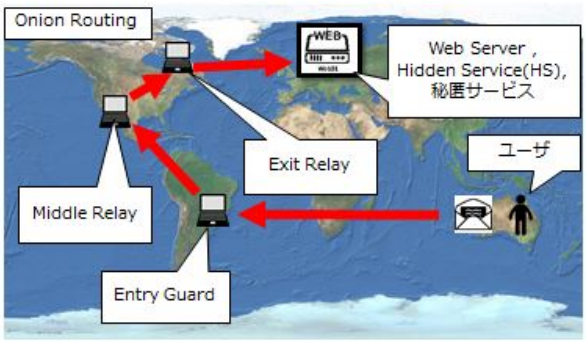

図 1　Onion Routing の Relay の仕組み
Fig. 1　Working of Relay in Onion Routing.

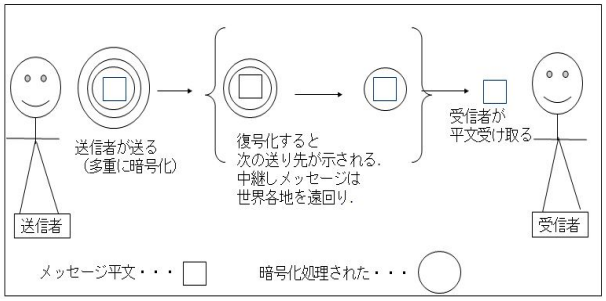

図 2　Tor での暗号化・復号化
Fig. 2　Encryption and decryption in Tor

下させる技術を考案した。そして、提案手法について性能評価を行い、その効果を実証した。

2. Tor と Tor 秘匿サービス

2.1 Onion Routing

Onion Routing は米海軍調査研究所が 1990 年代中頃に開発した暗号化通信技術である。通常、インターネット通信の際にはいくつかの中継ノードを経由してユーザとサーバが通信する。Onion Routing はこの中継ノードに Relay と呼ばれるノードを用いる。まずユーザは各 Relay とあらかじめ鍵を共有しておく。通信の際には、ユーザは各 Relay と共有した鍵で送信者のメッセージを多重に暗号化して、そのメッセージを世界各地の複数の中継 Relay を経由させて通信を行う（図 1)。多重に暗号化されたメッセージは、各中継 Relay が一つずつ復号化していく。ある中継 Relay の一度の復号化につき、その中継 Relay に対して次にメッセージを送るべきの Relay のアドレスが示される仕組みである。各 Relay の分担により復号化されたメッセージは最終的には平文となり、最後の宛先ノードに送信される (図 2)。このようにする事で、ユーザ以外の各 Relay は自分の隣の Relay のアドレスしか知る事ができず、各 Relay を完全に信用できなくとも単体ではユーザとサーバの繋がりを暴く事は困難なシステムである。

2.2 Tor（The Onion Router）

Tor は Onion Routing を実装した匿名化通信システムである。Tor を利用する際、ユーザは図 1 のようにオニオンルータ（以下、OR）と呼ばれる Relay を三つ選択し、それぞれの OR と鍵交換を行う。このときユーザに一番近い OR は Entry Guard と呼ばれる。Entry Guard は、ユーザが接続したい HTTP サーバのアドレスは分からないものの、ユーザの IP アドレスを知っているため、攻撃者の Relay が Entry Guard になると Tor の匿名性に非常に大きな脅威となる。攻撃者の Relay が簡単に Entry Guard になる事を防ぐために、この Relay は Guard フラグを与えられた信頼性のある Relay のみが選ばれる。一方、サーバに一番近い OR が Exit Relay、Entry Guard と Exit Relay との間の OR が Middle Relay と呼ばれる。Tor を実装した Relay の集まりによって匿名化されたネットワークを Tor ネットワークと呼ぶ。また、Tor ネットワークの中に設けられた匿名化された Web 空間をダークウェブと呼ぶ [5]。

2.3 Tor 秘匿サービス

Tor 秘匿サービスとは、ダークウェブ内に設けられた HTTP サーバ、HTTPS サーバの事である。本研究では Tor 秘匿サービスの HTTP サーバーを Tor サーバと呼ぶ事にする。Tor サーバは自身の IP アドレスは隠されている状態となる。よって、ダークウェブ空間に存在する Tor サーバの Web サイトにユーザがアクセスする場合、通常のブラウザからはアクセスできない。Tor サーバの Web サイトにアクセスしたい場合は Tor ブラウザを用いる。Tor ブラウザを用いての Tor サーバへのアクセスには 16 桁英数字の onion アドレスを用いる。

3. 指紋攻撃

指紋攻撃とはパケット観測によりユーザのアクセスする Web サイトを特定する攻撃である。指紋攻撃の手順を以下に示す。まず攻撃者が Tor ユーザの Entry Guard になる。次に Tor ブラウザを所持しているユーザに対し、Entry Guard も Tor ブラウザを準備する。Entry Guard である攻撃者は、自身の Tor ブラウザで、あらゆる Tor サーバの Web サイトにあらかじめアクセスする。その際に攻撃者は自身のパケットを観測し、アクセスした各 Web サイト別にパケット特徴量を作成する。この始めに採取した特徴量を「サンプル指紋情報」と呼ぶ事とする（図 3)。その次に攻撃者である Entry Guard は、攻撃対象であるユーザが Tor ブラウザで Tor サーバの Web サイトにアクセスしたタイミングを見計らって、そのユーザ側の通信のパケット特徴量を観測する。この特徴量を、「攻撃対象の指紋情報」と呼ぶ事とする（図 4)。そして攻撃者は「サンプル指紋情報」と「攻撃対象の指紋情報」を比較し、特徴量の類似度を検証する。実際に類似していればユーザのアクセス先の特定

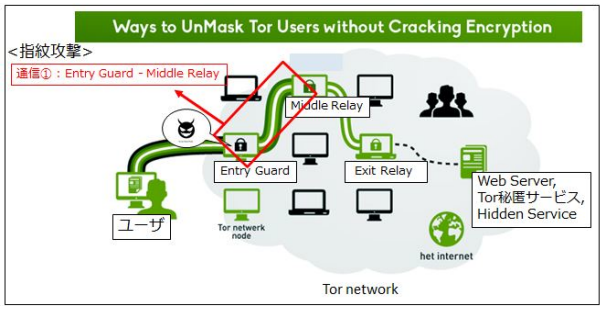

図 3　指紋攻撃（サンプル指紋情報の採取）

Fig. 3　Capturing of sample fingerprint information in Web Fingerprinting Attack

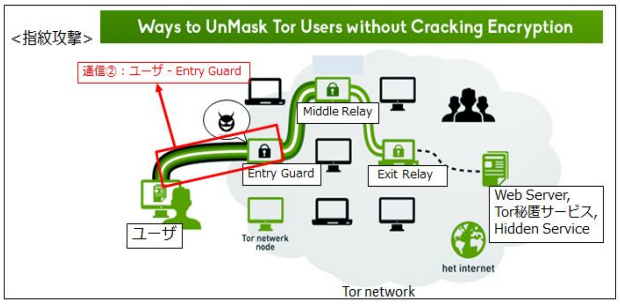

図 4　指紋攻撃（攻撃対象の指紋情報の採取）

Fig. 4　Capturing of target client's fingerprint information in Web Fingerprinting Attack

が成功した事になる。なお、パケット観測はサンプルパケット採取と攻撃対象のパケット採取の 2 段階に分けられるが、ネットワーク上の異なる 2 地点から Tor ブラウザを用いて接続が行われるこの条件において、各地点の Tor ブラウザを起点とした通信同士は、実際に異なる経路の Relay を確立する。

4.　指紋攻撃からの防御

第 3 節で、Entry Guard を占拠する事による指紋攻撃が Tor に対して脅威となる事を述べた。しかし一方で、Tor はこの種のパケット分析に着目した攻撃を回避するための対策をあらかじめ持っている。更に、指紋攻撃に対する防御技術も既に複数考案されている。

4.1　固定パケットサイズ (fixed packet size)

大容量のデータを送信する際、その通信が送信元とアクセス先サイトを紐づける特徴となってしまう場合がある。例えば、ある Web サイト B は 1MB の画像データを扱っており、他のサイトは最大でも 1KB 程度の画像データしか含まないとする。この時攻撃者が、あるユーザ A の Entry Guard になり、そこで 1MB の TCP パケットをキャプチャできれば、そのパケットデータは A が B にアクセスした事を示す有力な手掛かりになるだろう。Tor はこの種の指紋攻撃を回避する手段として、パケットを固定サイズに分割送信する機能をあらかじめ備えている。大量のパケット通信による指紋情報は、攻撃者にとって容易なサイト特定を促す要素と成り得るが、固定パケットサイズ通信の機能を用いる事でそれを回避している。

4.2　過通信 (overcommunication)

Tor の Relay が行う通信は、ユーザの TCP 通信やその中継以外にもある。例えば、Tor は匿名性への脅威への対策として定期的に通信経路を変更するが、その際には使用中の経路を切断するための処理、新たに使用する経路を検索するための処理、そして、新たな経路での通信を確立するための処理が必要である。すなわち、Entry Guard を占拠して Tor のトラフィックを観察すると、その中には「TCP 接続の中継に関する通信」と、「Tor のネットワークの制御に関する通信」が入り混じっている事になる。前者の通信の直後に後者の通信が重なると、両者の見分けが非常に難しく、あたかも後者が前者に含まれるかのように見える事がある [2]。ユーザが Web サイトへアクセスするときも、Web サイトへの接続に関するトラフィックの直後に Web アクセスと無関係なトラフィックが付加される事がある。このような Tor の仕組みは、指紋攻撃を困難にする要素となる。過通信による防御は、見かけの通信の遅延が生じないという利点もある。あくまでロードが完了した後にダミーの通信を付加するものであるため、過通信が生じても通常と同じ時間でロードが完了する。そのため、ユーザ側から見たときに利便性は一切損なわれていない。

4.3　カモフラージュ (camouflage)

カモフラージュは Panchenko らによって提案された指紋攻撃への防御手法である [1]。基本的な考え方はダミーパケットの挿入であるが、人工的なパケットでなく、他のサイトへのアクセスで生じたパケットを加える。具体的には、ユーザ A があるサイト B を訪問する際、バックグラウンドで無作為に選んだサイト C も同時に訪問する。この方法によって、ユーザ A の Entry Guard で攻撃者が指紋攻撃を仕掛けても、観測されるパケットにはサイト B のものとサイト C のものが混在し、指紋情報の正確な解読が困難になる。Panchenko の報告によれば、カモフラージュによって、通信の遅延が 2 倍近くに増加するが、指紋攻撃を大幅に弱められる事が示されている。

4.4　分離読込み (separeted loading)

Tor 通信において、Web コンテンツへの画像ファイル挿入は明確なパケットの特徴を生み出す結果となり、攻撃者に有利に働く一面もある。攻撃者にユーザのアクセス先が画像コンテンツ入りの Web サイトと判明すると、数ある Tor サイトの中からユーザのアクセス先を絞り込まれるため、サイト特定の可能性が高くなる。文献 [3] ではこれに対

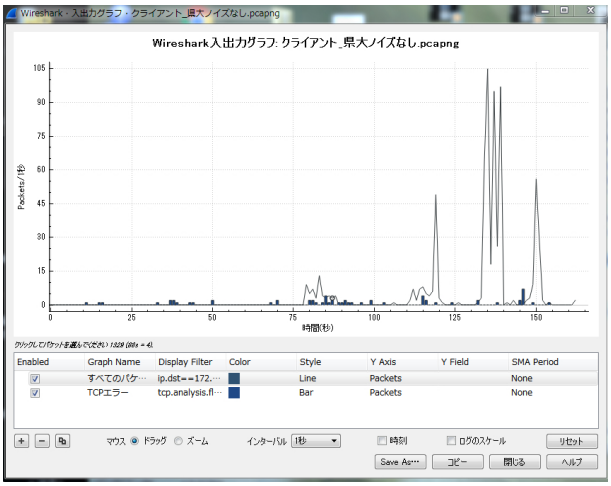

図 5　指紋情報 (パケット量)
Fig. 5　Fingerprint(quantity of packet)

図 6　画像ノイズの挿入
Fig. 6　Inserting of graphics noise

する対策が提案されており、Web サイトの HTML ファイルと画像コンテンツを別の経路で読み込ませる事で、ユーザの画像つき Web サイトの訪問を攻撃者から推測されにくくする形で指紋攻撃を回避する手法を提案している。

5.　提案手法

前節でも述べたように、パケット観測による指紋攻撃が匿名性の脅威となり、ユーザの身元とアクセス先の Tor サイトとの特定につながる。そこで、Tor サイト訪問の通信に Tor サーバ発のダミーパケット (ノイズ) を挿入する事で指紋攻撃を防ぐ技術を提案する。Tor サーバからのノイズ挿入は、Web ページにノイズ生成用画像を表示する仕組みとする。ノイズ画像はバナー広告のようなもので、ユーザが同じ Web ページ訪問においても表示する画像サイズはアクセスする度に異なる（図 6)。これにより、同一サイトへアクセスする際の通信の指紋情報がランダムに変化し、指紋攻撃の精度を低減できる。攻撃者は指紋攻撃において、「サンプル指紋情報」と「攻撃対象の指紋情報」を比較しサイト特定すると思われるが、提案手法はこの両者のパケット特徴量の類似度判定を妨害する効果があるため、十分な攻撃回避効果が望める。

6.　評価実験

5 節で述べた提案手法が指紋攻撃の効果低減にどれほど効果があるかを本節で示す。

6.1　攻撃者モデル

攻撃者は攻撃対象のユーザの Tor 通信の Entry Guard になっているものとする。また攻撃者は、Tor ネットワーク利用者の IP アドレスのリストを所持し、身元を隠したいユーザとそのアクセス先を事前にある程度しぼり込んでいるものとする。指紋攻撃の主な目的は、各 Tor ユーザと各 Tor 秘匿サービス（Tor の HTTP サーバ）の繋がりの関係性を明らかにする事である。実際の攻撃の手順を述べる。攻撃者は Tor Relay を自身の PC にインストールし、自身も Tor ネットワークの一員になる。次に攻撃者は Tor ブラウザを用意して自身が Tor サイトを訪問できる環境をつくる。Entry Guard になった攻撃者は、パケット観測ソフトを起動して、同時に自身の PC から Tor 秘匿サービスのサイトを訪問する。その際 Entry Guard に対しての HTTP レスポンスの通信をパケット観測ソフトで観測し、Tor 秘匿サービスからの「サンプル指紋情報」を採取する。そして Tor ユーザが Tor 秘匿サービスにアクセスするタイミングで、攻撃者は Entry Gurad からパケット観測ソフトを用い、ユーザの HTTP レスポンス通信を観測して、攻撃対象ユーザの指紋情報を採取する。「サンプル指紋情報」と「攻撃対象ユーザの指紋情報」を手に入れた攻撃者は、両方のパケット情報の加工処理を行い、類似判定プログラムを用いて類似度を計算する。最後にこの類似度の結果により、サイト特定を行う。なお指紋攻撃は、インターネットに接続可能な環境さえあれば、ラップトップ PC 上でも低コストで可能である。

6.2　被害者モデル

被害者は Tor ユーザと Tor 秘匿サービスの運営者とする。Tor ユーザは自身の接続の秘匿を目的とし、Tor 秘匿サービスは自身のサーバの IP アドレスを隠す事を目的とする。

6.2.1　Tor ユーザ

Tor ユーザは onion アドレスを用いて Tor ブラウザから Tor 秘匿サービスにアクセスする。

6.2.2　Tor 秘匿サービス

本研究において、Tor 秘匿サービスは Tor ネットワーク上の HTTP サーバである事を前提にする。Tor 秘匿サービスは、あらかじめ Tor Relay をインストールする事によって、Tor ネットワークの一員になる。その後、自身の PC に Web ページを開設する。更にその Web ページを Tor ネッ

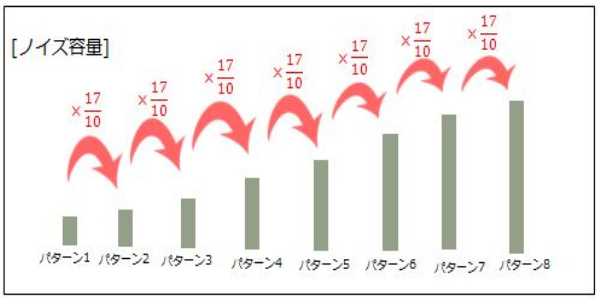

図 7　パターン別ノイズ容量
Fig. 7　Data size of each pattern's noise

トワーク上に配置する手続きを行い、IP アドレスが隠蔽された Tor 秘匿サービスを稼働させる。

6.3　実験環境

6.3.1　データセット

実際に、Tor ネットワーク上の HTTP サーバ（Tor 秘匿サービス）にアクセスし、通信を観測する事でデータセットを収集する。サーバマシンとして、2 つの VPS 上に各 2 個の、計 4 個の秘匿サービスを設置する。4 個の秘匿サービスについては、実験協力者である 2 名分のプロフィールサイトとなっている。協力者 1 名につき、提案手法のノイズありサイトとノイズなしサイトの 2 つを用意する。協力者 1 名分の 2 サイトはノイズ画像以外は全て同じ内容となっている。1 つの秘匿サービスにつき 1 つのサイトとなっている。各サイトは、リンクを張った計 3 つのコンテンツファイルからなるので、3 ページ構成である。実験の目的は、既存手法サイトでの「サンプル指紋情報」と「攻撃対象の指紋情報」の類似度と、提案手法サイトでのそれらの類似度との違いを検証する事である。

Tor 秘匿サービスへの接続とパケット観測について述べる。まず VPS1 のノイズなしサイトに対して、攻撃者とユーザの順番でアクセスする。攻撃者が 3 ページ読み込む様子を「サンプル指紋情報」として観測した後に、ユーザが 3 ページ読み込む様子を「攻撃対象の指紋情報」として観測する。そしてこの手順を 20 回行う。次に VPS1 のノイズありサイトに対して、攻撃者とユーザの順番でアクセスする。同じく攻撃者が 3 ページ読み込む様子を「サンプル指紋情報」として観測した後に、ユーザが 3 ページ読み込む様子を「攻撃対象の指紋情報」として観測する。そしてこの手順を 20 回繰り返す。ノイズ挿入の効果を検証するためのデータセットは上記手順で作成する。より信頼性の高い検証結果を得るために同手順を VPS2 に対しても行う。

6.3.2　OS とソフトウェア

サーバマシンはクラウドサーバの Conoha VPS で Ubuntu 16.04.3 LTS を用いた。Conoha VPS については、メモリが 512MB、ディスクは SSD で 20GB である。Tor のバージョンは 0.2.9.14、Tor ブラウザのバージョンは 7.5.6 である。秘匿サービスの Web ページ開設には、apache、PHP を用いた。なお、apache のバージョンは 2.4.18 である。観測には WireShark2.4.6 を用いる [4]。

次に観測対象の区間について述べる。1 回の指紋攻撃につき、サンプルパケット採取と攻撃対象ユーザのパケット採取の 2 段階の過程が含まれるが、サンプルパケット採取に関する Tor 通信の Relay とユーザパケット採取に関する Tor 通信の Relay は異なる。2 種類の Relay には個々に Entry Guard が存在する。サンプルパケット採取については、その Relay の Entry Guard と攻撃者のブラウザ間の通信を観測し、攻撃対象ユーザのパケット採取については、前者とは別の Relay においてそのユーザのブラウザと攻撃者のノード間の通信を観測する。類似度算出のプログラム作成には Python を使用した。

6.4　ノイズの発生方法

提案手法において、HTTP サーバである Tor 秘匿サービスからノイズを発生させる。ノイズ生成は、Web ページ内にバナー広告のような画像を挿入して行う。ノイズ用画像の表示については、各 Web ページの 1 コンテンツファイルにつきランダムに 8 パターン設ける。各パターンの画像は容量が異なるものとする。よって表示する広告用画像の容量に応じて、通信時の Tor ブラウザへのレスポンスのパケット量が変化する仕組みとなっている。1 つの Web ページは計 3 コンテンツファイルから成るため、512 パターンのノイズ生成が可能になる。各ノイズありサイトの i ページ目にはそれぞれ 8 つのノイズパターンがある。
i ページ目の j 番目のノイズパターンの容量は以下の通り。

$$\begin{aligned} \text{VPS1} &: x_{i_j} \\ \text{VPS2} &: y_{i_j} \qquad (j = 1,2,3,4,5,6,7,8) \end{aligned}$$

j の値が増えるほどノイズ容量（画像サイズ）が大きくなるものとする。

ゆえに、i ページ目のノイズ容量の最小値は以下の通りである。

$$\begin{aligned} \text{VPS1} &: x_{i_1} \\ \text{VPS2} &: y_{i_1} \end{aligned}$$

各ページのノイズパターン別容量の相互関係は以下の通りである。初項がノイズ容量最小値、公比 1.7 の等比数列で表される (図 7)。

$$\begin{cases} x_{i_j} = x_{i_1}\left(\dfrac{17}{10}\right)^{j-1} & (VPS1) \\ \\ y_{i_j} = y_{i_1}\left(\dfrac{17}{10}\right)^{j-1} & (VPS2) \end{cases}$$

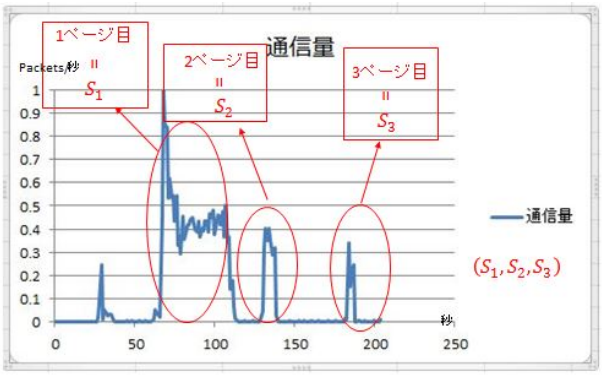

図 8　3 次元ベクトル (サンプル指紋情報)
Fig. 8　Three-dimensional vector (sample fingerprint)

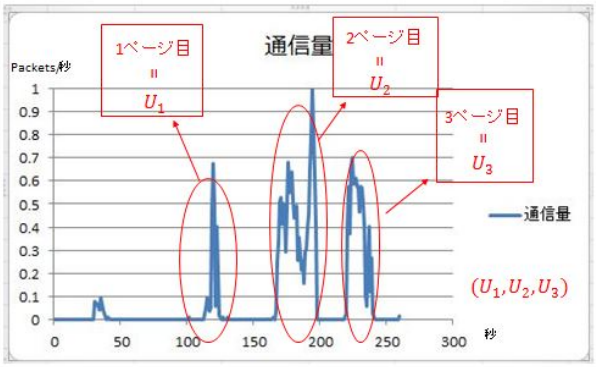

図 9　3 次元ベクトル (ユーザの指紋情報)
Fig. 9　Three-dimensional vector (user's fingerprint)

6.5　指紋情報と類似度

6.5.1　指紋情報

指紋情報の基になるパケット特徴量の項目はいくつか挙げられるが、本研究においては「パケット量」の項目のみ指紋情報として扱う。指紋情報は各サイトのページ別パケット量の合計値を基に求める。

合計値は以下に定める。$(i = 1, 2, 3)$

サンプルアクセスの i ページ目合計　：　S_i
ユーザアクセスの i ページ目合計　：　U_i

これらの各合計値を 3 次元ベクトルに格納したものが指紋情報である。

指紋情報を以下のベクトルの要素に定める (図 8, 図 9)。

サンプル指紋情報　：　$(S_1,\quad S_2,\quad S_3)$
攻撃対象ユーザの指紋情報　：　$(U_1,\quad U_2,\quad U_3)$

6.5.2　類似度

類似度算出には、ユークリッド距離を用いる。「サンプル指紋情報」と「攻撃対象ユーザの指紋情報」の 2 つのベクトルの類似度を算出する。

ユークリッド距離は以下により求める。

$$d = \sqrt{(S_1 - U_1)^2 + (S_2 - U_2)^2 + (S_3 - U_3)^2}$$

類似度は以下の数式で表される。

$$\text{類似度} = \frac{1}{1 + d}$$

類似度は 1 に近いほど類似しており、0 に近いほど類似していない事を示す。

6.6　評価手法

ノイズなしサイト、ノイズありサイトでの 20 回の観測結果の類似度から ROC 曲線を描き、そのカットオフ値を基に類似度における「ボーダライン」を定義する。ROC 曲線は縦軸に True Positive Rate（TPR)、横軸に False Positive Rate（FPR）をとる。TPR は実際に同一のサイトを正しく同一の判定した確率で、FPR は異なるサイトを誤って同一のサイトと判定した確率である。また、異なるサイトを正しく異なるサイトと判定した確率は True Negative Rate(TNR) と呼ぶ。$FPR = 1 - TNR$ で表される。ボーダラインは $TPR + TNR$ が最大になる類似度である。よって各サイトにおいてボーダラインより実際の類似度が高い場合はサイトが特定される確率が高くなる。そしてボーダラインの値が 1 に近いほど、そのサイトにおける脆弱性が高い事を意味する。類似度のボーダラインに関して、ノイズありサイトとノイズなしサイトで比較を行い提案手法のノイズ挿入の有効性を検証する。

6.7　結果

「サンプル指紋情報」、「攻撃対象ユーザの指紋情報」においては VPS1、VPS2 共にノイズなしサイトからのレスポンス通信のパケット量の特徴量は概ね類似していた。

$$VPS1 \text{ 平均類似度} \begin{cases} 0.597 & (\text{ノイズなし}) \\ 0.114 & (\text{ノイズあり}) \end{cases}$$

$$VPS2 \text{ 平均類似度} \begin{cases} 0.605 & (\text{ノイズなし}) \\ 0.112 & (\text{ノイズあり}) \end{cases}$$

攻撃者が指紋攻撃を行い、Tor 秘匿サービスに対してサイト特定を試みる場合、ノイズなしサイト、ノイズありサイト各々に対して「類似度のボーダライン」が存在すると考えられる。ボーダラインより実際の観測の類似度が高い場合のみに指紋攻撃が成功する可能性がある。ボーダラインの値が高いサイトほど脆弱性が高いと言える。実際にノイ

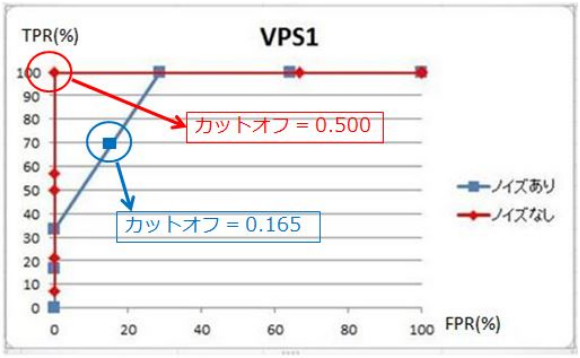

図 10　類似度の ROC 曲線（VPS1）
Fig. 10　ROC curve of similarity ratio (VPS1)

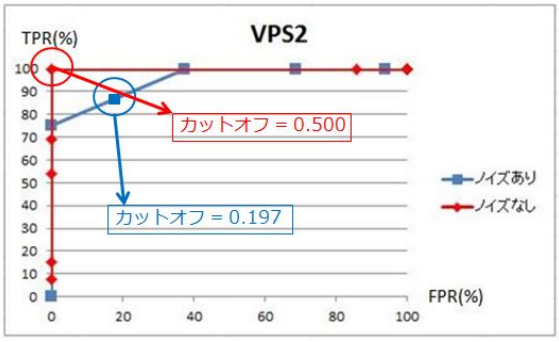

図 11　類似度の ROC 曲線（VPS2）
Fig. 11　ROC curve of similarity ratio (VPS2)

ズなしサイト、ノイズありサイトの類似度に関する ROC 曲線を描き、カットオフ値からボーダラインを算出した (図 10, 図 11)。類似度のボーダラインを、ノイズなしサイトでは d_{old}、ノイズありサイトでは d_{new} と定める。

ボーダーライン算出結果は以下の通り。

$$\text{ノイズなし} \left\{ d_{old} = 0.500 \quad (VPS1, \quad VPS2) \right.$$

$$\text{ノイズあり} \begin{cases} d_{new} = 0.165 & (VPS1) \\ d_{new} = 0.197 & (VPS2) \end{cases}$$

ボーダラインに関しては、ノイズありサイトはノイズなしサイトより値が低くなった。またノイズありサイトのパケット類似度について、ノイズなしサイトのボーダライン 0.5 を上回った確率は VPS1、VPS2 共に 0%であった。これらの事から指紋攻撃の脅威に対する防御として本提案手法が有効である事が分かる。

6.8　課題

本実験により、ノイズ挿入手法が指紋攻撃からの防御に有効である結果が示された。しかし本実験においては、デフォルト容量とノイズ容量の最小値の比率が各サイトのページ間で一定ではなかった。比率を一定にしてノイズ容量の最小値をデフォルト容量より高倍率にして、ノイズパターンを増やすとより防御精度が上がる事が予想できる。しかしノイズ容量を大きくするほどネットワーク負荷が上昇し、処理速度の低下から快適な Web ブラウジングが損なわれる。このようなトレードオフを解決するような匿名化通信技術を考案する事がシステムの課題の 1 つである。これを解決する手段の 1 つとして、デフォルトコンテンツと最小容量ノイズ画像との間の互いの容量と、各パターンのノイズ容量間の相互関係を適切に設定する事が考えられる。ページ別最小容量のノイズパターンを基準に、他のパターンの容量は最小ノイズ容量に対して最適の比率で掛け合わせていく方法を検討中である。また 2 つ目の課題として、ノイズ挿入防御に対するトラフィック逆加工を用いた攻撃を想定した匿名化通信技術を考案する事が挙げられる。攻撃者がレスポンスパケットにおけるノイズパケット挿入を見破り、ノイズ抜きのデフォルトコンテンツのパケット量を算出する技術を用いた場合、指紋攻撃の脅威は大きくなる。これには画像挿入以外のノイズ生成技術も視野に入れて検討したい。実験での課題点としては、攻撃時に観測する指紋情報の項目をさらに増やす事でより精度の高い性能評価を目指す事が挙げられる。具体的な指紋情報の項目としては「パケット量」以外にも「トラフィック総量 (byte)」、「トラフィック平均 (byte)」、「トラフィック分散」、「チャンク平均 (byte)」、「チャンク分散」などを検討する。別の課題点としては、Tor ネットワーク内で攻撃者とユーザのアクセス先不一致時のパケットデータ観測と、これに関係する類似度も性能評価の参考にする事が挙げられる。これにより、攻撃者とユーザのアクセス先一致時の類似度の基準と、そうでない時の類似度の基準を的確に定める事ができ、より適切な性能評価に繋がると考える。以上の点を踏まえて Tor による匿名化通信技術の改良に努めたい。

7.　おわりに

本研究において、Tor 秘匿サービスの匿名性の脅威の 1 つである指紋攻撃に対する防御手法として「サーバー発ランダム画像挿入」を提案し、Tor 秘匿サービスからユーザへ流れるレスポンスパケットにノイズを挿入して、攻撃によるサイト特定の精度に及ぶ影響について調査した。実際、ノイズのパターンと容量が多いほど防御の精度が高い事、ノイズ容量を大きくし過ぎると処理速度が低下する事が明らかとなった。今後の研究では、このトレードオフを解消する事や、画像挿入以外のノイズ生成方法について検討していく予定である。指紋攻撃の実験についてはパケット観測における指紋情報の項目として、「トラフィック総量 (byte)」、「トラフィック平均 (byte)」、「トラフィック分散」、「チャンク平均 (byte)」、「チャンク分散」なども新たに設定する必要があり、今後の課題としたい。

参考文献

[1] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel.“Website fingerprinting in onion routing based anonymization networks.” In Proceedings of the 10th annual ACM workshop on Privacy in the electronic society，WPES ’ 11，pp. 103-114, New York，NY, USA, 2011. ACM.

[2] 横手健一, 松浦幹太, 匿名通信システム Tor の安全性を低下させるトラフィック逆加工, Computer Security Symposium 2012, Vol.3, pp.624-631, (2012).

[3] 横山 絵美里, 宗 裕文, 山場 久昭, 久保田 真一郎, 朴 美娘, 岡崎 直宣, 匿名通信システム Tor に対する指紋攻撃とその対策に関する検討, 「マルチメディア，分散，協調とモバイル (DICOMO2014) シンポジウム」 (2014, 07)

[4] Wireshark：Wireshark ，（online），available from ，（https://www.wireshark.org/），（2018.05.1）

[5] Tor Project：Anonymity online,（online）, available from, (https://www.torproject.org/），（2017.12.20）