

# ダミーパケット挿入が Tor 秘匿サービスの匿名性に与える影響について

竹之内 玲<sup>1</sup> 松浦 幹太<sup>1</sup>

**概要:** 近年, インターネットを通じて個人情報を集めることが用意になっており, プライバシーエンハンスング技術の重要性が高まっている. 匿名通信システム Tor はユーザとサーバーの繋がりを秘匿することで, 通信に匿名性を持たせる実システムである. Tor はサーバーの 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. この Tor 秘匿サービスのプロトコルには 4 つの Tor ネットワーク上のリンクが含まれるが, Tor クライアントと Entry Guard または, Tor 秘匿サービスと Entry Guard の間の通信パケットを観測することでそれぞれを区別出来るということが報告されている. 本論文では, Tor 秘匿サービスの通信に加えるノイズがリンク分類推定精度にどう影響するか調べた結果について報告する.

**キーワード:** 匿名通信, Tor, 秘匿サービス

## The Effect of Dummy Packet on the Detection of Tor Hidden Service

AKIRA TAKENOUCHI<sup>1</sup> KANTA MATSUURA<sup>1</sup>

**Abstract:** Recently, it is not difficult to collect personal data on the Internet. So it becomes important to study on privacy-enhancing technology. Tor is a practical implementation of anonymous communication. Tor provides hidden service, which enables sensitive servers to hide. In the Tor hidden service protocol, there are four links on the Tor Network and they are said to be distinguishable by observation of packets between the Entry Guard and the end node. We examine how noise appended to the communications of these links can affect the accuracy of the link classifier.

**Keywords:** anonymus communication, Tor, hidden service

### 1. はじめに

今日, インターネットにおいて個人の情報を集め, その情報に基づき提供されるサービスが増えている. このような情報はサービス側が個人に合わせたサービスを提供するのに役に立つが, 一方でメールアドレスや住所などといった情報は悪用されるおそれがあり, 敏感な情報を隠したい人が確かに隠すことが出来るプライバシーエンハンスングの需要が高まってきた. 通信の情報も敏感な情報であり, プライバシーエンハンスングを守る技術の一つに暗号化通信がある. 暗号化通信では通信の際にその内容を暗号化

し, 傍受されても攻撃者はその内容を復号出来ずやり取りの内容が漏れないようになるという技術である. しかし暗号化通信においては, ユーザーがどのウェブサイトを閲覧しているのかという情報を秘匿することについては考慮されていない. この接続先の情報については, 匿名化通信を用いることでこの情報を秘匿することが出来る. 実用的な匿名化通信システムの一つに Tor が存在する [9].

Tor は匿名化通信技術の一つである Onion Routing を実装しており, 多くのユーザーが利用, さらに協力している. Tor が提供するシステムはユーザーのブラウジングを匿名化するものの他に, Tor 秘匿サービス (Tor Hidden Service) と呼ばれるものがある. これは, サービスを提供しているサーバーの IP アドレスを隠すというシステムで

<sup>1</sup> 東京大学 生産技術研究所  
Institute of Industrial Science, University of Tokyo

ある。実際に用いられている Tor 秘匿サービスには人権運動組織や内部告発サイトのようなものがあり、人々の役に立っていると言える。しかしその一方で、麻薬売買サイトや誹謗中傷が多く書き込まれる掲示板などの悪質なサービスも存在し、問題となっている。Tor 秘匿サービスが注目されているなかで、サーバーの IP アドレスを暴く様々な攻撃が提案されてきた。Tor 通信の接続先を推定する攻撃で高い精度を挙げている、指紋攻撃を Tor 秘匿サービスの通信に適用し Tor 秘匿サービスプロトコル上のどの段階の通信か推定することで、注目しているノードが秘匿サービスか判定できる、Circuit Fingerprinting Attack (CFA) という攻撃を Kwon らが提案した [1]。

今日の多くの指紋攻撃は機械学習に基づいているため、ノイズを加える事で精度が大きく下がると言われている [14]。我々は、CFA に対する防御手法として、Dummy Hidden Service (DHS) を提案する。DHS は秘匿サービスと紛らわしいトラフィックを能動的に生成するノードである。Tor ネットワークに DHS が生成したトラフィックが混ざることによって、機械学習にとって有効といえるデータの割合を下げることが可能だと考えられる。しかし、Tor の開発陣、Tor project が多くのダミーパケットを挿入することを避けていることから、DHS は Tor ネットワークとノードに小さい負担で指紋攻撃の精度を下げる必要がある。そこで本論文では、ダミーパケットを Tor 秘匿サービスのトラフィックに加える事で、CFA の精度がどのように変化するか実験し調べた。

本論文の構成を述べる。まず、2 章で Tor の根幹となっている技術 Onion Routing、さらに Tor と Tor 秘匿サービスについて詳しく述べる。次に、3 章で Tor や Tor 秘匿サービスの匿名性を破る攻撃に関する論文を紹介する。4 章で Tor 秘匿サービスプロトコルの匿名性を高める手法について述べ、5 章では、Tor 秘匿サービスプロトコルのトラフィックにノイズを加え、既存の攻撃手法の精度の変化を調べた実験について述べる。そして 6 章で結論を述べる。

## 2. Tor と Tor 秘匿サービス

本章では Tor 及び Tor 秘匿サービスに用いられている技術と実装、さらに利用実態について解説する。

### 2.1 Onion Routing

Goldschlag らは匿名化通信技術の一つとして Onion Routing を提案した [4]。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ーとサーバーがやり取りする。Onion Routing はこの中継ノードに Relay と呼ばれるノードを含める。図 1 に Onion Routing の概要を示す。まずユーザーは各 Relay と鍵を共有しておく。通信の際には、ユーザーは各 Relay と共有した鍵でメッセージを多重に暗号化しておく。暗号文を受け取った Relay は

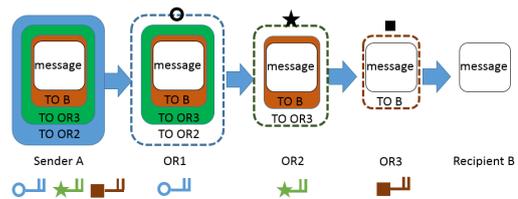


図 1 Onion Routing の概要

Fig. 1 Overview of Onion Routing

自分の持っている鍵で復号するが、ユーザーは暗号化の際に復号する Relay が次に渡すべきノードのアドレスを含めておく。すると Relay は隣の Relay がわかるのでこれを渡し、受け取った Relay は同様に自分の鍵で復号を行い、次の Relay に渡す。最終的にサーバーに平文が送られる。このようにすることで、ユーザー以外の各ノードは自分の隣のノードのアドレスしか知ることが出来ず、ユーザーとサーバーの繋がりが秘匿される。Onion Routing では各ノードを完全に信用できなくとも単体ではユーザーとサーバーの繋がりを暴くことは出来ない。このようにして、サーバーを誰が閲覧しているのかが匿名化される。

### 2.2 Tor (The onion router)

Tor は Onion Routing を実装した、非常に大規模な実システムである。2016 年 8 月時点で 7000 個程の有志の Relay が存在し \*1、利用者数は 160 万人を超えている。

Tor Network に接続するにはまず、Tor クライアントは Directory Authority と呼ばれる Relay から、Consensus File をダウンロードする。これには利用可能な Relay の情報が含まれており、この中からクライアントは中継ノードを 3 つ選択する。次にクライアントはそれぞれの Relay と鍵共有をした後、メッセージを多重に暗号化して通信を行う。この生成されたリンクは Tor circuit と呼ばれる。Tor circuit においてユーザーが一番近い Relay は Entry Guard と呼ばれる。Entry Guard は、ユーザーが接続したいサーバーのアドレスはわからないもののユーザーの IP アドレスを知っているため、匿名化通信においては非常に重要なノードと言える。悪意のある Relay が簡単に Entry Guard になることを防ぐために、このノードは Guard フラグを与えられた Relay のみが選ばれる。一方、サーバーが一番近い Relay を Exit Relay と呼び、間の Relay は Middle Relay と呼ばれる。

Tor project によれば Tor のデータ転送においては、パケットは 1 つが 512byte の固定長セルである。固定長を用いない場合、通信のトラフィック・パターンを観測することでどのようなウェブサイトを訪問しているのかを推定することが容易になるおそれがある。また、実際には 512 の倍数のサイズのパケットが使われることがある。

\*1 <https://metrics.torproject.org/>

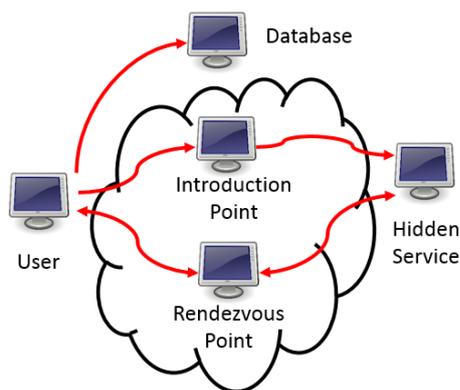


図 2 Tor 秘匿サービスの概要

Fig. 2 Overview of Hidden Service Protocol

### 2.3 Tor 秘匿サービス

Tor 秘匿サービス (Tor hidden service, HS) は Tor Network を用いて、サービスを提供しているサーバーの IP アドレスを隠すシステムである。メインとなるノードは図 2 の database, introduction point (IP), rendezvous point (RP) である。本節では Tor 秘匿サービスのプロトコルについて説明する。まず、HS は Tor Network 上の Relay から 3 つの IP を選び、生成した公開鍵を送る。ここで注意すべきは、Tor 秘匿サービスのプロトコルにおいて、全てのリンクが Tor Network 上で行われる点、つまり Tor circuit であるという点である。次に、HS は HS descriptor を生成する。HS descriptor には、HS の公開鍵と IP の IP アドレスの情報が含まれる。この descriptor と、decrypter を秘密鍵で署名したものを HS directory (HSDir) と呼ばれる Relay に送信する。図 2 において、HSDir は database とかかれたノードである。この時、descriptor に含まれる公開鍵から 16 桁英数字の onion アドレスが生成される。onion アドレスは HS を表す役割を担うが、文字列と HS を紐付けるためのみ用いられ、HS の IP アドレスの情報は含まれない。これで HS の登録が完了する。

次に、クライアントが HS に接続するプロトコルについて説明する。まずクライアントは接続したい秘匿サービスの onion アドレスを知る必要がある。何らかの方法で onion アドレスを入手したクライアントは、HSDir に秘匿サービスの descriptor を問い合わせる。問い合わせを受けた HSDir は、onion アドレスが存在すればその descriptor を返す。クライアントはここで Tor Network の Relay から RP を 1 つ選択する。RP は、クライアントと HS の中継を担う重要なノードであり、要求に対する返事としてワンタイムシークレットなクッキーを返す。RP のアドレスと、ワンタイムシークレットを HS の公開鍵で暗号化した introduce message を IP に送信する。同時に、RP にもワンタイムシークレットを送信しておき、後で正しく HS と

リンクがはられたのか確認する。IP は以前の Tor circuit を用いて HS に introduce message を送信する。秘匿サービスは自身の秘密鍵で受け取った introduce message を復号し、RP にワンタイムシークレットを送信する。RP はワンタイムシークレットを確認して、クライアントに正しくリンクが生成されたことを伝える。そして RP を中継して通信が開始される。

ここで注意すべきは、HS と RP の間の Tor circuit は HS に一番近い Relay が Entry Guard, RP に一番近い Relay が Exit Relay であり、Entry Guard は毎回同じセットの中から選ぶということである。そのような Tor circuit になっていない場合、タイミング攻撃と呼ばれる攻撃によって HS のアドレスを推定する手法が、Overlier と Syverson によって提案された [5]。この攻撃については次章で説明する。

## 3. 関連研究

本章では、Tor の匿名性を暴く既存の攻撃について説明する。Tor はユーザーとユーザーが閲覧しているウェブサイトの繋がりを秘匿することで匿名性を担保しているため、この繋がりを暴くことが匿名性を暴くことに繋がる。Wright らは匿名通信システムにおける攻撃の性能評価と防御手法について検討した [7]。Matic らは秘匿サービスの中から推定に使えるがサービス側が認識していない情報を探し、推定に用いた [11]。

### 3.1 指紋攻撃

トラフィック解析攻撃の中で近年研究が盛んな攻撃に、指紋攻撃が存在する [12], [13], [15], [16]。指紋攻撃とは、ウェブサイトと通信した際に観測できるトラフィック・パターンを Website Fingerprint (WF) とし、これをウェブサイトを表す特徴として推定に用いる攻撃である。指紋攻撃における攻撃者は、ユーザーの暗号化または匿名化通信を観測できる位置にいることを仮定として、ユーザーが閲覧しているウェブサイトを推定する受動的攻撃である。指紋攻撃を用いることで、Tor が秘匿しているユーザーとウェブサイトの繋がりを攻撃することが出来る。Tor においては、攻撃者は Entry Guard かネットワーク管理者、またはインターネットサービスプロバイダー等ユーザーのアドレスを知っていて且つユーザーのトラフィックを観測できる攻撃者である。攻撃者は Tor ブラウザのバージョン等ユーザーと同様の環境を構築し、自らウェブサイトを訪問しそのトラフィックを観測する。そして用いる攻撃手法に適した特徴量抽出を行い、これを用いて分類器に学習させておく。改めてユーザーの Tor を用いたトラフィックを観測し、そのトラフィックから得られる特徴量を分類器にかけて訪問しているウェブサイトを推定する。なお、Tor パケットは 2.2 節で説明したとおり 512byte の固定長セルで

あり，単体のパケットを観測して得られる情報は向きと時刻だけである。

Juarez らによれば，指紋攻撃には 6 つの仮定が存在する [6]。

- Closed-world: 訪問されるウェブサイトはせいぜい  $k$  個であるという仮定である。この仮定は非常に強く， $k$  は実際のウェブサイトの数と比べ非常に低い値にされるからである。これに対し，open-world では被害者は分類器の学習データセットに含まれないウェブサイトも訪問するというシナリオであるが，Juarez らは評価用のデータセットに含まれないウェブサイトは訪問されない点を指摘している。
- Browsing behaviour: ユーザーは特定の行動をするという仮定である。例えば，ユーザーは同時には 1 つのタブのみで閲覧を行い，1 つのウェブページから次のウェブページへ遷移するというシナリオが存在する。しかし，現実のユーザーは同時に複数のタブを用いてブラウジングを行うことが多い。Tor を用いたブラウジングのデータは公開されていないが，普通のブラウジングより遅いため，Tor においては同時に 1 つのタブで閲覧している，またはトラフィック解析においては区別可能であると Juarez らは言っている。
- Template websites: 全てのウェブサイトがテンプレートを用いて作られるという仮定である。Cai らは Hidden Markov Model を用いてウェブページの繋がりをモデル化し，指紋攻撃に適用した [15]。
- Page load parsing: 攻撃者はトラフィックを観測してユーザーのページ読み込みの始まりと終わりを認識できるとする仮定である。2.2 節で説明した通り Tor のパケットは 512 バイトの固定長セルであり，暗号化されているため，攻撃者がトラフィック・パターンを観測しただけでページの区切りが分かるわけではなく，この推定は容易ではない [10]。
- No background traffic: 攻撃者は，他のアプリケーションによるトラフィックや同じ Tor circuit を流れるトラフィックといったノイズを全て区別出来，これを観測から取り除くことが出来るという仮定である。Tor Network に接続する手段は Tor ブラウザを使うだけでなく，様々な方法が存在し，複数のアプリケーションから Tor Network に接続することが可能である。例えば，Talis\*<sup>2</sup> は全てのインターネット通信を Tor Network を通して行う。このような状況では，注目したいトラフィックのみを選択することは困難である。
- Repicability: 攻撃者は被害者と同じ環境を構築できるという仮定である。指紋攻撃において，攻撃者は予め Tor Network を通してウェブサイトを訪問しトラ

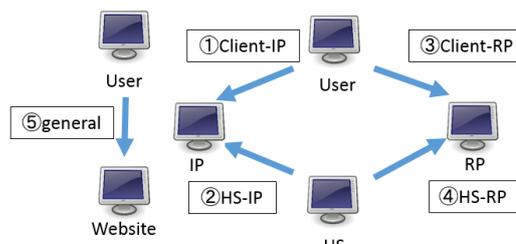


図 3 Tor 秘匿サービスプロトコルにおける 5 種類の Circuit  
Fig. 3 Five Circuits on Tor Hidden Service Protocol

フィックを解析しておくが，この環境が攻撃対象と同じでなければ正確なデータセットではなくなり，推定精度が落ちる可能性がある。環境としてはオペレーティングシステムやネットワーク接続，そして Tor ブラウザのバージョンといったものが挙げられる。

指紋攻撃の研究は，分類器に用いる学習アルゴリズムや特徴量を変えて推定精度を上げるだけではなく，攻撃者の仮定をより弱いものにし現実的なものに近づけることも重要である [14]。

Hermann らはナイーブベイズを用いた指紋攻撃の攻撃手法を提案した [3]。Pachenko らは Support Vector Machine を用い，775 個の close world のデータセットにおいて 50% 以上の精度で推定した [2]。Wang らは， $k$  近傍法を用いて非常に高い精度で攻撃に成功した [13]。

### 3.2 秘匿サービスの匿名性に対する攻撃

本節では，秘匿サービスが担保するサーバーの匿名性を暴く攻撃について紹介する。

#### 3.2.1 秘匿サービスに対するタイミング攻撃

攻撃者は予め多くの Relay を占拠しておく。Overlier の攻撃の仮定では，この Relay は Entry Guard である必要はないので，攻撃者自身のもつ Relay でも多く用意することが可能である。次に，攻撃したい秘匿サービスに多くのクライアントから接続する。もし秘匿サービスと rendezvous point の circuit において，秘匿サービスに一番近いノードが攻撃者の占拠した Relay ならば，クライアントからパケットを送信した際に短い時間の間に必ず攻撃者の Relay がこれを中継するので，攻撃者はこれを観測すれば Relay が秘匿サービスに接続していると推定する事が出来て，その Relay は秘匿サービスのアドレスを知っているため，秘匿サービスのアドレスが割り出されるという流れで攻撃が成功する。

#### 3.2.2 Circuit Fingerprinting Attack (CFA)

Kwon らは，トラフィック解析によって秘匿サービスを推定する受動的な攻撃を提案した [1]。攻撃者は攻撃対象と Entry Guard の通信を観測できる攻撃者であり，トラフィックから攻撃対象の IP アドレスを知ることが出来

\*2 <https://tails.boum.org/>

る。Kwon らの攻撃はまず観測した Tor Circuit がプロトコルで現れる複数の Circuit うちの Circuit であるかをトラフィックからクラス分類する。Circuit のクラスがわかればエンドノードが秘匿サービスかクライアントか判定できる。そして秘匿サービスと推定したならば、そのノードに対して指紋攻撃をしかけ秘匿サービスを特定する。このようにして、秘匿サービスと IP アドレスを紐付け、匿名性を破ることが出来る。Kwon らの攻撃は、Circuit の種類の推定、指紋攻撃の二つのプロセスから成り、論文は両プロセスの提案、実験について述べられた。前半の Circuit の分類を行う攻撃を特に Circuit Fingerprinting Attack という。

Kwon らによれば、図 3 に表した 5 つのリンクは判別可能である。攻撃者はトラフィックを観測することで自分の見ている circuit が、Tor circuit を用いた普通のウェブサイトの閲覧 (general) か、Client-rendezvous Point (Client-RP) か、Client-Introduction Point (Client-IP) か、Hidden Service-rendezvous Point (HS-RP) か、Hidden service-Introduction Point (HS-IP) か推定することが出来る。彼らは推定に用いる特徴量として 3 つの特徴量を用いた。

- 内向き外向きパケット: それぞれの circuit によって内向き、外向きパケットの数に傾向があるが、ここでいう内向きとは図 3 の Client または HS へ向かう向きが内向き、IP または RP へ向かう向きが外向きである。図 3 で IP とつながりがある Client-IP と HS-IP の判別に特に有効である。Client-IP ではそれぞれの向きのパケットの数が同じになるという特徴がある。彼らは今回は通信の最初の 50 個のパケットにおいて、それぞれの向きの数を計算し特徴とした。
- Duration of Activity: 注目している circuit で通信が行われている時間であり、Client-IP と HS-IP またはその他を区別するのに有効である。IP とのやりとりはプロトコル上長く行われるものではない一方で、実際のやりとりである Client-RP, HS-RP, general の 3 つのについてはより長い時間のやりとりとなる。
- サーキット構築シーケンス: 通信の最初パケットは circuit 構築のためのパケットであり、また Circuit の種類によってセルの数やプロセスが異なるため判別する特徴として利用できる。Kwon らは通信の最初の 10 個のセルの向きのシーケンスを特徴とした。

これらの特徴量を用いて機械学習を行う。Kwon らの実験では Tor 秘匿サービスプロトコルの 4 つの Circuit と、Tor を用いた一般のウェブサイトの閲覧の Circuit のトラフィックデータを収集し、交差検定で CFA の性能評価を行った。ある Circuit のクラスに注目した時、注目クラスの Circuit のトラフィックを、正しくクラス分類した場合を True Positive (TP)、間違えた場合を False Negative (FN)、また注目したクラスと別の Circuit のトラフィックを、注目ク

ラスと別のクラスに分類した場合を True Negative (TN)、注目クラスに分類した場合を False Positive と呼ぶ。Kwon らは評価指標に True Positive Rate ( $TPR = \frac{TP}{TP+FN}$ ), False Positive Rate ( $FPR = \frac{FP}{TN+FP}$ ) を用いた。この指標を全ての検定で平均をとったところ、TPR が 98%, FPR は 0.1%未満という結果になった。

## 4. Dummy Hidden Service

CFA はラップトップでも簡単に行うことが出来る上、受動的な攻撃であるためコストが低い。このように機械学習を用いたトラフィック分析は Tor には効果的な攻撃と言えるため、対策を提案する必要がある。一方で機械学習を用いた手法はノイズ耐性が低いと Kwon らは述べている。そこで我々は、秘匿サービスの匿名性をより頑強にするため、Dummy Hidden Service (DHS) を提案する。DHS は秘匿サービスと紛らわしいトラフィックを作り出すノードである。DHS が作ったトラフィックが学習データにまぎることで、実際に秘匿サービスであるか推定する精度を下げることを目的とする。DHS は多くの有志が Tor Network 上に接続しておく形で使用することを想定しており、ウェブサイトの指紋攻撃のタスクにおけるノイズパディングとは異なる。DHS は四個の要件を満たす必要がある。

- 低コスト: DHS を運用することは端末にとってなるべく少ない負担である必要がある。さらに、DHS のトラフィックも Tor Network 上には必要最低限である必要がある
- 導入の容易さ: 多くの協力を得るために、DHS は簡単に導入出来る実装である必要がある。
- 高性能: DHS のトラフィックは機械学習を用いた攻撃によって秘匿サービスと判定される必要がある。
- 雑音: ウェブサイト、または秘匿サービスの指紋攻撃によって特定のサーバーと推定されない必要がある。

## 5. 実験

DHS が 4 章で述べた要件を満たすために、我々は今回、実際の Tor 秘匿サービスプロトコル上のトラフィックに対してノイズを加え、機械学習の精度がどのように変化するか定性的な性質を調べた。導入の容易さを考えると、一番簡単な実装では外向きパケットをランダムに選んだ Entry Guard に送信するだけのノードが考えられる。そのため、今回加えるノイズは外向きパケットのみに限定する。

### 5.1 攻撃者モデル

今回の実験において、攻撃者は秘匿サービスまたはクライアントと Entry Guard の間の全てのトラフィックを観測できるが、観測以外のことはしない。Kwon らの論文における 2 つの攻撃プロセスのうち、前半のプロセスである CFA を行い、Circuit が RP-Client, RP-HS, general のど

れかなのかをの推定する。推定には機械学習を用い、学習モデルはk近傍法とする。KwonらのCFAにおいては、IPとの通信するCircuitや、Circuit構築シーケンスを考慮していた。しかし実際の通信においては、毎回Circuit構築シーケンスは行われるわけではなく、Circuitはしばらく保存される。そこで今回は、Client-RPとHS-RPそしてgeneralの通信に注目し、Circuit構築シーケンスは考慮しない。

今回の実験では最初の50個のパケットの向きに注目した三次元の特徴量を用いた。最初の50個のパケットのうち、内向きパケットの数、外向きパケットの数、そして三つ目の特徴量は内向きパケットと外向きパケットの比である。

## 5.2 被害者モデル

現実のTor Network上のトラフィックは多種のウェブサイトの閲覧によるものであり、また秘匿サービスの数も非常に多い。しかし、今回はノイズの影響を定性的に調べるため、3.1節で説明した、Closed-Worldの設定で実験を行う。攻撃者は予め、評価用データセットに用いられる全てのCircuitのトラフィックを観測し、学習することが出来る。

## 5.3 実験環境

### 5.3.1 データセット

実際にTor Network上の通信を行い、観測することでデータを集める。サーバーマシンとして、仮想OS上に4個の秘匿サービスをたてる。それぞれの内容は画像を含むものが二つ、オープンCMSを用いたブログ形式のホームページに何も手を加えていないものが二つである。クライアントマシンはサーバーマシンと同じ仮想OSを用いる。つまりインターネット上の位置はクライアントマシンとホストマシンで同じであるが、Tor秘匿サービスプロトコル上では別のCircuitが用いられるため、別のマシンを用いたデータとTorプロトコル上では差がないといえる。

Torブラウザから各秘匿サービスに一度接続を行い、Circuitを構築する。それぞれ40回接続し、クライアントの外向き内向きパケットと、秘匿サービスの外向き内向きパケットを読み込みが完了するまで観測する。

また、一般のウェブサイトの閲覧におけるCircuitの観測も行う。ネットワークアナリシスサービスの一つ、Alexaが公開しているアクセスランキングの上位から対象とするウェブサイトを6つ選択し、秘匿サービスと同様に40回の読み込みを観測した。今回選んだのはgoogle.com, youtube.com, facebook.com, yahoo.com, wikipedia.org, amazon.comである。

### 5.3.2 実験環境

サーバーマシン、クライアントマシンはともにVMware上でUbuntu14.04LTSを用いた。仮想マシンのメモリは

3GBである。またTorのバージョンは0.2.7、Torブラウザのバージョンは6.0.3である。秘匿サービスのためのホームページはLAMP環境を用意し、apache2.4を用いた。観測にはtcpdumpコマンドを使用し、クライアント側とサーバー側のCircuitそれぞれのEntry Guardとの通信を観測する。学習はデータマイニングツールの一つ、Weka[8]を用いる。

## 5.4 ノイズ (防御者モデル)

今回は、秘匿サービスとクライアント間の通信と、ウェブサイトとクライアント間の通信に対し、クライアント利用者がノイズを加えることを考える。加えるノイズは外向きパケットのみに限定する。外向きパケットのノイズをクライアント利用者が生成することは、現在のTor Networkの環境でも行うことが出来る。防御としては、ノイズを加えたデータを学習させることで、HS-RPの推定精度を下げることが目的とする。今回は以下の三種類のノイズで実験を行いHS-RPの推定精度を調べる。

- ランダムノイズ - 各パケット (Random-Each): 1パケットごとにランダムでパケットを挿入する。ここで、ランダムとは確率 $p$ で生起する事象であり、 $p$ は実験ごとに自由に設定できるパラメータである。この手法では、ノイズは最大で2倍になる。
- ランダムノイズ - 1クラスタ (Random-Cluster): 同じタイムスタンプの連続したパケットをひとまとまりとし、その後ろにノイズを挿入する。ひとまとまりのパケットのサイズを $N$ とし、乱数 $r$ は $r = (0.0, 1.0]$ をとるとする。そしてノイズを加えたパケットのサイズ $L$ は元のパケットのサイズ $N$ の $M$ 倍が最大であるという制約をつけると、

$$L = N * (M - 1) * r \quad (1)$$

となる。実験ごとに $M$ を変化させて精度の変化を見る。

- 擬態ノイズ (Morph): 一番最初のパケットはノイズにしないという制約のもと、元のパケットに外向きパケットのみを加えて、先頭の4パケットを内向き、外向き、内向き、外向きになるように近づける。そして、次のタイムスタンプがことなるパケットのうしろに外向きパケットを加える。今回観測したHS-RPのトラフィックを見ると、先頭の4パケットが内、外、内、外のシーケンスである傾向が見られた。これに対しクライアント側のトラフィックは、一番最初のパケットが外向きであることが多く、また加えるノイズは外向きに制限しているため、完全に一致させることは難しい。加え方としては、例えば元のパケットが「外、外、内、内」であれば、3パケット目と4パケット目の間に外向きパケットを加えると、最初の4パケットは「外、

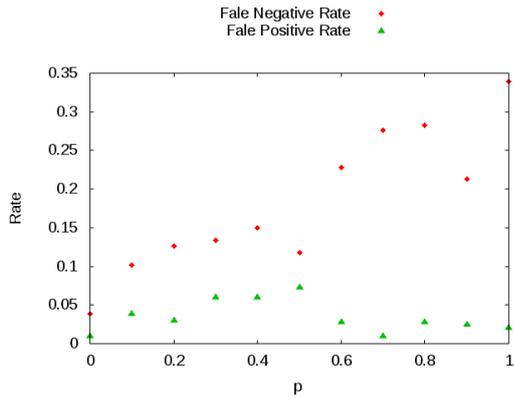


図 4 Random-Each の結果  
Fig. 4 The result of Random-Each

外, 内, 外」となり 3 パケットが一致する。このノイズに設定するパラメータとして、後ろに加えるパケットのサイズを実験ごとに自由に設定できる。

### 5.5 評価手法

今回の実験では、Kwon らの実験のうち、観測している Circuit が Client-RP, HS-RP, もしくは general であるかを判定する。学習、推定に用いる k 近傍法は、最近接ノードだけからクラス分類を行い、また重みは距離の逆数を用いる。HS-RP のデータに対し、HS-RP と推定できなかったものを False Negative, HS-RP ではないデータに対し HS-RP と推定された場合には False Positive とする。実験データセットは 5.3.1 項で述べたとおりで、Client-RP が 240 個, HS-RP が 240 個, general が 360 個である。5.4 節で述べたノイズそれぞれの手法に対して、パラメータを動かしながら Kwon らの CFA を行う。評価としては下式で表した False Positive Rate (FPR), False Negative Rate (FNR) を用いる。

$$FPR = \frac{\# \text{ of False Positive}}{\# \text{ of Client-RP} + \# \text{ of general}} \quad (2)$$

$$FNR = \frac{\# \text{ of False Negative}}{\# \text{ of HS-RP}} \quad (3)$$

### 5.6 結果

まず、図 4 は Random-Each の結果である。横軸はノイズの生起確率  $p$  である。 $p$  が 0.4 に近づくまでは FNR, FPR ともに徐々に上昇しているが、FPR は  $p = 0.5$  から下降し始める。一方で、FNR については  $p = 0.9$  で大きく下降するほかは上昇する。Random-Each は 1 パケットごとにランダムノイズを入れる機会があるため、徐々に外向きパケットの割合が増えてくる。これにより秘匿サービスの中で外向きと内向きパケットの割合が近いものがノイズパケットと間違われるようになったが、HS-RP に似たトラフィックを作り出せず False Negative は上昇しなかったと考えられる。

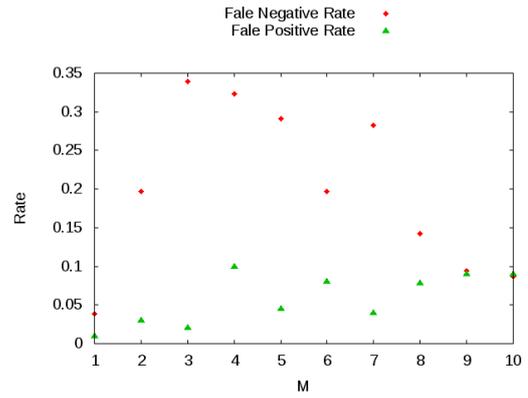


図 5 Random-Cluster の結果  
Fig. 5 The result of Random-Cluster

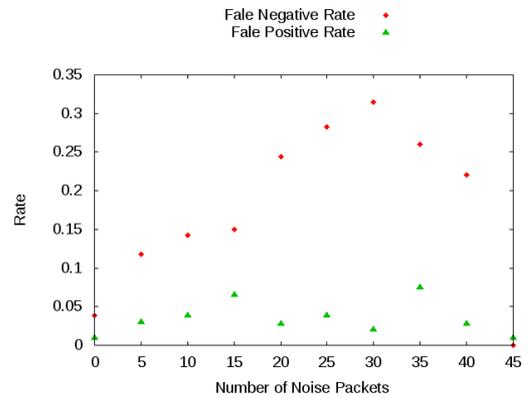


図 6 Morph の結果  
Fig. 6 The result of Morph

一方、図 5 は Random-Cluster の結果である。横軸  $M$  は、ノイズを入れたことでパケットが大きくなる上限ののパケットと比べた割合である。 $M = 2, 3$  では Random-Each と同様の傾向が見られるが、その後  $M = 5, 7$  以外では FPR が上昇する。一方で FNR に関しては  $M = 6$  で大きく下降し、 $M = 8$  以降は減少していく。FPR が上昇した  $M$  に注目すると、その  $M$  では FNR が減少している。今回はデータ取得に用いた秘匿サービス、ウェブサイトの種類が少なかったために、ノイズが加わった特定のウェブサイトのトラフィックが、ある秘匿サービスのトラフィックに近くなったと考えられる。そのため、学習の結果、HS-RP と判定されるデータが増えた事が FPR, FNR の上下につながった。

最後に、Morph の結果について図 6 に示す。FNR についてはノイズパケット数が 30 個近くになるまでは上昇するが、その後減少し始め、ノイズパケット数が 45 個の時には FNR が 0 になっている。FPR に関しては、Random-Each, Random-Cluster と同様特定の HS-RP と近いトラフィックが生成されたかどうかで上下している。FNR について、ノイズパケットが 30 個を超えると外向きが多すぎてしまい、45 個ともなると、もはや HS-RP にはない特徴として扱われ、かえって HS-RP は正しく推定されるということ

になったと考えられる。

## 5.7 考察

どの結果を見ても、FNR は 30%を超える程度であり、FPR に関しては 10%ほどしかでなかった。この原因として、用いた秘匿サービス、ウェブサイトの種類が少なく、トラフィックも特徴的であり、かつ接続回数が 40 回と多かったため、トラフィックに対する CFA 自体がウェブサイトの指紋攻撃と同じ働きをした。それによって FPR に関してある程度ノイズへの耐性が得られたと考えられる。また、今回加えたノイズはクライアント側だけが外向きパケットのみを加えるであるが、もし HS-RP のトラフィックや内向きパケットのノイズを使用できるモデルならば、CFA の推定精度により大きな影響が現れると考えられる。

最後に、より工夫した攻撃者モデルに触れる。今回の攻撃者は先頭 50 個の外向き、内向きパケットの数にのみ注目していたが、先頭 10 個のパケットの向きを特徴量に加え各手法のノイズを加えたデータに対し同様の実験を行った所、FPR は最大で 5.5%、FNR は最大で 1.0%となった。各秘匿サービス、ウェブサイトのトラフィックを学習する十分なデータ量があったことでノイズ耐性が高くなっていることが原因として考えられるが、パケットの向きのシーケンスが特徴量として重要であることも確認できる。

## 6. 結論

本論文では、秘匿サービスの匿名性を暴く攻撃の一つである Circuit 推定に対する防御手法として Dummy Hidden Service (DHS) を提案し、DHS の実装のためノイズを Tor 秘匿サービスプロトコル上のトラフィックに加え Circuit 推定の推定精度にどのような影響が出るか調べた。実験によって、ノイズを加え過ぎると逆に特徴的なトラフィックになり Circuit 推定の精度を上げること、クライアント側が外向きパケットのノイズのみを生成するという防御モデルでは十分に Circuit 推定の精度を下げる事が出来ないことが分かった。また、今後の実験では秘匿サービスもノイズ生成が可能であり、且つ内向きノイズの生成も可能である防御モデルについて検討していく。その際に、小規模な Closed-world において Circuit 推定はウェブサイトの指紋攻撃と同等の働きをしてしまうため、被害者モデルはより大規模なもの、そして Open-world の設定を考慮する必要があると今回の実験から言える。

## 参考文献

- [1] Albert Kwon, Mashaal AlSabah, David Lazar, Marc Dacier and Srinivas Devadas, *Circuit Fingerprinting Attacks: Passive De-anonymization of Tor Hidden Services*, In Proceeding of the 24th USENIX Security Symposium, (2015).
- [2] Andriy Panchenko, Lukas Niese, 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, (2011).
- [3] Dominik Herrmann, Rolf Wendolsky and Hannes Federath, *Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier*, In Proceedings of the 2009 ACM workshop on Cloud computing security, (2009).
- [4] David M. Goldschlag, Michael G. Reed and Paul F. Syverson, *Hiding Routing Information*, In Proceeding of the First International Workshop on Information Hiding, (1996).
- [5] Lasse Øverlier and Paul Syverson, *Locating Hidden Servers*, In Proceedings of the 2006 IEEE Symposium on Security and Privacy, (2006).
- [6] Marc Juarez, Sadia Afroz, Gunes Acar and Claudia Diaz, *A Critical Evaluation of Website Fingerprinting Attacks*, In Proceedings of the 2014 ACM Conference on Computer and communications security, (2014).
- [7] Matthew Wright, Micah Adler, Brian N. Levine and Clay Shields, *Defending Anonymous communications Against Passive Logging Attacks*, In Proceedings of the 2003 IEEE Symposium on Security and Privacy, (2003).
- [8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, *The WEKA data mining software: an update*, In ACM SIGKDD Explorations Newsletter, (2009).
- [9] Roger Dingledine, *Tor and Circumvention: Lessons Learned*, In the 31th International Cryptology Conference, (2011).
- [10] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose and M. K. Reiter, *On web browsing privacy in anonymized NetFlows*, In Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, (2007).
- [11] Srdjan Matic, Platon Kotzias and Juan Caballero, *CARONTE: Detecting Location Leaks for De-anonymizing Tor Hidden Services*, In Proceedings of the 2015 ACM SIGSAC Conference on Computer and Communications Security, (2015).
- [12] Tao Wang and Ian Goldberg, *Improved website fingerprinting on Tor*, In Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society, (2013).
- [13] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson and Ian Goldberg, *Effective attacks and provable defenses for website fingerprinting*, In Proceedings of the 23rd USENIX conference on Security Symposium, (2014).
- [14] Tao Wang and Ian Goldberg, *On Realistically Attacking Tor with Website Fingerprinting*, Technical report, (2015).
- [15] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi and Rob Johnson, *Touching from a distance: website fingerprinting attacks and defenses*, In Proceedings of the 2012 ACM Conference on Computer and communications security, (2012).
- [16] Yi Shi and Kanta Matsuura, *Fingerprinting attack on the tor anonymity system*, In Proceedings of the 11th international conference on Information and Communications Security, (2009).