ドメイン内フィンガープリント攻撃に適した特徴量の実験的評価

濱中 圭吾[1] 小泉 佑揮[1] 武政 淳二[1] 長谷川 亨[1]

概要：盗聴したパケット列が持つ特徴から、被害者の参照中の Web ページを特定するウェブフィンガープリント攻撃が注目されている。従来は、被害者が参照する Web サイトを特定するドメイン間フィンガープリント攻撃が主流であったが、ある同一のドメイン中から被害者が参照している Web ページを特定するドメイン内フィンガープリント攻撃が可能であることが報告されている。本稿では、パケット列から抽出した特徴量が持つ情報量を定量的に評価することで、ドメイン内フィンガープリント攻撃に有効な特徴量を調査する。Amazon.co.jp を対象とした Web 通信のトレースデータを作成し、そのトレースを用いて特徴量が有する情報量を定量化することで、ドメイン内フィンガープリント攻撃に有効な特徴量候補を選出し、さらに、それらの特徴量が攻撃に有用である理由を分析する。

An Empirical Analysis on Features Effective in Intra-domain Web Fingerprinting

KEIGO HAMANAKA[1] YUKI KOIZUMI[1] JUNJI TAKEMASA[1] TORU HASEGAWA[1]

1. はじめに

攻撃者が、ユーザーのアクセスした Web サイトを推定する Web サイトフィンガープリンティング (WSF: web site fingerprinting) 攻撃がプライバシーの脅威として懸念されている [1], [2], [3]。WSF 攻撃はユーザーが特定の Web サイトにアクセスするときのパケット列を盗聴し、ペイロードではなくパケットサイズなど仮に暗号化された状態であっても観測可能な情報を用いてユーザーがアクセスした Web サイトを推定する攻撃である。したがって、HTTPS (TLS) のような暗号通信や Tor [4] のような匿名通信プロトコルでは防ぐことができない。こうした攻撃を防ぐ手法としては、ダミーパケットや遅延の挿入によって通信中の特徴を隠蔽する手法が提案されている [5], [6], [7], [8]。しかし、機械学習技術の発展などに伴い攻撃の精度は向上しており、ダミー挿入などの方法で防ぐことは困難なことが指摘されている [9]。

さらに、同一ドメイン内でユーザーが参照している Web ページを推定する Web ページフィンガープリンティング (WPF: web page fingerprinting) 攻撃が可能になった [10]。WSF 攻撃と WPF 攻撃は、攻撃の対象の違いの観点で、それぞれ、ドメイン間 Web フィンガープリント攻撃およびドメイン内ウェブフィンガープリント攻撃とも呼ばれる。本稿では、表記の簡単化のため、WSF 攻撃と WPF 攻撃と表記する。WPF 攻撃の成功は、ユーザーのプライバシーがより細粒度に暴露されるということから、大きなプライバシー上の脅威である。

一般的に、同一ドメインの Web ページは、同様の構成からなる Web ページであるため、WPF 攻撃は WSF 攻撃よりも困難である。しかし、Wang ら [10] は、ソーシャルメディアサイトを対象としたときに、そのページを構成するコンテンツが複数の CDN サーバーに分散されていることに着目し、通信先の CDN サーバーごとに分類したパケット列から得られる情報を特徴量として攻撃に用いることで WPF 攻撃を成功させた。

一方で、Wang ら [10] の攻撃を成功させるためには、CDN サーバーごとの通信を特定する必要がある。そのためには、攻撃者は盗聴したパケット列から通信先の IP アドレ

[1] 大阪大学
Osaka University

スを抽出する必要があり、Wi-Fi アクセスポイントに侵入する、IP ルーターに侵入するなどの攻撃準備が必要である。さらに、Tor など通信先を匿名化する防御手法により容易に防御が可能である。

本研究の目的は、より簡易な攻撃として Wi-Fi アクセスポイントから漏洩する無線フレームを盗聴する攻撃者を想定し、WPF 攻撃に有用な特徴量を調査することである。Wi-Fi における無線フレームの盗聴は容易であるのに加えて、携帯電話網でも、無線フレームの盗聴の可能性が指摘されており [11]、無線フレームによる WPF 攻撃の成功は大きな脅威である。

本研究では、Li ら [12] の方法を用いて、WPF 攻撃の観点でそれぞれの特徴量が持つ情報量を評価することで、WPF 攻撃に適した特徴量を分析する。評価の対象のドメインとして、ソーシャルメディアではなく、一般的な電子商取引サイトである Amazon を対象とする。なお、想定する攻撃環境では、通信先の IP アドレスが参照できないため、CDN サーバーごとにパケット列を分類できないが、本稿では、CDN サーバーごとに分類したパケット列から得られる特徴も参考データとして評価する。つまり、Li ら [12] が整理した 3043 種類の特徴量に、Wang ら [10] が提唱した CDN サーバーごとにパケット列から抽出可能な 50 の特徴量を加えた全 3093 種類の特徴量を対象として、WPF 攻撃に有用な特徴量を評価する。

本稿の構成は以下の通りである。まず、2 章で関連研究を整理する。3 章で WPF 攻撃と想定する攻撃者モデルを説明し、4 章で評価方法を説明する。5 章で WPF 攻撃に適した特徴量を調査する。最後に、6 章で本稿をまとめる。

2. 関連研究

WSF 攻撃については既に多くの研究が行われており、機械学習を応用した分類手法により、ネットワーク上から盗聴したトレース情報からユーザーがアクセスした Web サイトを高い精度で推定できることが指摘されている [1], [2], [3], [9]。中でも、Panchenko ら [3] が提案した転送パケットサイズの累積和が、高い攻撃精度を示している。さらに、Sirinam ら [9] の研究では、パケットの転送方向のみを特徴量としても、深層学習による分類により高い精度で参照している Web サイトが特定できることが示されている。また、これらの攻撃に対する防御手法も [5], [6], [7], [8] 提案されている。これに対して、本稿では、同一ドメインから参照しているページを推定する WPF 攻撃を対象とする。

これらの研究では、高い分類精度を実現できる特徴量をアドホックに評価しており、評価指標は主に分類器による分類精度を用いているのが現状である。これに対して、Li ら [12] は、分類器による分類精度に依拠した特徴量の評価は不十分であることを指摘している。例えば、ある特徴量を用いた分類には本来それに適した構造の分類器を用いる必要があるのに対して、同一の分類器による評価が主流であることを指摘している。これに対して Li らは、WSF 攻撃の観点で特徴量から漏洩する情報を相互情報量の観点で定量的に評価する方法を提案した。本稿では、Li らの方法を用いて WPF 攻撃に有効な特徴量を評価するものである。さらに、Li らの方法に加えて、特徴量のばらつきの観点からも、特徴量の WPF 攻撃に対する有用性を評価する。

3. Web Page Fingerprinting 攻撃

本章では、想定する WPF 攻撃と攻撃者のモデルを説明する。

WPF 攻撃と WSF 攻撃は、攻撃者がユーザーが Web ページを参照する際の通信を盗聴することで得たパケット列から、ユーザーが参照している Web ページを推定する攻撃である。一般的に、攻撃者は、事前準備として、ユーザーと似た通信環境から様々な Web ページへ通信をし、それぞれの通信のパケット列から特徴量を抽出し、それをもとに分類器を生成する。生成した分類器に対して、ユーザーの通信を盗聴することで得たパケット列から得た特徴量を与え、参照している Web ページを推定する。

攻撃者は、攻撃の第一段階として WSF 攻撃により通信している Web サイトを特定する。攻撃者がターゲットとする Web サイトへの通信を検出した場合に、第二段階として WPF 攻撃を適用する。本稿では、この攻撃の第二段階で用いる WPF 攻撃を想定して、特徴量を評価する。

攻撃モデルとして、ユーザーの Wi-Fi 通信を盗聴を想定する攻撃を想定する。攻撃者モデルを図 1 に示す。攻撃者は、ユーザーの Wi-Fi アクセスポイントから漏洩する L2 フレームをキャプチャーして得られたパケット列から特徴量を抽出して WSF 攻撃および WPF 攻撃を実施する。この環境下では、攻撃者は、ペイロード上のユーザーデータのみならず、送信元および送信先 IP アドレスも識別ができない。MAC アドレスは参照可能であるため、その無線フレームが送信パケットあるいは受信パケットのいずれのパケットを含んでいるかについては識別が可能である。これに加え、パケットの送受信時間やパケットサイズについても識別が可能である。

本稿で想定する WPF 攻撃と攻撃者をモデル化する。攻撃者の攻撃対象である Web サイトを構成する Web ページ群を $\mathcal{W}$ とする。攻撃者はその中からいくつかの Web ページ $\mathcal{V}$ ($\mathcal{V} \subset \mathcal{W}$) を監視対象とする。攻撃者は $\mathcal{V}$ 内のそれぞれのページ w_i ($\forall w_i \in \mathcal{V}$) に対して複数回アクセスをして、そのときのパケット集合であるトレース $t_{i,j}$ を収集し、学習用データセット $\mathcal{D}$ ($t_{ij} \in \mathcal{D}$) を生成する。$\mathcal{D}$ を用いて分類器 C を学習する。ある Web ページ w_i に対するトレースの集合を $\mathcal{T}_i$ と表記する。その後、攻撃者は、ユーザーが $w_i \in \mathcal{V}$ に対してアクセスするときの通信を盗聴するこ

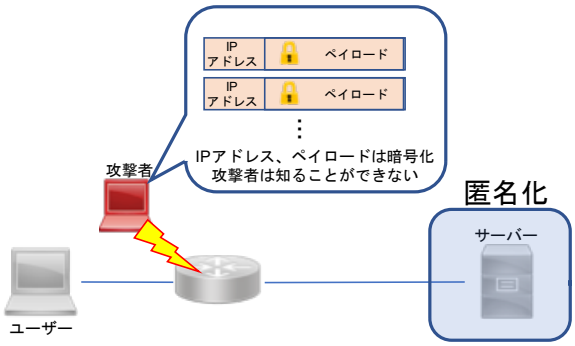


図 1 攻撃者モデル

とで得たトレースを a_{ij} $(a_{ij} \notin \mathcal{T}_i)$ から抽出した特徴量を C で分類することでユーザーが参照した Web ページを推定する。

4. 特徴量の評価法

本章では、特徴量の評価法と評価に用いる特徴量を定義する。

4.1 情報量

本研究では、Li [12] らの手法を用いて、ある Web ページを参照する通信についてのある特徴量を得たときに、その特徴量から漏洩する Web ページに関する情報量を定量化する。具体的には、攻撃者がある特徴量 F を観測したときにウェブページ W に関して得られる情報の量を、F を観測したときの W に関する相互情報量 $I(F;W)$ としてモデル化する。式は以下の通りである。

$$I(F;W) = H(W) - H(W|F) \tag{1}$$

$H(\cdot)$ はエントロピーを表す。実験では、この値を導出するため、確率密度関数 PDF(F)、および PDF(F,W) については、ガウスカーネル関数によるカーネル密度推定を用いて推定する。

4.2 特徴量の定義

本研究では、トレースから得られる特徴量として Li ら [12] が評価した 3043 種類の特徴量に加えて、CDN バースト [10] のサイズについて 50 次元の特徴量を含めた計 3093 種類の特徴量を評価する。紙面の都合上、全ての特徴量の定義を説明することはできないため、特に本稿で注目する 3 種類の特徴量、パケットサイズ累積和、累積パケット転送間隔、CDN バーストサイズについて具体的に説明する。

本稿で用いる表記法を表 1 にまとめる。ある Web ページにアクセスするときの通信を構成するパケットの集合表記を用いて $\mathcal{P}$ と表記する。なお、t_{ij} や a_{ij} もパケットの集合を意味するが、本章の特徴量の定義においては、攻撃者あるいはユーザーの通信から得られたパケット列を区別せずに、一般的に $\mathcal{P}$ と表記する。$\mathcal{P}$ に含まれるパケットを

表 1 本稿で用いる表記のまとめ

記号	意味
$\mathcal{W}$	攻撃対象 Web サイトを構成する Web ページの集合
$\mathcal{V}$	$\mathcal{W}$ 内で攻撃対象とする Web ページの集合
w_i	Web ページ
$\mathcal{T}_i$	攻撃者が分類器生成用に用いた Web ページ w_i を参照したときの通信のトレースの集合
t_{ij}	攻撃者が生成した通信のトレース $(t_{ij} \in \mathcal{T}_i)$
a_{ij}	ユーザーが w_i にアクセスしたときの通信のトレース
$\mathcal{P}$	ある Web サイトにアクセスするときのパケットの集合
$\mathcal{P}_{\text{in}}$	ある Web サイトにアクセスするときの受信パケットの集合
$\mathcal{P}_{\text{out}}$	ある Web サイトにアクセスするときの送信パケットの集合
p_i	パケット
s_i	パケット p_i のサイズ
d_i	パケット p_i の送信・受信方向
x_i	パケット p_i を観測した時間
y_i	パケット p_{i-1} と p_i の到着間隔
α_i	パケット p_i に記載されたユーザーの通信先の IP アドレス
S	累積パケットサイズ
S_{in}	受信パケットに関する累積パケットサイズ
S_{out}	送信パケットに関する累積パケットサイズ
I	累積パケット転送間隔

観測時間の早い順に $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ とする。

パケットサイズ累積和については、送受信パケットを混合した定義に加え、送信パケットあるいは受信パケットのみを対象とした 2 つの定義を考える。ある Web ページにアクセスするときの通信を構成する受信パケットと送信パケットの集合をそれぞれ $\mathcal{P}_{in}$、$\mathcal{P}_{\text{out}}$ とする。このとき、$\mathcal{P} = \mathcal{P}_{\text{in}} \cup \mathcal{P}_{\text{out}}$ かつ $\mathcal{P}_{\text{in}} \cap \mathcal{P}_{\text{out}} = \emptyset$ となる。変数 d_i をパケット p_i の方向を示す変数とし、p_i が受信パケットであれば $d_i = 1$、送信パケットであれば $d_i = -1$ とする。パケットサイズの累積和 S を以下の通りに定義する。

$$S(i) = \sum_{p_j \in \mathcal{P},\ j \le i} d_j s_j \tag{2}$$

$S(i)$ の推移から、等間隔に 50 箇所の値をサンプルし、それぞれのサンプルを特徴量とする。受信パケットおよび送信パケットに関するパケットサイズの累積和を、それぞれ S_{in}、S_{out} とすると、以下のように定義する。

$$S_{\text{in}}(i) = \sum_{p_j \in \mathcal{P}_{\text{in}},\ j \le i} d_j s_j \tag{3}$$

$$S_{\text{out}}(i) = \sum_{p_j \in \mathcal{P}_{\text{out}},\ j \le i} d_j s_j \tag{4}$$

同様に、このパケットサイズ累積和の推移から、等間隔に 50 箇所の値をサンプルし、それぞれを特徴量とする。

続いて、CDN バーストを定義する前にバーストを定義する。パケット p_i を観測した時間を x_i、パケット p_{i-1} と p_i の到着間隔を y_i とする。$y_i = x_{i-1} - x_i$ $(i > 1)$ であり、$y_1 = 0$ とする。バーストは、連続して y_i がある閾値

δ 以下となるパケットの集合とする。CDN バーストは、ユーザーと対向側の通信先の IP アドレスごとにパケット列を分類した後に、抽出したバーストである。実験では、$\delta = 0.05$ と設定した。CDN バーストに含まれるパケットサイズの合計を CDN バーストサイズとする。CDN バーストのサイズを降順にソートしたときの上位 50 個のそれぞれを CDN バーストサイズに関する特徴量とする。

最後に、累積パケット転送間隔を定義する。累積パケット転送間隔は、以下で定義する。

$$I(i) = \sum_{p_j \in \mathcal{P},\ j \leq i} y_j \tag{5}$$

これまでの特徴量と同様に、上式の累積和の推移から等間隔に 50 箇所の値をサンプルし、それぞれを特徴量とする。

5. 特徴量の評価

前章で定義した評価法を用いて、WPF 攻撃に有用な特徴量を評価する。

5.1 評価の概要

本稿では、次の 2 つのステップで、WPF 攻撃に適した特徴量を評価する。第一に、WPF 攻撃に用いるデータセットを想定して、4 章で説明した 3093 種類の特徴量を評価し、WPF 攻撃に適した特徴量を選出する。第二に、選出した特徴量に対して、それらが持つ情報量が多い要因を、特徴量のばらつきと Web ページの構成の観点から分析する。

5.2 データセット

WPF 攻撃に有用な特徴量の評価と、WSF 攻撃時の特徴量が持つ情報量と比較をするために、WPF 攻撃と WSF 攻撃を想定した 2 つのデータセットを作成した。

本稿では、電子商取引サイトを WPF 攻撃の対象とし、世界的に広く利用されている一般的な電子商取引サイトとして Amazon を対象としたデータセットを作成する。本研究では、Amazon.co.jp を対象とする。攻撃対象とする Web ページは、売れ筋ランキングが規定されているカテゴリからランダムに 20 種類のカテゴリを抽出し、そのカテゴリの 2022 年 3 月 10 日時点の上位 5 商品から構成される全 100 ページを選択した。この Web ページに対して、1 ページあたり 100 回のアクセストレースを収集し、総計 10,000 のアクセストレースをデータセットとする。これを WPF データセットとする。

さらに、WSF 攻撃をする際の情報量と比較することを目的として、Alexa Top Sites の上位 10 サイトのトップページに対して、同様に 100 回のアクセストレースを収集し、総計 1,000 のアクセストレースを作成した。これに、上述の Amazon.co.jp の 1 ページのトレースを加えた計 1,100 トレースからなるデータセットを WSF データセットとする。

5.3 WPF 攻撃と WSF 攻撃時の情報量比較

はじめに、WPF 攻撃に有効な特徴量の候補を選択するために、WPF データセットを用いて 3093 種類の特徴量が有する情報量を評価する。さらに、WSF 攻撃時にそれぞれの特徴量が有する情報量と比較することで、それぞれの特徴量が WPF 攻撃に適している理由を分析する。

WPF データセットにおいて、各特徴量が持つ情報量を評価した結果を図 2a に示す。横軸は各特徴量に割り当てられたインデックスであり、縦軸は情報量である。青色のラインは、上位 100 位の情報量の値を示している。比較のため、図 2b に WSF データセットを用いたときの特徴量の情報量をプロットした。

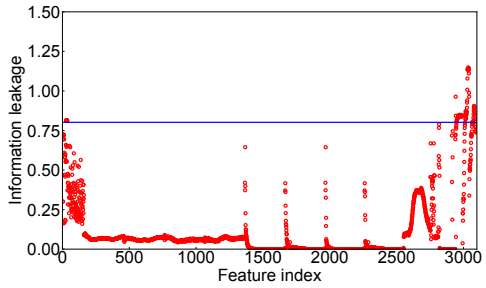
まず、全体の傾向として WPF 用のデータセットの方が情報量が少ない傾向にあることが分かる。この傾向が現れる理由として、WPF 攻撃時は分類する対象の Web ページが類似していることが挙げられる。具体的には、同一ドメインの Web ページは、同様のフレームワークあるいは同様のデザインであることが多く、異なる Web ページであっても特徴量が近い値を取るためである。そのため、特徴量から漏洩する情報量が低下していると考えられる。

カーネル密度推定から得られる各特徴量が取る値の確率密度関数を用いて、情報量が低下している理由を検証する。図 3 に、特徴量の 1 つである受信パケットサイズ累積和の第 45 番目の値の確率密度を示す。横軸は特徴量が取り得る値であり、縦軸は確率密度である。それぞれの線は、あるページを参照する際の特徴量の確率密度である。図から、同じ特徴量であっても、WSF 攻撃時よりも WPF 攻撃時の方がばらつきが小さいことがわかる。確率密度のばらつきについては、次章でより詳細に分析する。

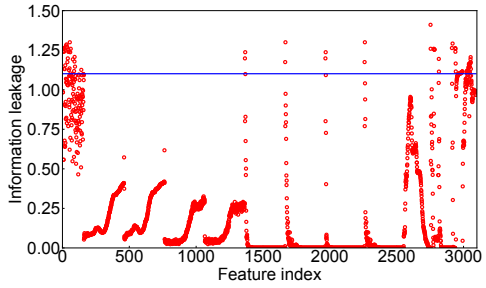
また、送受信のパケット数や秒間パケット数、パケットの転送方向などから得られる特徴量の多くは、WSF データセットでは 100 位以内に含まれていたのに対して、WPF データセットでは 100 位以下に低下している。これについても、同一ドメイン内の Web ページ群は、同様のタイミングで同様のファイルを要求することに起因していると考えられる。

5.4 WPF 攻撃時に情報量の多い特徴量

図 2a の結果で、WPF 攻撃に関して情報量の上位 100 位以内に入った特徴量は、パケットサイズ累積和、CDN バーストのサイズ、累積パケット転送間隔の 3 種類であった。これら 3 種類の特徴量が持つ情報量を図 4 にまとめる。横軸は、それぞれの特徴量を構成する 50 次元のベクトルの次元である。いずれの特徴量についても、後半の要素になるにつれて情報量が増加していることが分かる。これは、いずれの特徴量も、前半の要素は主に全てのページに共通の Web フレームワークなどのダウンロード時点での通信の特徴を反映しているため特徴量に差がなく、情報量が低

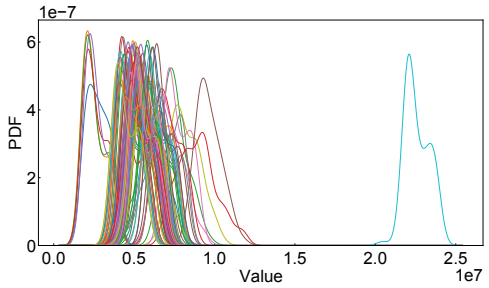


(a) WPF 攻撃時

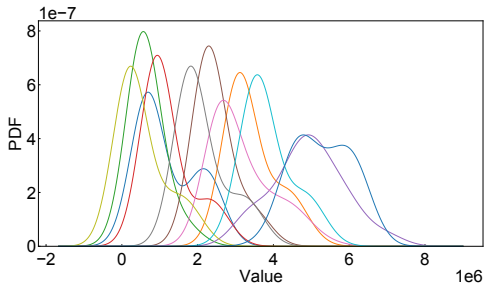


(b) WSF 攻撃時

図 2 特徴量の情報量



(a) WPF 攻撃時



(b) WSF 攻撃時

図 3 特徴量（受信パケットサイズ累積和の第 45 番目値）の確率密度関数

下している。

次章では、特徴量の値の分布の観点から、より詳細に分析する。

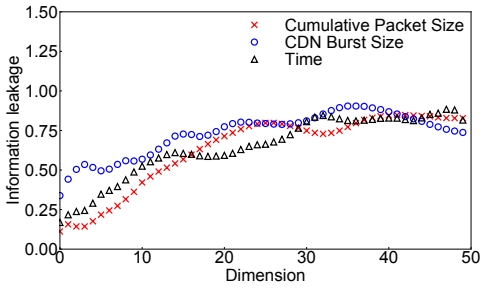


図 4 パケットサイズ累積和、CDN バーストサイズ、累積パケット転送間隔の情報量比較

5.5 特徴量の分布

本章では、特徴量の分布を評価することで、前章で示した 3 つの特徴量の情報量が多い要因を分析する。

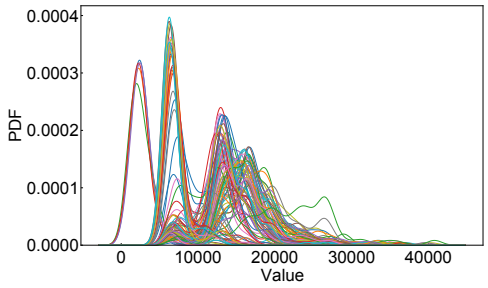
5.5.1 評価指標

主な解析手法として、図 3 で用いたのと同様のカーネル密度推定を用いて推定した特徴量の確率密度関数と、そのばらつきを定量化する指標として四分位偏差を用いる。

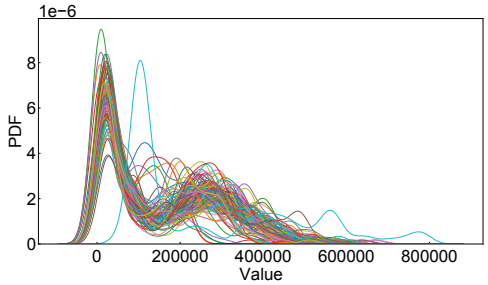
例として、図 2a の評価で情報量が最も多い特徴量と最も少ない特徴量について、カーネル密度推定から得られたそれらの特徴量の確率密度関数を図 5 に示す。1 つの線は、ある Web ページを参照するときのその特徴量の取り得る値についての確率密度関数を示しており、100 回のトレースで得た値に対してカーネル密度推定を適用して得られた結果である。図 5b から、情報量の少ない特徴量については、最頻値がおおよそ似た範囲に分布しているのに対して、情報量の多い特徴量は、最頻値がばらついている。

さらに、ばらつきを定量的に評価するために四分位偏差を用いる。具体的には、ある特徴量に対して、全ての Web ページの全てのトレースから得た特徴量に対して導出した四分位偏差と、各 Web ページの特徴量から導出した四分位偏差を全ての Web ページについて平均を取った値を用いる。図 5 で評価した 2 つの特徴量について、四分位偏差を導出した結果を表 2 に示す。

この結果を確認すると、情報量が最も少ない特徴量は、データ全体の四分位偏差が Web ページごとの四分位偏差の 1.1 倍程度であり差が小さい。これは、特徴量のページごとの分布とデータ全体の分布範囲に大きな差がないことを示しており、この特徴量を参照してもその値がどの Web ページへアクセスしたことで得られた値であるかを推定することが困難であるといえる。これに対し、情報量が最も多い特徴量では、データ全体の四分位偏差がページごとの四分位偏差の 2.5 倍以上であり、Web ページごとの分布範囲に対してデータ全体の分布範囲が広いことがわかる。これはアクセス先の Web ページが異なれば特徴量の分布範囲に差があり、特徴量を参照したときに、その値がどの Web ページへアクセスしたことで得られた値であるかを推定できる可能性が高いことを示している。



(a) 情報量が最大の特徴量



(b) 情報量が最小の特徴量

図 5 図 4 の評価で情報量が最多および最少の特徴量の特徴量の確率密度関数

表 2 図 4 の評価で情報量が最多および最少の特徴量の四分位偏差

	全体	ページごとの平均
情報量最多	4,613	1,495
情報量最少	119,630	117,382

次章以降では、WPF 攻撃に有益であるとした 3 種類の特徴量、パケットサイズ累積和、CDN バーストサイズ、および、累積パケット転送間隔に対して、確率密度関数と四分位偏差を用いて分析するとともに、そこで得られた特徴が、参照した Web ページのどの特徴から現れるものであるかを検討する。

5.5.2 パケットサイズ累積和

本章では、パケットサイズの累積和として、式 (2) 、式 (3) 、式 (4) で定義した、全パケットに関するパケットサイズの累積和 (S)、受信パケットに関するパケットサイズの累積和 (S_{in})、および、送信パケットに関するパケットサイズの累積和 (S_{out}) の 3 種類を評価する。それぞれのパケットサイズ累積和について、その累積和の推移から等間隔に 50 箇所の値をサンプルして得た 50 個の特徴量の情報量を図 6 に示す。S_{in} は累積和の推移の序盤から終盤までほぼ一定の情報量を示している。一方で、S と S_{out} は序盤の情報量は少ないが、終盤になり累積されたパケットの数が増えるにつれて情報量が増加する。

この差が生まれる要因を明らかにするために、特徴量の確率密度関数を調べる。ここでは、序盤と終盤の情報量の差が大きい送信パケットに関するパケットサイズの累積

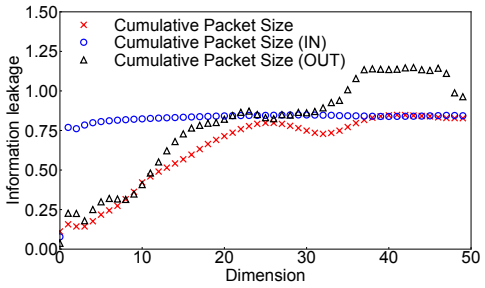
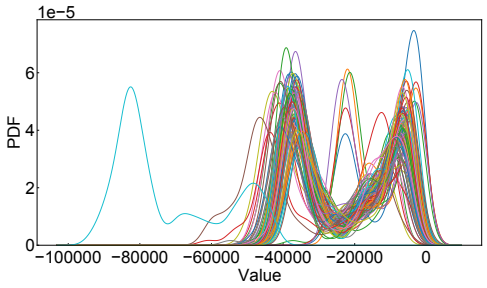
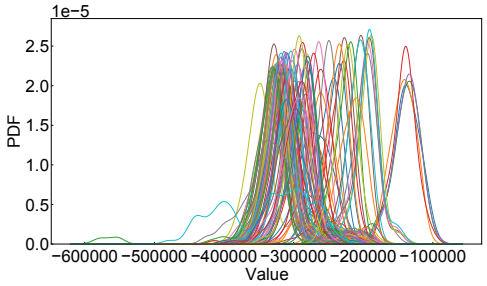


図 6 パケットサイズ累積和の情報量の比較



(a) 第 5 要素



(b) 第 45 要素

図 7 S_{out} の確率密度関数のの確率密度関数

和に注目し、その第 5 要素と第 45 要素の確率密度関数を図 7 に示す。図 7a では、多くの Web ページに対する確率密度関数の形状が類似していること、および確率密度関数の最大値が近い範囲に分布していることが分かる。この結果は、それぞれ、同一のページであっても特徴量が取る値が比較的 1 点に集中していないこと、複数のページが同じ特徴量で近い値を取る可能性が高いことを表しており、結果的に特徴量を参照したときにその値がどのページへのアクセスから得られたものであるかを特定することが難しいことを示している。これに対して図 7b では、ページごとの確率密度関数のピークが一点に集中しているとともに、ページごとに特徴量が取る値の最頻値にばらつきがある。これは、第 5 要素とは逆に、ページごとに第 45 要素の取る可能性が高い値に差があることを示している。つまり、特徴量を参照したときに、その値がどのページへのアクセスから得られた値であるかを特定できる可能性が高いことを示している。

表 3　S、S_{in}、S_{out} の第 5 要素の四分位偏差

	全体	ページごとの平均
S	390131	391066
S_{out}	13926	13489
S_{in}	78183	69086

表 4　S、S_{in}、S_{out} の第 45 要素の四分位偏差

	全体	ページごとの平均
S	849638	659776
S_{out}	32739	11351
S_{in}	775422	601351

次に、S、S_{in}、S_{out} の 3 種類のパケットサイズの累積和について、それぞれ第 5、第 45 要素の特徴量の四分位偏差を、5.5.1 章で議論したように評価する。結果を表 3 と表 4 にまとめる。表 3 から、S_{out} の第 5 要素については、ページごとの四分位偏差の平均とデータ全体に対する四分位偏差に、ほとんど差がないことがわかる。これはページごとの分布範囲とデータ全体の分布範囲に大きな差がなく、異なるページ同士がこの特徴量において近い値を取る確率が高いことを示している。逆に、表 4 から、S_{out} の第 45 要素については、ページごとの四分位偏差の平均と、データ全体に対する四分位偏差の間には 2.5 倍以上の差があることがわかる。これはページごとの分布に対してデータ全体の分布のばらつきが大きいことを示しており、異なるページへのアクセスから得られる特徴量は、異なる範囲の値を取る傾向にあることが確認できる。

S、S_{in} に対しても同様の比較を行うと、S では第 5 要素については、ページごとの四分位偏差とデータ全体の四分位偏差の差が小さいのに対して、第 45 要素は、その差が大きい。また、S_{in} は、S_{out} などと比較しても、第 5 および第 45 要素ともに、データ全体の四分位偏差がページごとの四分位偏差の 1.3 倍未満である。以上より、Amazon を対象として WPF 攻撃には、パケットサイズの累積和の中でも送信パケットに注目した特徴量を用いることが有効であることが分かる。

5.5.3　CDN バースト

既存研究 [10] において、ソーシャルネットワークを対象とした WPF 攻撃において、サイズの大きい CDN バーストの持つ情報量は多いことが指摘されている。これは、図 2b に示した WSF 攻撃のときも同様の傾向を示していた。しかし、図 2a や図 4 の評価で示したように、WPF 攻撃時は、サイズの大きい CDN バーストの情報量が小さい。この理由を明らかにするために、四分位偏差の評価に加え、Web ページの構成とブラウザによるファイルロードの内容を分析する。

まず、WPF 攻撃と WSF 攻撃時の CDN バーストサイズの最大値の四分位偏差を比較する。結果を図 5 に示す。この結果から、WSF 攻撃時には、最大 CDN バーストサイズは、データ全体の四分位偏差とページごとの四分位偏差に 3 倍以上の差があるのに対して、WPF 攻撃時にはその差が小さい。

表 5　最大 CDN バーストサイズの四分位偏差

	全体	ページごとの平均
WPF 攻撃	182,599	158,756
WSF 攻撃	635,772	185,118

表 6　累積パケット転送間隔の四分位偏差

	全体	ページごとの平均
第 5 要素	0.063	0.058
第 45 要素	0.544	0.284

続いて、この傾向が現れる要因を明らかにするために、本実験の WPF データセットから無作為に選んだ 5 種類のページに関して、ページを構成するファイルをブラウザ上で確認した。結果、5 ページ中 4 ページで、読み込まれたファイルのうちサイズの大きいファイルが同一であることを確認した。これらは、主に JavaScript ファイルであった。Amazon.co.jp の商品ページが全体的にサイズの小さいファイルで構成されており、ページ間で共通のファイルがサイズの大きいファイルを占めている。この傾向は、Wang ら [10] の Instagram などのソーシャルメディアを対象とした実験と異なる。Wang らの実験で対象としたソーシャルメディアでは、動画像などサイズの大きいファイルが各ページごとに異なっていたことから、サイズの大きい CDN バーストの情報量が大きく分類に寄与していた。本実験から、CDN バーストサイズの持つ情報は対象とするドメインによって異なることが分かった。本実験で用いた Amazon.co.jp の商品ページに関しては、WPF 攻撃時には、大きいサイズの CDN バーストの情報量が少なく、分類とって有益ではなく、ある程度サイズの小さい CDN バーストサイズを用いる必要があることが分かった。

5.5.4　累積パケット転送間隔

累積パケット転送間隔は、序盤の情報量は少なく、中盤以降にかけて情報量が増加する。まず、パケットサイズ累積和と同様に、累積和のインデックス第 5 要素と第 45 要素の特徴量について、その四分位偏差を表 6 にまとめる。情報量の少ない第 5 要素は、データ全体の四分位偏差とページごとの四分位偏差の差が小さいのに対して、第 45 要素はデータ全体の四分位偏差がページごとの四分位偏差の 1.9 倍以上である。

この傾向が現れる理由を明らかにするために、CDN バーストサイズの調査でも用いた無作為に選択した 5 種類の Web ページについて、序盤にロードされるファイルをブラウザ上で調査した。検証の結果、5 種類中 4 種類のページで、読み込まれるファイルのうち先頭 8 種類のファイルが

同一であることが分かった。これらは JavaScript や CSS、PNG ファイルであった。ここに含まれていた PNG ファイルは、主にアイコン画像などであり、複数のページで共通して用いられるファイルであった。なお、5 種類のページの中で例外であった 1 種類のページは Prime Video による映像作品のページであり、他の 4 種のページと異なるフレームワークによる Web ページであった。この結果から、同様のフレームワークからなる Web ページであれば、ページロードの初めの段階で同様のデータを読み込む可能性が高く、累積パケット転送間隔については、序盤にその値の差が小さくなることが分かる。

6. おわりに

本稿では、WPF 攻撃に適した特徴量を明らかにし、その傾向が現れる要因の調査を目的として、Amazon.co.jp を対象とした、特徴量の持つ情報量やページ参照時のパケットトレースの分析をした。具体的には、既存研究で扱われた 3093 種類の特徴量が持つ情報量を分析し、その後、パケットサイズ累積和、CDN バーストサイズ、および累積パケット転送間隔について、その確率密度と値のばらつきを評価した。Amazon.co.jp を対象とした WPF 攻撃に有用な特徴量として、パケットサイズ累積和、CDN バースト、累積パケット転送間隔の 3 種類があることが分かった。Wi-Fi を盗聴する攻撃者にとっては、パケットサイズ累積和と累積パケット転送間隔が有用であることが分かった。さらに、特徴量の確率密度と四分位偏差の観点、および、その特徴量と Web ページロードの関係を分析した。これらの検証の結果から、パケットサイズ累積和や累積パケット転送間隔は、累積されるパケット数の増加に伴い情報量も多くなる傾向にあることが明らかとなった。この傾向は、同一フレームワークからなる Web ページでは、ページロードの序盤で同様のファイルが読み込まれ、終盤につれて各ページごとに異なるファイルが読み込まれる。結果として、序盤の情報量が小さく終盤の情報量が大きくなる傾向がある。CDN バーストのサイズについては、WPF 攻撃時には、大きいサイズの CDN バーストから得られる情報量が少ないことが明らかとなった。Amazon.co.jp では、ページごとに異なる動画像などのファイルのサイズに対して、全ページに共通のページのフレームワークを規定する JavaScript などのファイルサイズが比較的大きいことが要因であった。これらの結果から、Amazon.co.jp を対象とした場合、パケットサイズ累積和と累積パケット転送間隔が有用であり、CDN バーストが利用できる環境であった場合は、ある程度サイズの小さい CDN バーストも含めて利用することが有用であることが分かった。

謝辞

本研究は、科研費 22K11994 によるものである。

参考文献

[1] T. Wang and I. Goldberg, “Comparing website fingerprinting attacks and defenses,” tech. rep., The University of Waterloo, 2013.

[2] B. Miller, L. Huang, A. D. Joseph, and J. D. Tygar, “I know why you went to the clinic: Risks and realization of HTTPS traffic analysis,” in *Proceedings of Privacy Enhancing Technologies Symposium*, July 2014.

[3] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Pennekamp, K. Wehrle, and T. Engel, “Website fingerprinting at internet scale,” in *Proceedings of Network and Distributed System Security Symposium*, 2016.

[4] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proceedings of USENIX Security Symposium*, 2004.

[5] J. Gong and T. Wang, “Zero-delay lightweight defenses against website fingerprinting,” in *Proceedings of USENIX Conference on Security Symposium*, 2020.

[6] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, “Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail,” in *Proceedings of IEEE Symposium on Security and Privacy*, 2012.

[7] M. Juarez, M. Imani, M. Perry, and C. Diaz, “Toward an efficient website fingerprinting defense,” in *Proceedings of European Symposium on Research in Computer Security*, 2016.

[8] T. Wang, H. Kong, and I. Goldberg, “Walkie-talkie: An efficient defense against passive website fingerprinting attacks,” in *Proceedings of USENIX Security Symposium*, 2017.

[9] P. Sirinam, M. Imani, M. Juarez, and M. Wright, “Deep fingerprinting: Undermining website fingerprinting defenses with deep learning,” in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[10] K. Wang, J. Zhang, G. Bai, R. Ko, and J. S. Dong, “It’s not just the site, it’s the contents: Intra-domain fingerprinting social media websites through CDN bursts,” in *Proceedings of International World Wide Web Conference*, 2021.

[11] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, “Breaking LTE on layer two,” in *Proceedings of IEEE Symposium on Security and Privacy*, May 2019.

[12] S. Li, H. Guo, and N. Hopper, “Measuring information leakage in website fingerprinting attacks and defenses,” in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2018.