ブラウザの特徴情報を用いたクロスブラウザのユーザ追跡手法

ヴー・スアン・ズオン† 碓井利宜† 重松邦彦‡ 武田圭史†

†慶應義塾大学環境情報学部
nobita@sfc.wide.ad.jp, alc@sfc.wide.ad.jp, keiji@sfc.keio.ac.jp
‡慶應義塾大学大学院政策・メディア研究科
sigematu@sfc.wide.ad.jp

あらまし Web ブラウザの高機能化に伴い, Web ブラウザが Web サーバにアクセスする際に, ユーザが意識することなく様々な情報が送信されている. 本研究では HTTP リクエ クトヘッダ情報等 Web ブラウザが送信する情報が持つ特徴をブラウザフィンガー プリントとして取得し, これらの情報の組み合わせを分析することで, ユーザが 複数のブラウザや機器を利用する場合でもそれらの関係性を把握してユーザのオ ンラインでの振る舞いを継続的に追跡する仕組みとその実装について解説する.

A Proposal of a Cross-browser User Tracking Method Using Distinct Browser's Information

Vu Xuan Duong† Toshinori Usui† Kunihiko Shigematu ‡ Keiji Takeda†

†Keio University - Faculty of Environments and Information Studies
nobita@sfc.wide.ad.jp, alc@sfc.wide.ad.jp, keiji@sfc.keio.ac.jp
‡Keio University - Graduate School of Media and Govermance
sigematu@sfc.wide.ad.jp

Abstract Modern web browsers and browser plug-ins provide a rich set of interfaces for many Rich Internet Applications. The other side, when web users access to the Internet, there are many browser's information sending upon each request without their awareness. This paper presents a method to tracking users who are using different browsers and devices at the same time by analyzing the relative correlation between finger-printed information. We also explain an implementation of a online user behavior tracking system used in this paper's experiment.

1 はじめに

インターネット上の広告社は, 様々な方法で Web ユーザの閲覧歴史ページや購買歴史などを集まっている. その一つの古典的な追跡手法として, Cookie を利用した方法が挙げられる. 特に Flash Cookie などの Super Cookie はブラウザが直接制御する以外に, ユーザが削除したり, 管理したりすろことは難しい. [4] では保存させた Cookie を削除することの難易度を調査し, Cookie を利用したユーザ追跡手法がかなり有効性があると明らかになった [5].

また, 近年 Twitter をはじめ, SNS(ソーシャルネットワークサービス) の普及によって, 個人情報を抜き取られる可能性が高くなっている.[3] では SNS に載せられている様々な個人情報と他の情報を組み合わせることで, 個人を特定出来る情報 (Personal identifiable information) を利

用し, オンライン上の振る舞いを追跡することができると述べている.

一方で, ブラウザの進化に伴い, Web サイトを見るだけのものではなく, プラグインを追加することで, 簡単に様々な機能を追加したり, 見かけを変えたりすることが出来るようになった. Web ユーザの中にはプライバシーの問題を意識しながら, Web サーフィンの際に Cookie を無効にしたり, SNS に個人情報を載せない人が徐々に増えている. しかし, ユーザが意識することなく, Web サーバにアクセスする際はプラグインの情報をはじめ, 様々な情報を送信されている. また, 個人によって, ブラウザの種類やバージョンや設定が異なっているため, 送信される情報も極めて異なる. したがって, そういったブラウザの情報を扱うことでユーザを識別したり, 追跡したりすることができると考えられる [1].

本研究では, Web サイトにアクセスする際に, ユーザが意識することなく発信している情報を収集する仕組みについて解説する. そして, これらの情報を組み合わせて解析することで, ユーザが 複数のブラウザや機器を利用する場合でもそれらの関係性を把握してオ ンライン上の振る舞いを追跡する手法を提案する.

2 関連研究

2.1 ブラウザフィンガープリント

How Unique is your browser の論文 [1] ではブラウザフィンガープリントのユニークさについて調査した. [1] によると, インターネットでブラウジングする際, OS やブラウザ, 使用しているブラグインなどの情報を取得し, ブラウザフィンガープリントとして, 8 割を超えたユーザを追跡可能である. [1] で実装した Web サイトを訪れた実験対象者の OS, ブラウザプラグインの設定, バージョンなどの情報を収集している. 結果としては, 全てのユーザのうち 84%がユニークフィンガープリントで識別可能だった. 更に Adobe Flash あるいは Java プラグインがインストールされている場合, 94%のフィンガープリントがユニークであった.

また, *Fingerprinting Information in JavaScript Implementations* の論文 [2] ではブラウザフィンガープリントを作成する手法を二つ挙げた. 一つ目は JavaScript の実行速度を測り, ブラウザのフィンガープリントを作成する. JavaScript の実行速度を測ることで, ほとんどのブラウザのバージョンや OS などを区別することができたという. 二つ目は Firefox の利用者に対して, NoScript Extension をインストールされている端末は NoScript の Whilelist をフィンガープリントとして扱われている. さらに, [2] で提案した手法を利用し, 利用者の端末の CPU Clock 速度や Cache Size などといった情報も把握できると述べた.

本研究では [1] の同じような Web サイトを開設し, ブラウザフィンガープリントのユニークさを再度解析する. 解析したことによって, ユーザが複数のブラウザや機器を利用した場合にもフィンガープリントの関係性の紐付ける手法を提案する.

2.2 ソーシャルネットワークでの個人特徴の情報を利用したユーザ追跡

B. Krishnamurthy と C.E. Wills. の論文 *On the leakage of personally identifiable information via online social networks*[3] では, ソーシャルネットワークを利用しているユーザは, ユーザの追跡できる情報をサードパーティーに漏洩されていると述べた. ユーザの追跡できる情報を取得することは簡単であり, ユーザのリクエストヘッダ情報の中に全て含まれている. そして, サードパーティーは Request-URIs や Referer ヘッダや Cookie などを収集することで, ユーザをプロファイリングして, そのソーシャルネットワーク内だけでなく, 複数のソーシャルネットワークを利用している人もリンクすることができる. [3] では, ユーザの意図しないことでオンライン振る舞いを追跡されてしまう危険性についても述べた.

3　ブラウザフィンガープリントを利用したユーザ追跡手法

3.1　情報収集プログラムの概要

本研究の目的はブラウザフィンガープリントを利用することで, ユーザが複数のブラウザや機器を利用した場合にユーザのオンライン振る舞いを追跡することは目的である. そのため, フィンガープリントを作成する際に, 必要となる情報を取得しなければならない. 第 ??節では, フィンガープリントを作成することにあったって, 必要となる情報を取得しるための仕組みについて述べていく. 図 1 はこの仕組みの設計図を表す.

研究室の Web サーバ上にブラウザの情報を収集するプログラムを起動し, AJAX を利用するこで, 収集した情報をサーバ側に送信する. ユーザが Web サーバにアクセスすると自動的に情報を送信し, それらの情報をデータベースに挿入する. フィンガープリントの作成や解析をする際はデータベースに保存された情報を利用する.

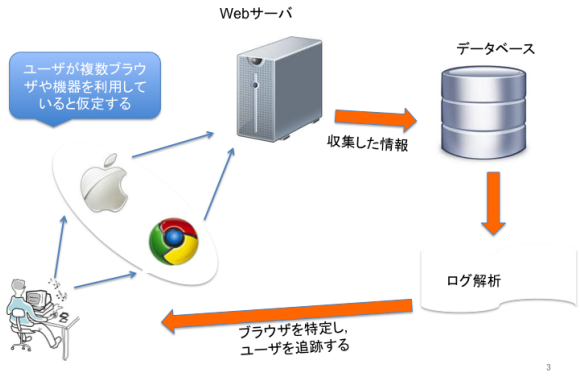

図 1: 情報収集仕組みの実装

具体的に取得する情報は以下の情報になっている.

- HTTP リクエストヘッダの中に含まれている情報
  HTTP リクエストはブラウザが送るもので, クライアントとサーバはこれを利用してデータやソフトウェア自身の情報やり取りしている. HTTP リクエストの中身は様々な情報を含まれているが, 本研究ではブラウザの種類や OS の情報などを含んだ User-Agent と HTTP ACCEPT ヘッダを利用する.
- インストールされているプラグインの名前またはバージョン
  プラグインとはブラウザの拡張機能であり, Internet Explorer を除き, 現在は 4 つの主なブラウザ (Chrome, Firefox, Safari, Opera) は全てプラグインが追加する機能をサポートしている. ブラウザフィンガープリントにとって, プラグインの情報が極めて重要な役割を果たしている.
- インストールされているフォント
  Flash applet または Java applet を利用することで, ホストでインストールされているフォントを取得する.
- アクセスしてきたホストの IP アドレス
- 端末の画面のサイズ

3.2　フィンガープリントの定義

Variable	Value
User Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_1) (省略)
Plugin	QuickTime Plug-in 7.7.1 (省略)
HTTP ACCEPT ヘッダ	text/html, application/xml; q=0.9, (省略)
System font	Abadi MT Condensed Light, (省略)
端末の画面サイズ	1440x900

表 1: ブラウザフィンガープリントの例

フィンガープリントとは指紋という意味で, 一般にはデジタル証明書などが改竄されていないことを証明するデータを指す. 本研究におけるフィンガープリントとは Web サーバに残っているブラウザの情報やその情報の組み合わせのことであり, ブラウザを識別したり, ユーザを追跡したりするために利用される.

第 3.1 節に挙げられたものの組み合わせることで, 本研究におけるブラウザフィンガープリントと言う. 表 3.2 ではブラウザフィンガープリントの一つの例とする.

3.3 ブラウザを紐付ける条件

本稿では, ブラウザフィンガープリントを解析することで, ユーザが複数のブラウザや機器を利用した場合にはブラウザの関係性を把握し, ユーザのオンライン振る舞いを追跡できることを目的とする. つまり, フィンガープリントを利用することで, ブラウザの関係性を把握しなければならない. これを踏まえ, 本稿ではブラウザの関係性を結ぶ付く手法について提案する.

第 1 節では述べた通り, ブラウザの進化に伴い, ブラウザを利用し, ただ Web サイトを見ることだけではなく, 簡単にブラウザのプラグインを追加することで, 様々な機能を拡張することができる. そして, 個人によって, 利用しているプラグインが異なっているため, ユーザを追跡するのに極めて重要な役割を果たすと考えられる. [1] ではプラグインと Font と User Agent は順番の三つのが Entropy bit の値が一番高いと述べた. つまり, ユーザを識別フィンガープリントに重要な三つの情報のことであった.

次にフィンガープリントを結び付く条件について述べる. 同じユーザであるが, 複数のブラウザや複数の機器を利用した場合にはフィンガープリントを異なっている. 以下の条件を一つ満たした場合は, フィンガープリントの関係性を作成する.

1. ブラウザ間
   一つの機器上で複数のブラウザを利用されている場合は, 二つのフィンガープリントの Font をマッチングした場合に同じ機器上での二つのブラウザだと判定する.

2. 機器間
   NAT をしているネットワークでは複数の機器を利用した場合にも関わらず, Web サーバにアクセスする際には同じ IP アドレスが出てくる. そのため, ユーザが複数の機器を利用した場合には IP アドレスを基にブラウザを紐付けると考える.

   それに加え, Web サイトアクセスするユーザは同じ Web ページを閲覧し, ユーザのアクセスパータンを解析することで, 違う端末でも, ある程度同じ Web ページを閲覧する場合にブラウザフィンガープリントを紐づけられると考える. しかし, 本稿における実験はユーザを募集した形で, 自然的なアクセスではなかったため, アクセスパータンの解析と実験をすることが今後の課題とする.

4 実験

提案した手法の有効性を検証するため, 各ユーザのフィンガープリントの違いの検知と紐付けるフィンガープリントについて実験した. 本研究の目的とする検証はブラウザのフィンガープリントを基にユーザを追跡できなければならない. 実験するために, 実験対象者のブラウザに Cookie を保存させる. 同じユーザは同じブラウザを利用すれば, Cookie の値は同じであることで, ユーザのフィンガープリントを比較する. 本稿ではブラウザの通常モードと Private Browsing モードにおいて実験した.

4.1 実験方法

研究室のサーバに設定した Web サイトをアクセスする実験者を募集し, 実験を行ってきた. 実験者のネットワークの環境は全員家のネットワークの環境であり, NAT しているネットワークであることを確認した以上に実験した. そして, 実験を受けた実験者の IP アドレスを確認した. これで, 実験者のアクセスしてきた端末の ID をマッチングすることができ, 第 3.3 節で提案した手法の有効性について検証した.

4.2 実験結果

4.2.1 ブラウザの通常モードによる実験

まず , 実験者が同じブラウザを利用する際に , 2 回以上のアクセスがあると , フィンガープリントが同じであることを確認した . 実験者の端末をに Cookie を保存させることで, 同じブラウザを利用した場合 Cookie の値を同じでなければならない. 結果は表 4.2.1 で示している. フィンガープリントが同じであるが, 保存させた Cookie の値が違うのは 1 件. これは iPhone の Opera Mini の最新版で, 恐らく違う携帯端末のからのアクセスであったが, 端末の OS バージョンやブラウザの設定が全く同じであるからだと考えられる.

次に提案手法を用いてフィンガープリントセットを作成する. フィンガープリントセットは提案した手法のルールを応用することで, 二つのフィンガープリントの関係性がある場合に同じセットに挿入する. 表 4.2.1 で表した結果は 42 個の違うフィンガープリントから 19 個のフィンガープリントセットに減少した.

実験対象者数	15
アクセスしてきた数	107
保存させた Cookie の値の数	53
フィンガープリントの数（条件を結び付く前)	42
フィンガープリント同じであるが, Cookie 値が違う (Private Browsing モードを除く)	1
条件を結びついた後のフィンガープリントセット	19

表 2: 実験結果

4.2.2 Private Browsing モードによる実験

現在多くのブラウザベンダーがPrivate Browsing モードを提供しており, ブラウザによって, ポリシーが違うが, Private Browsing モードでは, Private Browsing セークショーン中 Cookie を保存するが, セークショーンを終了させると Cookie を削除されてしまうという特徴がある. 本実験では, Chrome, Firefox, Internet Explorer, Opera, Safari といった 5 つ主なブラウザの Private Browsing モードを調査し, 調査した結果は以下に述べる. 調査対象は通常モードと Private Browsing モードに対して, ブラウザを特定するための情報, つまりブラウザフィンガープリントはどのような違いが出ているかについて調査する. 表 4.2.2 では Firefox と Chrome の Private Browsing の比較表である.

各ブラウザは Private Browsing モードで 5 回 Web サーバにアクセスする. 結果としては Private モードでは通常の通り, フィンガープリントを取得することができる. Internet Explorer は Plugin の機能をサポートしていないため, 通常のモードと Private Browsing モードでも両方 Plugin リストを取得することができないが, Chrome 以外に残っているブラウザのフィンガープリントの変化がなかった. Chrome の場合は Private Browsing モードでは 5 回のアクセスの中に 1 回取得したプラグインリストの順番が変わった.

	User Agent		Plugin		Font	
	Normal	Private	Normal	Private	Normal	Private
Firefox	○	○	○	○	○	○
Chrome	○	○	○	△	○	○
Opera	○	○	○	○	○	○
Safari	○	○	○	○	○	○
IE	○	○	×	×	○	○

表 3: 通常モードと Private Browsing モードの比較

5 考察と今後の課題

第 4 節の結果は, 提案方法の有効性を示している. それぞれのアクセスに対して, ブラウザフィンガープリントを作成することで, 同じユーザが同じブラウザを利用する場合を発見することができた. また, ユーザが二つ以上のブラウザを持っている場合にも, フィンガープリントの関係性を発見することにある程度成功した. 誤検知を発生した場合にその原因についても解説した. ユーザが Web サーバにアクセスする際に意識することなく送信されている情報を利用ることで, ブラウザの関係性を把握することができ, オンラインユーザの振る舞いを追跡できることが分かった.

しかし, 本稿で提案した手法と検証に用いた実験は単純であり, 実験人数と実験規模も限定

であった. また提案した方法では誤検知を発生しており, 次の問題点が存在している.

- 情報収集するプログラムは JavaScript を多く利用しているため, JavaScript を無効する場合に情報を収集することができなくなる.
- 複数機器を利用しているユーザはブラウザフィンガープリントを紐付けることが難しくあり, IP アドレスを用いた提案手法では IP アドレスを変わる時に紐付けることが困難である. それに同じ NAT の環境でも複数のユーザがいる場合に検知することができない.
- Private Browsing モードにも関わらず, ブラウザを新しいバージョンに更新した際など, ブラウザフィンガープリントの変更に対する問題点がある.

今後そのような問題点を踏まえ, 実験規模を広めつつ, 更に有効な方法を考えなければならない. そして, ブラウザフィンガープリントを用いたユーザ追跡手法の危険性について検討する必要がある.

6 まとめ

本稿では, Web サーバにアクセスする際にユーザが意識することなく送信されている情報の取得と解析をすることで, ブラウザのフィンガープリントを作成し, それらの情報を用いたユーザ追跡手法を提案した. ユーザが複数のブラウザや機器を利用した場合にある程度発見できることを実験で検証した. 今後更に有効性がある手法を検討しなければならない.

参考文献

[1] Peter E. *How unique is your browser*. Electronic Frontier Foundation (May 2001). http://panopticlick.eff.org/browser-uniqueness.pdf

[2] Keaton M., Dilon B., Scott Y.,& Hovav S. Fingerprinting Information in JavaScript Implementations. In *Proceedings of W2SP 2011*. IEEE Computer Society, May 2011.

[3] B. Krishnamurthy and C.E. Wills. On the leakage of personally identifiable information via online social networks. In *Proccedings of the 2nd ACM workshop on Online social networks*, pages 7-12, 2009.

[4] K. McKinley. Cleaning up after cookies. iSec Partners White Paper(2008). Retrieved August 25th, 2011. http://www.isecpartners.com/white-papers/2010/7/22/cleaning-up-after-cookies.html

[5] Seth S. *New cookie technologies: Harder to see and remove, widely used to track you*. September 14, 2009. Retrieved July 13, 2011. https://www.eff.org/deeplinks/2009/09/new-cookie-technologies-harder-see-and-remove-wide

[6] Wikipedia. *TCP/IP stack fingerprinting*. Retrieved July 13, 2011 http://en.wikipedia.org/wiki/TCP/IP_stack_fingerprinting

[7] Firewall & Forest. やさしいセキュリティ(トラッキング Cookie). Retrieved July 6, 2011. http://eazyfox.homelinux.org/security/beginner/beginner08.html/