Web Browser Fingerprint 技術の現状と課題

齋藤孝道† 高須 航‡ 山田智隆‡ 武居直樹‡ 石川貴之‡ 細井理央‡ 安田昂樹† 高橋和司†
†明治大学，‡明治大学大学院

あらまし Webのターゲティング広告などの目的のため，Web閲覧者を識別する技術が広く利用されている．識別技術の一つに，Web Browser Fingerprint技術がある．Webサーバへのアクセスに付随して採取できるブラウザの特徴的な情報を用いてブラウザや端末を特定する技術である．2010年に実施されたPanopticlick Projectを発端として，現在まで数多くの研究が発表されてきた．本論文では，これらの研究を踏まえ，Web Browser Fingerprint技術の現状と，この技術をリスクベース認証などセキュリティ技術として利用する際の課題について考察する．

Current Status and Issues of Web Browser Fingerprinting

Takamichi SAITO† Ko TAKASU‡ Tomotaka YAMADA‡ Naoki TAKEI‡
Takayuki ISHIKAWA‡ Rio HOSOI‡ Kouki YASUDA† Kazushi TAKAHASHI†

†Meiji University, ‡Graduate School of Meiji University

Abstract For the purpose of targeted Web advertising, it is popular to provide customized information or advertisements to an identified Web viewer. One of the promising identifying methods is Web browser fingerprinting. The Web browser fingerprinting is defined as the process of an observer uniquely identifying a device or browser based on multiple information elements communicated to Web server. As it is originated by Panopticlick Project in 2010, many fingerprinting researches have been conducted. In the paper, we study current status and issues of Web browser fingerprinting from the viewpoint of security technology.

1 はじめに

今日の商業用の Web サイトでは，Web サイト閲覧者の端末の識別を行う手段の一つである Web Browser Fingerprint（以降，Fingerprint という）を用いて，閲覧履歴などによる行動追跡を行い，閲覧者の嗜好に基づく情報を提示することが普及している．実際，文献[1]によると，Alexa[2]の提供する Web サイトアクセス数トップ 10 万サイトの 5.5%が，代表的技術の一つである Canvas Fingerprint を利用しているとされている．

Fingerprint 技術は，Web サーバへのアクセスに付随して採取できる特徴的な情報を用いて閲覧者のブラウザや端末の特定を行うので，HTTP クッキーを用いた手法が機能しない場合においても有効である．HTTP クッキーの利用に対する EU の規制（Cookie Law[14]，2012 年）の施行により，行動追跡を目的とした行為を閲覧者の同意無しに行うことが禁止されるなど HTTP クッキーを取り巻く状況は厳しくなっている．そのような背景もあり，Fingerprint 技術は，さらなる技術的な発展が期待されている．

Fingerprint を用いた識別の研究の起源に，2010 年，Electronic Frontier Foundation（EFF）により実施された Panopticlick Project[3]がある．その後，様々な研究が発表され課題も見えてきた．本論文では既存研究を中心に Fingerprint 技術の現状と，この技術をリスクベース認証などのセキュリティ技術として利用する際の課題について考察する．

本論文での新たな成果としては，特徴点を複数組み合せた Fingerprint の議論に加え，モバイルデバイスにおける端末の識別力について示した点である．Eckersley[4]は，iPhone や Android などのモバイルデバイスにおいてブラウザから採取できる Fingerprint は均一であり，PC 端末と比べ Fingerprint となる

情報がないとした[4]が，我々の実験ではモバイルデバイスにおいても PC 端末と同様の精度で識別できる可能性があることが分かった．また，Fingerprint 技術の応用として，閲覧者の行動追跡やリスクベース認証に留まらず，マルウエアの不正通信活動の検知での利用など様々な応用の可能性を示す．

2 Fingerprint について

2.1 Fingerprint の定義

ブラウザが Web サーバへアクセスした際，Web サーバが取得できるブラウザや端末に関する情報を特徴点と呼ぶ．本論文では，特徴点を 1 つ以上組み合わせたもの及びそのハッシュ値を Fingerprint と呼ぶ．それを採取する行為を，Web Browser Fingerprinting（以降，Fingerprinting と呼ぶ）と呼ぶ．ただし，この際，閲覧者の自発的な操作(API 利用の許可など)は必要とせず，ソフトウエアの脆弱性を利用することもない．

Fingerprint による識別は，閲覧者の使用する端末上のブラウザ（以降，クライアントという）を対象とする．また，同一端末であってもブラウザを変更すると，同一と見なせないことがある．ただし，後述のハードウエア特徴点などのブラウザに依存しない特徴点により，ブラウザを跨いだ識別も可能である．

2.2 Fingerprint 技術の歴史

2002 年，ブラウザでのページの表示の際，ハイパーリンクの色が，訪問済みか否かによって異なることを利用した，Cascading Style Sheets（CSS）によるブラウザの閲覧履歴の特定をする history stealing 攻撃が提唱された[10]．ただし，2010 年 4 月，Firefox4 ではパッチが当てられ，現在では多くのブラウザで実行不可となっている．

同年，Miller[11]は TCP の SYN パケットを用いて OS の種類の特定を行う OS Fingerprinting を提唱した．

2008 年，Fingerprinting サイトの一つである Browserrecon Project[12]では，HTTP リクエストヘッダを用いてブラウザのバージョンと種類の特定を行った．Fingerprint 技術に関する最初の研究だと推測される．

2010 年，Eckersley[4]は，Panopticlick[3]サイトにおいて HTTP ヘッダ，JavaScript や Flash を用いて特徴点の採取を行った．収集したサンプルのうちの 94.2%が瞬間的にユニークであることを示した．Fingerprint を用いた識別の研究の原点である．

2012 年，Mowery ら[13]は，HTML5 の Canvas 要素を用いて文字列や図形を描画した際の結果の違いにより，ブラウザ，OS，GPU を識別する Canvas Fingerprinting を提唱した．

2013 年, Nikiforakis ら[15]は, BlueCava, Iovation 及び ThreatMetrix の広告事業 3 社と，Panopticlick の Fingerprinting の手法の違いにより，Fingerprint の分析を行った．また，JavaScript プロパティとブラウザベンダー毎に固有なプロパティの実装状況からブラウザの種類とバージョンを識別する手法も提案している．

2.3 Fingerprint 技術の利用例

Fingerprint 技術はいくつかの分野で応用されている．最もポピュラーなのは広告業界での利用である．その用途は行動ターゲティング型広告に必要な行動追跡や，成功報酬型広告の不正対策がある[6][7]．

注目を集めつつある応用として，リスクベース認証がある．OpenAM では，Fingerprint をリスクベース認証として用いることでなりすましを防いでいる[8]．

今後，期待される用途として，サイバー犯罪における証拠の採取法としての利用が挙げられる．論文[9]では，Fingerprint 技術を発展させ，ペネトレーションツールとしての応用が示された．ゲーム業界では，ソフトウエ

アで利用者の Universally Unique Identifier（UUID）により不正利用者の特定をするが，Fingerprint でも同様な特定ができる．さらに，Fingerprint 技術を利用した Bot 検知機構も商品化されている[37]．この応用として，本論文では，Fingerprint 技術を用いて，標的型マルウエアの通信の検知への応用の可能性があることをここに示す．これは，その種のマルウエアの通信では UserAgent 値が適切でない場合があり[40]，ブラウザによる通信の Fingerprint の違いにより判断する．

3 Fingerprint 技術の現状

3.1 Fingerprint/Fingerprinting の分類

Fingerprint の種類は，ソフトウエア，ネットワーク，ハードウエアの 3 つに分類される[5]．ソフトウエアは，OS やブラウザに関連する特徴点を指す．ネットワークは，閲覧者から送信される情報や閲覧者の属するネットワークセグメント内の情報などの特徴点を指す．ハードウエアは，CPU やカメラなど，端末自身や付随するデバイス情報を指す．

Fingerprinting の手法には，Web サーバへのアクセスに付随する情報を採取する Passive Fingerprinting と，ブラウザ上で JavaScript などのスクリプトにより採取する Active Fingerprinting の二つに分類される．なお，本論文では一部を除き HTTP ヘッダ及び JavaScript での採取を対象とする．

3.1.1 Passive Fingerprinting

この手法では，Web サーバは，ブラウザから送信される TCP/IP ヘッダと HTTP ヘッダから Fingerprint を採取する．例として，HTTP ヘッダに含まれる UserAgent 文字列を以下に示す．

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:40.0) Gecko/20100101 Firefox/40.0
```

この例では，”Mozilla/5.0”によりブラウザが Mozilla 互換であることが分かる．現在の全てのブラウザで共通して提示される．続く括弧内の情報は，プラットフォーム名，及び，レンダリングエンジンのバージョンを示す．”Gecko/20100101”は，ブラウザが Gecko ベースであることを示す．”Firefox/40.0”は，ブラウザの種類とバージョン番号を示す．

3.1.2 Active Fingerprinting

この手法では，Fingerprinting に必要なスクリプトを含む Web ページを閲覧者へ送信する．スクリプトとしては，JavaScript を用いることが最もポピュラーだが，その他にも Flash や CSS などを用いることもある．

JavaScript を用いた採取方法としては主に 2 つある．1 つはプロパティの値を採取するものである．プロパティの値はブラウザが保持するので，採取に時間がかからない点が特徴である．もう 1 つは，ベンチマーク計算などの処理により，その結果を採取するものである．処理によっては時間がかかるので，処理中に閲覧者が他のページへ移動する場合，採取することができないことがある．

3.1.3 CSS のみによる Fingerprinting

スクリプトによる採取法の他に，CSS のみでの採取法がある．ブラウザのレンダリングエンジンにおける CSS プロパティの対応状況の違いを利用したブラウザの判別，Media Queries を用いた端末の画面に関する情報，@font-face を用いたフォントリストの採取法がある[20]．これらの手法は JavaScript が機能しない状況においても有効である．

3.2 採取可能な特徴点

Fingerprinting を行う Web サイト [3] [21] [22]の多くは，UserAgent，Accept ヘッダ，Accept Encoding，Accept Language，

プラグイン，タイムゾーン，画面解像度，フォントリスト，HTTP クッキーの利用可否，Session Storage の利用可否，Local Storage の利用可否，Do Not Track(DNT)宣言情報，Canvas Fingerprint などを採取している．

SSE2，物理 CPU コア数，ハードディスク空き容量，リフレッシュレートの採取法も提案されている[16][17].

CSS のみでしか採取できない情報として，Firefox に限定されるが，Mac OS でのアピアランスの色，Desktop Windows Manager を使用しているか，Windows OS におけるテーマがクラシックモードであるか，Windows の既定のテーマを使用しているかという情報がある[20].

閲覧者の端末の情報だけでなく，閲覧者の端末が属するネットワークに関する情報も採取することが可能である[9]．また，端末にインストールされた一部のアプリケーションの推定も可能である[39].

3.3 類似端末での Fingerprinting

ハードウエア，OS 及び設定が同一の端末（以降，類似端末と呼ぶ）での Fingerprinting においては，ソフトウエア及びハードウエアの特徴点の差は実質的に無い．個人利用の場合，使用するブラウザや端末の設置場所によるが，オフィスや学校などでは利用環境が均一なので，特徴点の差は無いとされる[38].

しかしながら，グローバル・プライベート IP アドレスを用いての識別が可能となる．特に，NAPT 環境下にある端末の場合，プライベート IP アドレスによる識別は有効である．

3.4 対策技術

本節では，Fingerprinting の対策技術について説明する．現状で，確実に効果のある対策は存在せず，今後の研究に期待されている．

ブラウザでの JavaScript の無効化が単純な対策の一つとしてあるが，現在の Web サイトにおける JavaScript の使用率は約 90%なので[24]，現実的な対策ではない．

なお，ここでの対策とは主に PC 端末における対策であり，モバイルデバイスにおいても有効とは限らない．

3.4.1 ブラックリストを用いた対策

AdBlockPlus[25]は，ブラック・ホワイトリストに設定されている URI を元に HTTP 通信の制御を行う，ブラウザの拡張機能である．デフォルトで用意されているブラック・ホワイトリストの設定に加え，Web 上からダウンロードした設定を適用することもできる．現状で特定のサイトのみで Fingerprinting が行われているので，それらのサイトをブロックするように設定することで Canvas Fingerprinting への対策になることが示されている[1].

Ghostery[26]も，同様のブラウザの拡張機能である．追跡を行う 2,000 以上の企業リストがブラックリストに含まれている．

3.4.2 Fingerprint の書換えによる対策

FireGloves[27]は，Firefox の拡張機能で，閲覧者の端末の Fingerprint の値を任意の値に書き換えることで，識別されることを防ぐ．しかし，端末が持ち得ない値（例えば画面解像度が 1920*1080 の iPhone など）を生成する場合には，それがさらなる Fingerprint になってしまうことが示されている[16].

Privaricator[28]は，Google Chromium に実装された対策技術である．プラグインとフォントリストの値をランダムな値にすることで Fingerprinting の対策を行う．

Chameleon[29]は，Fingerprinting の検知，及び，書換えによる対策を行う Google Chrome 用拡張機能である．閲覧者の端末の Fingerprint を Tor Browser の Fingerprint に書き換えることで対策を行っている．

3.4.3 Tor による対策

Tor Browser[30]は，通信経路の匿名化を実現する．それに加え，Fingerprinting 対策としてプラグインやフォントリストが採取されないようになっている．また，Canvas Fingerprinting 対策として，HTML5 の Canvas API を用いた画像データへのアクセスの際にダイアログを表示する機能もある．Tor Browser にプラグインを追加することは可能であるが，デフォルトの Tor が生成する Fingerprint とは異なるものになるので，デフォルトの利用が望ましいとされる[31].

3.4.4 その他の対策

DNT[32]は，行動追跡の可否要求を Web サイトに通知するモダンブラウザの標準機能である．しかし，その可否要求に強制力はなく，効果がない[33].

プライベートブラウジングは，履歴を残さずに閲覧を可能にする機能である．フォントリストの採取を妨げるが，効果がない[34].

4 識別に関する議論

4.1 識別力・識別精度に関する考え方

Fingerprint の識別力はエントロピーにより示されることが多い．分散が多く発生頻度が一様であると，エントロピー値が高くなるので，その場合に高い識別力をもつ．4.2 節で詳しく説明する．

しかしながら，文献[36]によれば，エントロピー値が高い Fingerprint であっても，時間経過と共に変化することが示されており，単純に，エントロピー値の高低のみで判断できない．4.3 節で詳しく説明する．

Fingerprint の識別力とは別に，クライアントを追跡し続けるには，変化する特徴を同一とみなす必要がある．文献[18]では，採取の間隔が 4 週間以上でも，プラグイン情報を文字列としたときの距離(類似度)の利用のみで，97.57%の精度で識別できることが示された．また，Fuzzy Hashing を用いた手法では，複数の特徴点を用いて，2 ヶ月以内で 97.5%，また，2 ヶ月以上間隔が空いた場合 85%の精度で識別できることが示された[19].

4.2 エントロピー

著者らの運用する Fingerprint 収集サイト[23]で収集したサンプルから，各特徴点のエントロピーを算出した（表 1）．なお，ブラウザの識別には，HTTP クッキー（以降，UID と呼ぶ）を用いた．2013/12/6 から 2015/6/20 までに採取したサンプルを用いており，サンプル数は 3,536，UID 数は 1,513 である．重複を防ぐために特徴点が一致する場合 1 つと数えた．よって，計算用サンプル数は 2,104 となる．20_fingerprints は#1~20 までの 20 個を一つの特徴点とみなす Fingerprint，15_fingerprints は 20_fingerprints から，後述の変化しやすい特徴点（#1,2,3,5,6）を除いた Fingerprint である．

表 1. 各特徴点のエントロピー

No.	特徴点	エントロピー
1	プラグイン	8.760177532
2	グローバル IP アドレス	8.123545301
3	JavaScript UserAgent	8.052320057
4	フォントリスト	7.595955103
5	HTTP UserAgent	7.549623036
6	プライベート IP アドレス	7.294615906
7	画面解像度	4.238163875
8	SSE2	3.257869287
9	HTTP Accept-Language	3.013280404
10	HTTP Accept	2.244344739
11	HTTP Origin	1.886926683
12	HTTP Connection	1.857771589
13	HTTP Referer	1.687704693
14	Device Pixel Ratio	1.662121644
15	HTTP Accept-Encoding	1.346142395
16	Touch	0.445354607
17	タイムゾーン	0.391412209
18	HTTP Accept-Charset	0.142117209
19	Local Storage	0.124951728

20	Session Storage	0.10377298
21	20_fingerprints	10.9780639
22	17_fingerprints	10.21384421
23	15_fingerprints	9.70306589
	理論値	11.0389189

この結果は，類似研究における結果と同様である．また，重要な結果として，プラグイン単体よりも組み合わせると，Fingerprint のエントロピーが高くなることが分かる．

4.3　時間経過に伴う特徴点の変化

文献[36]では，比較する二つの同一 UID において，採取期間の差が 1 日未満であれば，高い一致率になるが，差が大きくなると一致率が低下することがあることが示された．特に，グローバル／プライベート IP アドレスは 1 日以上経過すると半分程度が変化し，プラグインは 14 日以上経過すると 80%以上が変化，UserAgent は 21 日以上経過すると約 40%が変化したことが示された．長期的な識別には，時間経過に伴い変化しにくい特徴点を用いるのが有効であるので，エントロピーが高い特徴点が長期的な識別において好ましい特徴点とは限らないこととされている．

4.4　モバイルデバイスの Fingerprint

著者らの運用する収集サイト[23]から，モバイルデバイスのサンプルを取り出し，モバイルデバイスにおける特徴点のユニーク率を求める．ここで，モバイルデバイスとは，UserAgent に iPhone または Android と表記されているサンプルとした．また，2014/1/20 から 2015/6/17 までに採取し，同一 UID は除いた 116 個をサンプルとした．

収集したサンプルにおける各特徴点のユニーク率を，ユニーク数の高い順に並べ，表 2 に示す．モバイルデバイスを含めた全ての端末でのユニーク率も併記した．

各特徴点のユニーク率とは，（各特徴点のユニーク数）/（全体のサンプル数）とした．また，17_fingerprints は 20_fingerprints からグローバル・プライベート IP アドレス，プラグインを除いたものである．all はサンプル数を示す．なお，フォントリストはソートされない時期のものも含むので，全ての端末の結果にはフォントリスト（no-sorted）を追加した．

表 2 で，モバイルデバイスのグローバル IP アドレスでは約 70%となり，UserAgent はいずれも約 60%がユニークな値を示した．その他はユニーク数が少ないことが分かる．特に，フォントリスト，プラグインのユニーク数は全ての端末の場合と比べ少ない．これは，モバイルデバイスでは Flash による採取ができないケースや，フォント，及び，プラグインを追加する仕組みがないことが考えられる．しかし，20_fingerprints と 17_fingerprints では，ユニーク率が高い値となり，全ての端末を含む場合のユニーク率よりも高い値を示している．このことから，モバイルデバイスではユニークではない特徴点が多くあるが，特徴点を組み合わせることにより，ユニーク率が高くなる可能性がある．

4.5　Fingerprint 技術の課題

Fingerprint による識別の課題としては，エントロピー値が高く，時間経過と共に変化しにくく，様々な端末で採取可能な特徴点を数多く見つけることがある．また，変化する Fingerprint を同一とみなす方法における精度向上と効率化もある．さらに，Fingerprint の偽装への対処も必要とされる．

5　まとめ

本論文では，Fingerprinting の現状と課題について示した．また，複数の特徴点を組み合わせることでモバイルデバイスにおいても，高い精度で識別できる可能性があることを示した．Fingerprint 技術の応用として，マルウエアの不正通信活動の検知に利用など様々

な応用の可能性を示した.

参考文献

[1] G Acar, C Eubank, S Englehardt, M Juarez, A Narayanan, C Diaz, The Web Never Forgets: Persistent Tracking Mechanisms in the Wild, in Proc. of the 21th ACM Conference on Computer and Communications Security, 674-689

[2] Alexa：http://www.alexa.com/

[3] Panopticlick project https://panopticlick.eff.org/

[4] P Eckersley, How Unique Is Your Web Browser?, in Proc. of Privacy Enhancing Technologies Symposium (2010), LNCS vol. 6205 2010.

[5] 齋藤孝道，磯侑斗，桐生直輝，2014，Web Browser Fingerprintingに関する技術的観点での一考察，2014暗号と情報セキュリティシンポジウム.

[6] ADTRUTH：http://www.adtruth.com/japan

[7] zanox http://www.zanox.com/us/what-we-offer/tracking/

[8] OpenAM https://www.forgerock.com/products/access-management/

[9] 細井 理央，高須 航，山田 智隆，武居 直樹，石川 貴之，高橋 和司，安田 昂樹，齋藤 孝道，ブラウザが属するネットワークの情報を採取する Browser Scannerの提案，コンピュータセキュリティシンポジウム2015論文集，2015.

[10] E. Felten, M. Schneider, Timing Attacks on Web Privacy, in Proc. of the 7th ACM Conference on Computer and Communications Security, 25-32

[11] "Passive OS Fingerprinting: Details and Techniques"(Miller,2002) http://www.ouah.org/incosfingerp.htm

[12] Browserrecon project – Advanced web browser fingerprinting http://www.computec.ch/projekte/browserrecon/

[13] K. Mowery, H. Shacham, Pixel Perfect: Fingerprinting Canvas in HTML5, in Proc. of Web 2.0 Security and Privacy (W2SP), 2012.

[14] EU Cookie Law http://www.cookielaw.org/the-cookie-law/

[15] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F Piessens, G Vigna, Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting, in Proc. of 34th IEEE Symposium of Security and Privacy (IEEE S&P 2013), 2013.

[16] 桐生直輝，磯侑斗，金子洋平，齋藤孝道，2014, Web Workers を用いた演算処理性能の差による CPUコア数の推定, コンピュータセキュリティシンポジウム2014 論文集，2014.

[17] K. Takasu, T. Saito, T. Yamada, T. Ishikawa, 2015, A Survey of Hardware Features in Modern Browsers: 2015 Edition, in Proc. of the 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS) 2015.

[18] 山田智隆，磯侑斗，桐生直輝，塚本耕司，高須航，武居直樹，齋藤孝道，2015，編集距離を用いたロバストなBrowser Fingerprint間の識別方式の提案，2015暗号と情報セキュリティシンポジウム.

[19] 石川 貴之，高須 航，山田 智隆，武居 直樹，細井 理央，高橋 和司，安田 昂樹，齋藤 孝道，Fuzzy Hashingを用いた比較による長期的なBrowser Fingerprintingの端末識別手法の提案，コンピュータセキュリティシンポジウム2015論文集，2015.

[20] 武居直樹，磯侑斗，桐生直輝，塚本耕司，高須航，山田智隆，齋藤孝道，閲覧者の端末のFingerprintをCSSのみで採用する手法の提案と実装，コンピュータセキュリティシンポジウム2014 論文集，2014.

[21] Am I Unique?：https://amiunique.org/

[22] How Unique Are You? http://www.howuniqueareyou.com/

[23] 齋藤研Fingerprint収集サイト www.saitolab.org/fingerprint

[24] W3Techs - World Wide Web Technology Surveys：http://w3techs.com/

[25] Adblock Plus https://addons.mozilla.org/ja/firefox/addon/adblock-plus/

[26] Ghostery：https://www.ghostery.com/en/

[27] FireGloves：http://fingerprint.pet-portal.eu/

[28] N. Nikiforakis, W. Joosen, B. Livshits, PriVaricator: Deceiving fingerprinters with Little White Lies, in Proc. of the 24th International Conference on World Wide Web, 820-830

[29] Chameleon https://github.com/ghostwords/chameleon

[30] Tor project https://www.torproject.org/projects/torbrowser.html.en

[31] P. Laperdrix, W. Rudametkin, B. Baudry, Mitigating browser fingerprint tracking: multi-level reconfiguration and diversification, in Proc. of the International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'15), 2015.

[32] Tracking Compliance and Scope http://www.w3.org/TR/tracking-compliance/

[33] G. Acar, M. Juárez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, B. Preneel, FPDetective: Dusting the web for fingerprinters, in Proc. of 20th ACM Conference on Computer and Communications Security (CCS 2013), 2013.

[34] G. Aggarwal, E. Bursztein, C. Jackson, and D. Boneh, An Analysis of Private Browsing Modes in Modern Browsers, In proceedings of Usenix Security 2010.

[35] A Primer on Information Theory and Privacy https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy

[36] 磯侑斗, 桐生直輝, 塚本耕司, 高須航, 山田智隆, 武居直樹, 齋藤孝道, Web Browser Fingerprint を採取する Web サイトの構築と採集データの分析, コンピュータセキュリティシンポジウム2014 論文集, 2014.
[37] http://www.shieldsquare.com/how-it-works/
[38] K . Boda, A. Földes, G. Gulyás, S. Imre, User tracking on the Web via cross-browser fingerprinting, in Proc. of 16th Nordic Conference on Information Security Technology for Applications, 2011.
[39] 塚本耕司, 磯侑斗, 桐生直輝, 高須航, 山田智隆, 武居直樹, 細井理央, 石川貴之, 齋藤孝道, 端末にインストールされているフォント情報を用いた OS とアプリケーションの特定, 2015 暗号と情報セキュリティシンポジウム論文集, 2015.
[40] 匿名の SOC 要員による情報提供.

表 2. 収集したサンプルにおける各特徴点のユニーク率

Mobile Device			全ての端末		
特徴点	ユニーク数	ユニーク率	特徴点	ユニーク数	ユニーク率
グローバル IP アドレス	81	0.698275862	プラグイン	790	0.522141441
JavaScript UserAgent	69	0.594827586	グローバル IP アドレス	639	0.422339722
HTTP UserAgent	68	0.586206897	プライベート IP アドレス	631	0.417052214
プライベート IP アドレス	28	0.24137931	フォントリスト	586	0.38730998
画面解像度	11	0.094827586	フォントリスト(no-sorted)	551	0.364177132
フォントリスト	7	0.060344828	JavaScript UserAgent	433	0.286186385
HTTP Accept-Language	3	0.025862069	HTTP UserAgent	351	0.231989425
HTTP Accept	2	0.017241379	画面解像度	79	0.052214144
HTTP Accept-Encoding	2	0.017241379	HTTP Accept Language	23	0.015201586
SSE2	2	0.017241379	Device Pixel Ratio	16	0.010575017
HTTP-Connection	1	0.00862069	HTTP Accept	10	0.006609385
HTTP Referer	1	0.00862069	タイムゾーン	5	0.003304693
タイムゾーン	1	0.00862069	HTTP Referer	4	0.002643754
Session Storage	1	0.00862069	HTTP Accept-Encoding	3	0.001982816
Local Storage	1	0.00862069	SSE2	3	0.001982816
プラグイン	1	0.00862069	HTTP Accept-Charset	3	0.001982816
HTTP Accept-Charset	0	0	HTTP Connection	1	0.000660939
HTTP Origin	0	0	Session Storage	0	0
Touch	0	0	Local Storage	0	0
Device Pixel Ratio	0	0	Touch	0	0
			HTTP Origin	0	0
20_fingerprints	112	0.965517241	21_fingerprints	1409	0.931262393
17_fingerprints	93	0.801724138	18_fingerprints	1205	0.796430932
15_fingerprints	44	0.379310345	15_fingerprints	993	0.656311963
all	116		all	1513	