Browser Fingerprinting によるスマートフォンの識別

高橋和司[1] 石川貴之[1] 細井理央[1] 安田昂樹[1] 齋藤孝道[2]

概要：端末から採取可能な複数の情報の組み合わせによって端末を識別する Browser Fingerprinting と呼ばれる手法がある．この手法は，Web 広告事業者を中心に利用されている．先行研究では，スマートフォンは PC に比べて採取可能な情報で差異が表れにくく，識別が困難であるといわれている．しかしながら，HTML5 API によって採取可能な情報が増加し，スマートフォンを識別できる可能性が出てきた．本論文では，モバイル端末における Browser Fingerprinting の実験で収集したデータを iOS 端末と Android 端末に分けて分析し，Fingerprinting を用いたスマートフォンの識別の具体的な精度を示す.

Identifying Smartphone by Browser Fingerprinting

KAZUSHI TAKAHASHI[1] TAKAYUKI ISHIKAWA[1] RIO HOSOI[1]
KOKI YASUDA[1] TAKAMICHI SAITO[2]

1. はじめに

端末から採取可能な複数の情報の組み合わせによって端末を識別する Browser Fingerprinting（以降，Fingerprinting という）と呼ばれる手法がある．この手法は，Web 広告事業者を中心に利用されている．先行研究[1][2]において，Fingerprinting を行った結果，PC の場合 90%以上の端末を識別できるが，スマートフォンでは識別が困難とされている．

しかし，HTML5 API に代表される Web 技術の発展に伴い，スマートフォンのみで採取可能な情報が表れ，Fingerprinting を用いてスマートフォンを識別できる可能性が出てきた．本論文では，HTTP ヘッダから得られる情報や JavaScript の実行により得られる情報に加えて，HTML5 API を活用し得られる情報も用いて，スマートフォンにおける Fingerprinting を行う．特に，スマートフォンでの Fingerprinting によって端末上のブラウザを識別する精度（以降，識別精度という）と，追跡の精度（以降，追跡精度という）を定め，それらをもとに，採取したサンプルに適用する．

これらの定義に関して，Eckersley[1]は Fingerprint がユニークであれば，識別が可能であると述べている．本論文ではこれをもとに，端末上のブラウザの Fingerprint がユニークであるとき識別が可能であり，一定の期間で継続的に識別が可能であるとき追跡が可能であると定義する．

本論文での実験の前提として，採取可能な情報が偽装されている場合や HTTP クッキーの削除による精度への影響を考慮しない．また，実験で採取したデータは，iOS 端末と Android 端末のデータに分けて分析する．

2. Browser Fingerprinting

2.1 Browser Fingerprint

Web サーバがブラウザを通して採取可能なブラウザや端末の情報を特徴点という．特に，端末やブラウザの利用者の特定につながるものを指す．特徴点の例としてユーザエージェント文字列（以降，UA 文字列という），インストール済みフォントリスト，画面解像度が挙げられる．特徴点の値を 1 つ以上組み合わせたものを Browser Fingerprint（以降，Fingerprint という）という．Fingerprint を採取し識別を行う手法を Fingerprinting という．

2.2 スマートフォンにおける特徴点

スマートフォンにおいて採取可能であり，スマートフォン特有の値をとる特徴点には，UA 文字列，グローバル IP アドレスおよび画面解像度がある．本節では，これらの特徴点について説明する．

・ UA 文字列

UA 文字列はブラウザや OS の種類，バージョンを表す文字列である．UA 文字列は HTTP リクエストヘッダと JavaScript の navigator.userAgent プロパティの 2 つの方法で採取することができる．UA 文字列の例を表 1 に示す．iOS 端末上や Android 端末上のブラウザでは，表 1 に示すように UA 文字列に iPhone や Android という文字列が含まれ

1 明治大学大学院
Graduate School of Meiji University
2 明治大学
Meiji University

る．よって，UA 文字列によってスマートフォンのアクセスかどうかを判別することができる．特に，Android 端末においては UA 文字列中に機種名を含むので，端末ごとの違いが表れやすい．表 1 では SonySO-04E が機種名に該当する．

表 1　UA 文字列の例

iOS 端末	Mozilla/5.0 (iPhone; CPU **iPhone OS 9_0_2** like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) CriOS/45.0.2454.89 Mobile/13A452 Safari/600.1.4
Android 端末	Mozilla/5.0 (Linux; U; **Android 4.2.2;** ja-jp; ***SonySO-04E*** Build/10.3.1.B.0.256) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30

・　グローバル IP アドレス

グローバル IP アドレスは HTTP リクエストの送信元 IP アドレスである．スマートフォンにおいて，端末にグローバル IP アドレスを割り振っている ISP（Internet Service Provider）の名前をグローバル IP アドレスから求めることができる．その方法としては，各キャリアが公開している IP アドレス帯の情報を確認する方法や Linux の whois コマンドを用いる方法が挙げられる．特に，大手キャリア三社（docomo，au，SoftBank）から販売された端末では，キャリアの名前と ISP 名を容易に紐付けられるので，グローバル IP アドレスからキャリアを推定できる．ただし，例えば docomo で販売された端末が SoftBank ルータを通じてインターネットに接続した場合, SoftBank が所持する IP アドレス帯の IP アドレスが端末に割り振られる. このような場合では, グローバル IP アドレスによる端末の販売キャリアの推定は誤った推定となる．

・　画面解像度

画面解像度は端末の画面の横幅と縦幅で表される．この値は, JavaScript の window.screen.width, window.screen.height プロパティよりそれぞれ採取することができる．画面解像度の例を表 2 に示す．画面解像度は機種によって異なるので，機種を識別する情報の 1 つとして利用できる．

表 2　iOS 端末の画面解像度の比較

	機種名		
	iPhone 5s	iPhone 6s	iPhone 6s Plus
画面解像度	320×568	375×667	414×736

2.3 先行研究

Eckersley[1]は，iOS 端末や Android 端末のようなスマートフォンにおいて，ブラウザから採取できるインストール済みプラグインリストやインストール済みフォントリストがどの端末でも似通ったものとなり，PC と比べて Fingerprint を採取しにくいと述べている．

Hupperich ら[3]は，HTTP クッキーを含む 45 個の特徴点を用いてスマートフォンの識別と追跡について評価した．HTTP クッキーを除いた 44 個の特徴点と 45 個の特徴点を用いて識別できる端末数はほぼ変化しないことから, HTTP クッキーに頼らずに識別が可能であることを示した．

Kurtz ら[4]はネイティブアプリケーションをインストールさせた場合のスマートフォンの Device Fingerprinting を行い，100%の端末が識別可能，97%の端末が追跡可能であることを示した．

Laperdrix ら[5]は，実験でスマートフォンから収集した 13,105 の Fingerprint のサンプルのうち，81%がユニークな値となることを示した．

2.4 Fingerprinting サイト

我々の研究グループでは，Fingerprint を採取する Web サイト（以降，Fingerprinting サイトという）を運営している[6]．Fingerprinting サイトでは，同一の端末上のブラウザからのアクセスであることを確認するために，ブラウザ識別用の HTTP クッキー（以降，UID という）を生成し利用している．Fingerprinting サイトの採取画面の一部を図 1 に示す．

3. 採取データの分析

3.1 データセット

本論文で使用するデータセットは，Fingerprinting サイトで 2013 年 12 月 30 日から 2015 年 10 月 12 日の期間に採取したデータの中で，UA 文字列に iPhone または Android を含むデータとした. サンプル数は iOS 端末が 519 件, Android 端末が 326 件であった. UID 数は iOS 端末が 218 件, Android 端末が 162 件であった．複数回アクセスがあった UID 数は iOS 端末が 114 件，Android 端末が 75 件であった．データセットのまとめを表 3 に示す．

表 3　データセットのまとめ

	サンプル数	UID 数	
		全て	複数回
iOS 端末	519	218	114
Android 端末	326	162	75

図 1　Fingerprinting サイトの採取画面の一部

3.2 特徴点の扱い

本節では，分析を行う際に 2.2 節で述べた特徴点をどのように扱ったかについて説明する．

・　UA 文字列

識別・追跡の実験では，UA 文字列を加工せずに用いた．UA 文字列は，加工せずに用いる利用方法とブラウザや OS のバージョンを無視する利用方法がある．識別・追跡の実験において，どちらの利用方法が妥当かを確かめるために，文字列に関する 2 つの予備実験を行った．UA 文字列に関する 1 つ目の実験（以降，予備実験 3.2-1 という）として，ある UID を持つ UA 文字列が他の全ての UID を持つ UA 文字列と異なっている割合を調査した．UA 文字列に関する 2 つ目の実験（以降，予備実験 3.2-2 という）として，同一の UID 内で UA 文字列が時間的経過を伴っても全て同じ値である割合を調査した．これらの結果を表 4 に示す．表 4 より，実験 3.2-1 では iOS 端末，Android 端末ともに割合が大きくなったのは加工せずに用いる利用方法で，実験 3.2-2 では iOS 端末，Android 端末ともに割合が大きくなったのはブラウザや OS のバージョンを無視する利用方法である．よって，加工せずに用いる．

表 4　バージョンの有無による UA 文字列の識別精度および追跡精度の比較

	iOS		Android	
バージョンの有無	有	無	有	無
予備実験 3.2-1	21.93%	5.26%	68.00%	57.33%
予備実験 3.2-2	77.19%	100.00%	90.67%	98.67%

・　グローバル IP アドレス

識別・追跡の実験では，Linux の whois コマンドによってグローバル IP アドレスから推定した ISP 名を特徴点として用いた．グローバル IP アドレスは，加工せずに用いる利用方法と ISP 名に変換する利用方法がある．前者は，他の端末が持つグローバル IP アドレスと異なるので，識別しやすいという利点がある．後者は，グローバル IP アドレスの変化を無視できるので，同一 UID を持つサンプルを継続的に同一視しやすいという利点がある．しかし，先行研究[7]により，グローバル IP アドレスは短期間で値が変わり，追跡精度を低下させることが示されている．

スマートフォンにおいてもグローバル IP アドレスが短期間で変化するかを確認するために，スマートフォンのグローバル IP アドレスが変化する期間と状況に関して，2 つの調査を行った．

初めに，スマートフォンのグローバル IP アドレスが変化する期間を明らかにするために，3.1 節で示したデータセットを用いてどれほどの期間，同一のグローバル IP アドレスを使用しているかを調査した．その結果を，図 2 に示す．ここで，同一のグローバル IP アドレスを使用しているというのは，同一 UID と ISP 名を持つサンプルのグローバル IP アドレスが変化しないことを指す．

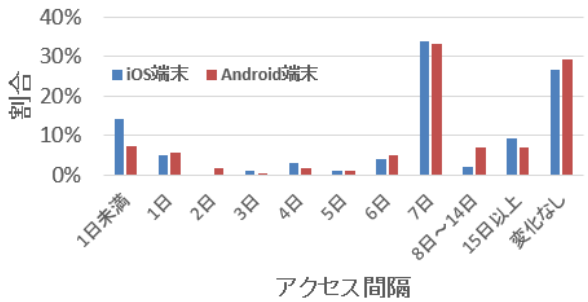

図 2　グローバル IP アドレスが変化する期間

図 2 より，iOS 端末，Android 端末ともに約 60%の端末が長くとも 1 週間，同一のグローバル IP アドレスを使用することが分かる．

次に，実際の端末を用いてグローバル IP アドレスが変化する状況について調査した結果，グローバル IP アドレスは通信の接続が切断する前後で変化することが確認できた．確認できた状況は，LTE 環境→Wi-Fi 環境→LTE 環境のよ

うに他の ISP と接続した前後，端末を再起動した前後，圏外または機内モードになった前後である．

1 つ目の調査より 60%のスマートフォンのグローバル IP アドレスが 1 週間以内に変化することが分かった．また，2 つ目の調査で示した通信の接続が切断する状況は，日常生活において多く発生すると予想される．これらの結果より，グローバル IP アドレスの値を特徴点として使用した場合，1 週間以上の追跡は困難となることが分かる．よって，Linux の whois コマンドによってグローバル IP アドレスから推定した ISP 名を特徴点として用いる．

グローバル IP アドレスを ISP 名に変換し，ISP 名が変化する頻度を示した結果を図 3 に示す．図 3 より，ISP の保有する IP アドレス帯が変更されることがなかった．

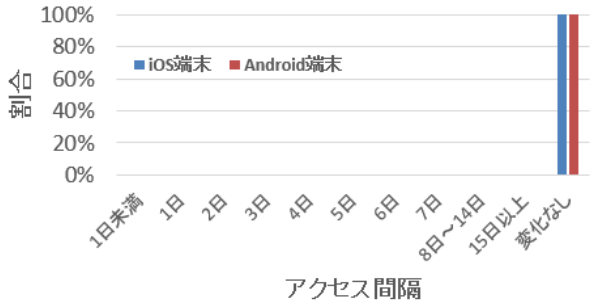

図 3　ISP 名が変化する期間

・　画面解像度

識別・追跡の実験において，画面解像度は縦幅と横幅の順序を考慮しない処理をした．画面解像度は，縦幅と横幅の順序を考慮しない利用方法と加工せずに用いる利用方法がある．例えば，2 つのサンプルの画面解像度が 360×640 と 640×360 であった場合に，前者の方法では，これらを同一の値であるとみなす．それぞれの利用方法において同一の UID 内で画面解像度が全て同じ値である割合を調査した．その結果を表 5 に示す．表 5 より，縦幅と横幅の順序を考慮しない利用方法は加工せずに用いる利用方法と比べて，iOS 端末では割合が等しく，Android 端末では精度が高い．これは，縦横が反転しているサンプルが iOS 端末のデータセットには含まれず，Android 端末のデータセットには 2 件含まれていたことが原因である．一般的に，縦横が反転しているサンプルを採取する場合は少なく，縦幅と横幅の順序を考慮しない利用方法を採用することによる識別精度の低下は起こりにくいと予想する．よって，縦幅と横幅の順序を考慮しない利用方法を採用する．

3.3 エントロピー

特徴点の識別能力の評価のための指標として Shannon エントロピー（以降，エントロピーという）を用いる．しかし，特徴点のエントロピーは，サンプル数によって異なる．

表 5　ソートの有無によって画面解像度を同一視できる割合の比較

	iOS		Android	
ソート	有	無	有	無
同一である割合	99.12%	99.12%	97.33%	94.67%

そこで，(1) 式で表す Normalized Shannon's entropy（以降，NE という）を使用する．ここで，H (X) は特徴点 X のエントロピーを表す．また，N はサンプル数を表す．

$$NE = \frac{H(X)}{\log_2(N)} \quad (1)$$

PC，iOS 端末，Android 端末における各特徴点の NE の比較を表 6 に示す．表 6 は PC において NE の値が大きい順にソートしている．識別に役立つ情報を区別するために NE の値が 0.5 より大きい欄は背景色を橙色にしている．ここで，3_Fingerprints は UA 文字列，ISP 名および画面解像度の組み合わせとする．

表 6　各端末における NE の比較

#	特徴点	PC	iOS 端末	Android 端末
1	インストール済みフォントリスト	0.700	0.000	0.068
2	インストール済みプラグインリスト	0.679	0.134	0.052
3	プライベート IP アドレス	0.661	0.007	0.692
4	UA 文字列（HTTP ヘッダ）	0.646	0.552	0.818
5	UA 文字列（JavaScript）	0.635	0.564	0.818
6	グローバル IP アドレス	0.609	0.897	0.876
7	画面解像度	0.368	0.165	0.327
8	http_accept_language	0.252	0.110	0.361
9	デバイスピクセル比	0.211	0.055	0.201
10	タッチ機能	0.170	0.013	0.146
11	http_accept	0.142	0.059	0.134
12	http_origin	0.131	0.050	0.123
13	タイムゾーン	0.128	0.026	0.050
14	http_referer	0.120	0.092	0.108
15	http_connection	0.119	0.049	0.101
16	SSE2	0.112	0.000	0.000
17	http_accept_encoding	0.110	0.004	0.201
18	ローカルストレージ利用可否	0.091	0.000	0.009
19	セッションストレージ利用可否	0.042	0.000	0.006
20	http_accept_charset	0.007	0.000	0.106
21	3_Fingerprints (#5, #6, #7)		0.786	0.841

表 6 より，PC においては NE の値が大きいインストール

済みフォントリストとインストール済みプラグインリストが iOS 端末および Android 端末では小さくなっていることが分かる．また，UA 文字列と比較した際の 3_Fingerprints の NE に着目すると，iOS 端末では Android 端末に比べて NE の値が大きく増加している．これは，iOS 端末では UA 文字列に機種名を含まないので，機種を推定する情報が画面解像度から得られたことが理由として考えられる．

3.4 識別精度および追跡精度の算出方法

識別・追跡の実験では，複数のアクセス間で 3_Fingerprints が完全一致したときのみ同一の端末上のブラウザからのアクセスと判定する．判定の成否は，HTTP クッキーによる UID を用い，UID が同一であるサンプルは同一の端末上のブラウザからのアクセスとする. 本論文では, TP, TN, FP, FN の 4 つの判定結果を識別精度と追跡精度の算出に用いる．以下，これら 4 つの判定結果について説明する．

・TP (True Positive)

TP は真陽性ともいう．実際に同一であるものを，予測でも同一であると判定した結果が TP である．識別・追跡の実験では，それぞれの UID で，判定対象となる UID を持つすべてのサンプルの 3_Fingerprints が変わらないとき, TP と判定する．TP と判定される UID が多いほど追跡精度が良くなる．

・TN (True Negative)

TN は真陰性ともいう．実際に異なるものを，予測でも異なると判定した結果が TN である．識別・追跡の実験においては，それぞれの UID で，対象となる UID を持つすべてのサンプルの 3_Fingerprints が他の UID の 3_Fingerprints と一致しないとき，TN と判定する．TN と判定される UID が多いほど識別精度が良くなる．

・FP (False Positive)

FP は偽陽性，誤検知ともいう．実際には異なるものを，予測では同一であると判定した結果が FP である．識別・追跡の実験では，それぞれの UID で，対象となる UID を持つサンプルの 3_Fingerprints が 1 つでも他の UID の 3_Fingerprints と一致するとき，FP と判定する．FP と判定される UID が多いほど識別精度が悪くなる．

・FN (False Negative)

FN は偽陰性，見逃しともいう．実際には同一であるものを，予測では異なると判定した結果が FN である．識別・追跡の実験では，それぞれの UID で，判定対象となる UID を持つサンプルの 3_Fingerprints が，1 つでも異なるとき，FN と判定する．FN と判定される UID が多いほど追跡精度が悪くなる．

以上，4 つの判定結果をまとめたものを表 7 に示す．

表 7　TP・TN・FP・FN の関係

		3_Fingerprints	
		同一	異なる
UID	同一	TP	FN
	異なる	FP	TN

3.4.1 識別精度の算出方法

識別精度（以降，I という）は，以降の (2) 式で算出する．式中では，複数回アクセスがあった UID 数を S と表記する．

$$I = \frac{|TN|}{S} \times 100 \quad (2)$$

これは, 異なる端末上のブラウザを 3_Fingerprints によって異なる端末上のブラウザであると判定できる割合である．本論文では，これを識別精度と定義する．

3.4.2 追跡精度の算出方法

追跡精度（以降，T という）は，以降の (3) 式および (4) 式で算出する．

$$T = \frac{TB}{S} \times 100 \quad (3)$$

$$TB = |S - FP \cup FN| \quad (4)$$

FP が発生しないことは識別できることを表し，FN が発生しないことはあるブラウザを継続的に同一視できている状態を表す. よって, (4) 式は継続的に識別できる端末上のブラウザ数を表している．本論文では，これを追跡精度と定義する．

今回の実験では，どれほどの期間で追跡精度 T が低下するかを調べるために，アクセス間隔ごとにサンプルを限定した場合の追跡精度 T も算出した．今回の実験において，アクセス間隔は UID ごとのあるアクセスと次のアクセスの間隔とする．アクセス間隔を 1 週間ごとに 1 週間から 6 週間まで限定した 6 種類の追跡精度 T を算出する．

4. 実験

4.1 識別精度

3_Fingerprints による識別精度 I を算出した（以降，実験 1 という）ところ, 識別精度 I は iOS 端末で 50.88%, Android 端末で 80.00%であった．また，3_Fingerprints から ISP 名を除いた組み合わせでも識別精度 I を算出した（以降，実験 2 という）ところ, 識別精度 I は iOS 端末で 27.19%, Android 端末で 77.33%であった．これらの実験の結果を表 8 に示す．

表 8　識別精度

	iOS	Android
実験 1	50.88%	80.00%
実験 2	27.19%	77.33%

3_Fingerprints を用いて識別精度 I を算出する実験 1，3_Fingerprints から ISP 名を除いた組み合わせを用いて識別精度 I を算出する実験 2 の TP，TN，FP，FN となる UID 数を表 9 に示す．

表 9　TP，TN，FP，FN となる UID 数

	実験 1		実験 2	
	iOS	Android	iOS	Android
| TP |	76	57	86	61
| TN |	58	60	31	58
| FP |	56	15	83	17
| FN |	38	18	28	14
S	114	75	114	75

4.2 追跡精度

3_Fingerprints による追跡精度 T は iOS 端末で 32.46%，Android 端末で 60.00%であった．また，アクセス間隔ごとにサンプルを限定した追跡精度 T を算出した．これらの結果を表 10 に示す．表 10 より，iOS 端末では 1 週間と 2 週間の間で精度が低下し，Android 端末では 2 週間と 3 週間の間で精度が低下していることが分かる．

表 10　追跡精度

	iOS	Android
1 週間	35.14%	67.12%
2 週間	33.93%	65.75%
3 週間	33.63%	62.16%
4 週間	33.63%	61.33%
5 週間	33.33%	61.33%
6 週間	33.33%	61.33%
全期間	32.46%	60.00%

5. 結果の考察

表 7 より，iOS，Android 端末ともに ISP を除いた実験 2 は実験 1 に比べ識別精度 I が低い．これは ISP 名のみで識別できていたブラウザが存在していたことが原因と考えられる．

実験 1 に比べ，実験 2 での識別精度 I の低下は Android 端末の方が小さい．これは，Android 端末では UA 文字列に含まれる機種名が，特定のキャリアからしか販売されていないケースがあり，ISP 名から得られるはずのキャリアの情報が一部の UA 文字列から得られたことが理由である．

iOS 端末は Android 端末に比べて追跡精度 T が低い．これは，iOS 端末は UA 文字列に機種名を含まないので，UA 文字列により識別できる端末上のブラウザが少なくなり，それが追跡精度 T に影響したと考えられる．

UID ごとにアクセス間隔を限定した場合の追跡精度 T は，iOS 端末では 1 週間と 2 週間の間，Android 端末では 2 週間と 3 週間の間で追跡精度 T が比較的大きく低下している．これは，それぞれの端末で先ほど示した期間中にブラウザのマイナーバージョンアップがあったことが理由として考えられる．

アクセス間隔を 1 週間に限定したサンプルでの追跡精度 T は，iOS 端末，Android 端末ともに全期間での追跡精度 T に比べて高いブラウザのマイナーバージョンアップの影響を受ける端末上のブラウザが少なくなったことが理由として考えられる．

6. 今後の課題

今回の実験では，iOS 端末，Android 端末の両端末において識別精度 I，追跡精度 T ともに iOS 端末は Android 端末より精度が低い結果となった．

3_Fingerprints を用いた識別・追跡では，機種，ブラウザ，ISP が同一だった場合は同一の Fingerprint を生成することとなる．今後の課題としては，同一の機種，ブラウザ，ISP を利用している利用者を識別できるような特徴点を導入することが挙げられる．

7. まとめ

本論文では，iOS 端末および Android 端末における Fingerprinting を用いた識別について示した．UA 文字列，ISP 名および画面解像度を用いることで Android 端末では 80%の精度で識別でき，60%の精度で追跡できることを示した．また，iOS 端末は Android 端末に比べて Fingerprinting を用いた識別能力および追跡能力が低いことが分かった．

8. 参考文献

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

[2] 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.

[3] T Hupperich, D Maiorca, M Kührer, T Holz, G Giacinto, On the Robustness of Mobile Device Fingerprinting, the 31th Annual Computer Security Applications Conference (ACSAC), pp.191-200,

2015.

[4] A Kurtz, H Gascon, T Becker, K Rieck, F Freiling, Fingerprinting Mobile Devices Using Personalized Configurations, in Proc. of Privacy Enhancing Technologies (PoPETS), pp.4–19, 2016.

[5] P Laperdrix, W Rudametkin, B Baudry, Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints, in Proc. of 37th IEEE Symposium on Security and Privacy (S&P 2016), 2016.

[6] https://www.saitolab.org/fingerprint/.

[7] 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 サイトの構築と採取データの分析”，コンピュータセキュリティシンポジウム 2014 , 2014.