スマートフォンにおける Browser Fingerprinting

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を用いて端末内のブラウザを識別する Browser Fingerprinting と呼ばれる手法がある．先行研究において，スマートフォンから採取可能な情報は端末ごとの差異が表れにくく，PC と比べて識別が難しいと指摘されている．しかし，我々の実験では，HTML5 API などを活用した独自の Browser Fingerprinting により，スマートフォンで端末ごとの差異が表れた．さらに，収集したデータを iOS 端末と Android 端末に分けて分析した結果，iOS 端末は 90.5%，Android 端末は 96.6%で識別でき，スマートフォンを追跡できる可能性があることがわかった．

キーワード：Browser Fingerprinting，スマートフォン，Web トラッキング，プライバシー

Browser Fingerprinting in Smartphone

Kazushi TAKAHASHI [†1]　Takayuki ISHIKAWA [†1]　Rio HOSOI [†1]
Koki YASUDA [†1]　Takamichi SAITO [†2]

Abstract: There is a technique to identify browser on the device by using plenty of information collected from the devices, it called Browser Fingerprinting. In previous research, it was pointed out that identifying smartphone is difficult compared with PC. Because information collected from smartphone is uniformed. However, in the our experiment, by its own Browser Fingerprinting utilizing such as HTML5 API, differences of each devices appeared in the smartphone. Further, when the collected data were analyzed separately to iOS devices and Android devices, identification accuracy was 90.5% in iOS devices, 96.6% in the Android devices. We found probability of tracking smartphone.

Keywords: Browser Fingerprinting, Smartphone, Web Tracking, Privacy

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内のブラウザを識別する 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．Englehardt ら[1]の調査によると，Alexa の Top100 万サイトのうち Fingerprinting に関する技術を用いるサイトは 14,371（1.6%）であった．

先行研究[2][3]において，Fingerprinting を行った結果，PC の場合 90%以上の端末内のブラウザを識別できるが，スマートフォンでは識別が困難とされている．識別が困難な主な理由として，端末にインストールされているフォントやプラグインのリストをスマートフォンから採取するのが難しいということが示されている．

本論文では，スマートフォンでの Fingerprinting によって端末内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 PC，iOS 端末および Android 端末に分けて算出する．Fingerprinting 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行う．その結果，識別精度が iOS 端末では 90.5%，Android 端末では 96.6%となり，スマートフォンにおいても追跡できる可能性があることがわかった．

2 節では，Browser Fingerprinting の定義やスマートフォンで採取できる情報について説明する．我々の研究室が Fingerprinting を行うために開設した Web サイト（以降，Fingerprinting サイトという）についても説明する．3 節では，Fingerprinting サイトで収集したデータの中で，識別・追跡の実験に用いたデータについて説明する．また，Fingerprinting における，識別精度，追跡精度の算出方法についても本論文独自のものを定義するので，ここで説明する．4 節では，実験結果である識別精度と追跡精度を示す．PC における精度と比較し，スマートフォンにおいて Fingerprinting がどれほどの端末内のブラウザに対して可能かを示す．5 節では，実験結果の考察を行う．6 節では今後の課題を述べる．

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

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

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 つの方法で採取できる．UA 文字列の例を表 1 に示す．

iOS 端末上や Android 端末内のブラウザでは，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, ISP（Internet Service Provider）から端末に割り振られた IP アドレスである．HTTP リクエストヘッダから採取できる．

スマートフォンにおいて，ISP の名前（以降，ISP 名という）はグローバル IP アドレスを基にして求められる．その方法として, 各 ISP に割り振られている 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，通信キャリア大手三社（docomo，au，SoftBank）から販売された端末などにより，利用するキャリアの名前も推定できる．これは，キャリアの名前と ISP 名が同一である場合や一意に特定できる場合が多いからである．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結果となる．

・ 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．画面解像度の例を表 2 に示す．

表 2　画面解像度の例

	機種名		
	iPhone 5s	iPhone 6s	Sony SO-04E
画面解像度	320×568	375×667	360×640

2.3 関連研究

Eckersley[2]は，iOS 端末や Android 端末のようなスマートフォンでは，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，Fingerprinting が困難であると述べている．また，多くの端末ではインストール済みフォントリストが採取できない．これは，採取に用いる Flash がスマートフォンには基本的にインストールされていないことが理由である．

Hupperich ら[4]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 クッキーを用いず，LocalStorage や SessionStorage に格納された識別子を含めた場合，高い精度で識別できることを示した．

Kurtz ら[5]はアプリケーションをインストールさせ，そのアプリケーションから収集した情報を用いてスマートフォンの 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．ただし，Browser Fingerprinting の結果ではない．

Laperdrix ら[6]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．この実験では，他の値と重複しないサンプルが 81%あったことのみを示しており，識別は行っていない．

2.4 Fingerprinting サイト

Fingerprinting サイト[7]では，同一の端末内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 クッキーを生成し利用している．以降，この HTTP クッキーの値を UID（Unique IDentifier）という．Fingerprinting サイトの採取画面の一部を図 1 に示す．

図 1　Fingerprinting サイトの採取画面の一部

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 2014 年 9 月 10 日から 2016 年 7 月 28 日の期間に採取したデータである．このうち，UA 文字列に iPhone を含むデータを iOS 端末のデータ，Android を含むデータを Android 端末のデータ，どちらも含まれないデータを PC のデータとした. サンプル数は PC が 5,254 件, iOS 端末が 1,711 件, Android 端末が 1,036 件であった. UID 数は PC が 1,881 件, iOS 端末が 394 件，Android 端末が 269 件であった．複数回アクセスがあった UID 数は PC が 730 件，iOS 端末が 193 件，Android 端末が 144 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
PC	5,254	1,881	730
iOS 端末	1,711	394	193
Android 端末	1,036	269	144

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 文字列の利用には，加工せずに用いる方法（以降，バージョン有という）と，ブラウザや OS のバージョンを無視する方法（以降，バージョン無）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．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 内で時間的経過を伴っても UA 文字列が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，予備実験 3.2-1 においてバージョン無では他の端末のサンプルとの重複が多くなってしまう．予備実験 3.2-2 の割合は予備実験 3.2-1 で示された UA 文字列のうちどれほどの UA 文字列が変化しなかったかを示す. よって, 予備実験 3.2-1 の結果を重視し, バージョン有を採用する．

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 端末		Android 端末	
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 アドレスから求めた 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[8]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．

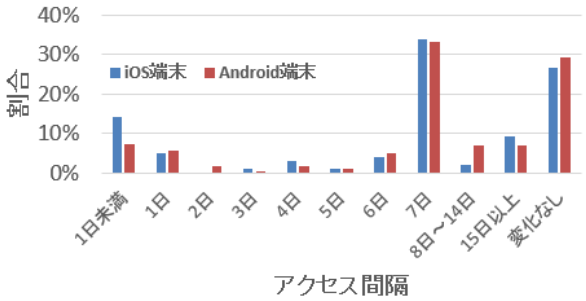

図 2　グローバル IP アドレスが変化する期間

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わかる．

次に，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わかった．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わかる．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，グローバル IP アドレスとは異なり，ISP 名は長期間にわたって変化しないことがわかる．

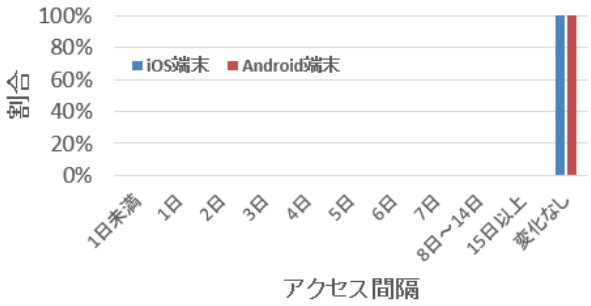

図 3　ISP 名が変化する期間

・　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．画面解像度の利用には，縦幅と横幅の順序を考慮しない方法と加工せずに用いる方法がある．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方法を採用する．

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

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）を用いることが一般的だが，特徴点のエントロピーは，サンプル数によって異なる．そこで，式 (1)で表す Normalized Shannon's entropy（以降，NE 値という）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．

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

PC, iOS 端末および Android 端末における各特徴点の NE 値の比較を表 6 に示す．表 6 は PC において NE 値が大きい順にソートしている．ここで，NE 値が 0.5 より大きい欄は背景色を橙色にしている．

本論文では，各特徴点の NE 値以外に 20_Fingerprints, 18_Fingerprints および 3_Fingerprints の NE 値も示す．ここで，20_Fingerprints は我々の先行研究で使用した 21_Fingerprints[8]からソートされたフォントリストを除いた組み合わせである．18_Fingerprints は 20_Fingerprints から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を除いた組み合わせである．この 2 特徴点を除いた理由は, PC とスマートフォンの識別精度の差に関して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の採取可否以外の要因を明らかにするためである．3_Fingerprints はユーザエージェント文字列（JavaScript），グローバル IP アドレスを基にした ISP 名，画面解像度である．この 3 特徴点は，採取したデータを目視し，スマートフォンにおいて識別に役立つと判断した．

表 6　各端末における NE 値の比較

#	特徴点	PC	iOS 端末	Android 端末
1	プライベート IP アドレス	0.759	0.012	0.697
2	インストール済みプラグインリスト	0.686	0.064	0.012
3	UA 文字列（JavaScript）	0.681	0.509	0.726
4	UA 文字列（HTTP ヘッダ）	0.660	0.483	0.726
5	インストール済みフォントリスト	0.597	0.001	0.015
6	グローバル IP アドレス	0.534	0.917	0.911
7	画面解像度	0.300	0.126	0.255
8	http_accept_language	0.222	0.101	0.271
9	デバイスピクセル比	0.136	0.033	0.164
10	http_accept	0.121	0.050	0.099
11	http_accept_encoding	0.115	0.003	0.135
12	http_origin	0.094	0.004	0.027
13	http_referer	0.090	0.084	0.050
14	タッチ機能	0.035	0.008	0.123
15	タイムゾーン	0.030	0.010	0.025
16	http_connection	0.020	0.001	0.002
17	ローカルストレージ利用可否	0.004	0.000	0.009
18	セッションストレージ利用可否	0.003	0.000	0.008
19	SSE2	0.001	0.000	0.001
20	http_accept_charset	0.001	0.000	0.086
	20_Fingerprints	0.949	0.951	0.953
	18_Fingerprints（#2, #5 を除く）	0.940	0.951	0.953
	3_Fingerprints（#3, #6, #7）	0.825	0.946	0.939

表 6 より，PC においては NE 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わかる．また，UA 文字列と比較した際の 3_Fingerprints の NE 値に着目すると，iOS 端末では Android 端末に比べて NE 値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

識別精度は，サンプル間の Fingerprint による全ての比較結果のうち正解の比較結果の割合で算出する．ここで，正解の比較結果の割合とは，Fingerprint による比較結果と UID による比較結果が一致する割合とする．追跡精度は，アクセス間隔が 21 日～28 日のサンプルにおける識別精度とする．

サンプル間の検証方法には LOOCV（Leave-One-Out Cross Validation）を採用する．

同一の端末内のブラウザからのアクセスであるか否かは，Fingerprint の各特徴点で不一致となる数（以降，不一致数という）を求め，不一致数が不一致を許す閾値を超えるかで判定する．不一致を許す閾値は 0 から特徴点の数までの範囲で値をとる．特に，不一致を許す閾値が 0 のとき，Fingerprint の完全一致による識別となる．例えば，不一致を許す閾値が 2 の場合は，不一致である特徴点が 2 つ以内であれば同一の端末内のブラウザからのアクセスと判定する．

識別精度および追跡精度は，AUC（Area Under the Curve）によって示す．ここで，AUC は ROC（Receiver Operating Characteristic）曲線下の面積であり，識別精度が高いほど AUC が高い値をとる． ROC 曲線は縦軸に TP 率，横軸に FP 率をとることで描画する．TP 率，FP 率はこの節の末尾にて後述する．ここで，本論文での TP・TN・FP・FN についての定義を示す．

・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みなした結果が TP である．識別・追跡の実験では，サンプル間の UID が同一の場合に，不一致数が不一致を許す閾値以下であれば，比較結果は TP となる．

・TN (True Negative)

TN は真陰性ともいう．実際に異なるものを，予測でも異なるとみなした結果が TN である．識別・追跡の実験では，サンプル間の UID が異なる場合に，不一致数が不一致を許す閾値より大きければ，比較結果は TN となる．

・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みなした結果が FP である．識別・追跡の実験では，サンプル間の UID が異なる場合に，不一致

数が不一致を許す閾値以下であれば，比較結果は FP となる．

・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，サンプル間の UID が同一の場合に，不一致数が不一致を許す閾値より大きければ，比較結果は FN となる．

TP 率と FP 率は以下の式で求まる．

$$\text{TP 率} = \frac{|\text{TP}|}{|\text{TP} + \text{FN}|} \quad (2)$$

$$\text{FP 率} = \frac{|\text{FP}|}{|\text{FP} + \text{TN}|} \quad (3)$$

閾値を特徴点の数だけ変化させたときの TP 率と FP 率を求めることで，ROC 曲線を描画する．

3.4.1 識別精度・追跡精度の算出方法

本論文では識別精度として，AUC を用いる．

追跡精度は，アクセス間隔が 21-28 日の場合における識別精度とした．

識別精度を求めるために他に考えられる方法として，正確度がある．これは，予測結果が実際の結果をどれだけ当てられているかを求めるために用いられる．正確度（Accuracy）は以下の式で求められる．

$$\text{Accuracy} = \frac{|\text{TP} + \text{TN}|}{|\text{TP} + \text{TN} + \text{FP} + \text{FN}|} \quad (4)$$

サンプルの偏りが大きく影響するので，本論文において正確度を識別精度として用いるのは不適切と判断した．具体例として，20_Fingeprints を用いて，不一致を許す閾値を 0 としたときの TP・TN・FP・FN の数を表 7 として示す．

表 7　20_Fingerprints を用いた閾値 0 の場合の比較総数

	PC	iOS 端末	Android 端末
TP	2,686	757	322
FN	26,041	11,237	4,541
TN	13,838,912	1,450,851	531,240
FP	372	60	27
TP+FN	28,727	11,994	4863
TN+FP	13,839,284	1,450,911	531,267

表 7 より，識別・追跡の実験で用いたサンプルでは，TN・FP のサンプル数の和が TP・FN のサンプル数の和よりも圧倒的に多いことがわかる．これは，前者の比較対象が他の UID を持つ全てのサンプルであるのに対し，後者の比較対象は同一の UID を持つ他の全てのサンプルのみであることが理由である．

次に，先行研究[9]で示された識別方法と，本論文での識別方法の違いを示す．

1 点目として，UID 単位の比較からサンプル単位の比較に修正した．以前の手法では，全ての UID の中で正解の UID の割合を求めることで精度を算出していた．この場合，同じ UID を持つサンプルの中に 1 つでも不正解の比較結果が存在すると，その他のサンプルも使用できず，精度低下の要因となった．

2 点目に同一の端末内のブラウザからのアクセスか否かの判定を完全一致のみならず閾値も用いる修正を行った．特徴点ごとに比較し，それらの不一致数を閾値と比較することで識別の成否を判断した．以前の手法ではサンプル間の比較を行う際に完全一致した場合のみ同一，それ以外は異なるとしていた．これは本論文の手法における不一致を許す閾値が 0 の場合のみを表している．

これらの改善により，柔軟な識別精度の評価が可能となった．

4. 実験

本節では，識別精度と追跡精度を ROC 曲線も用いて示す．

実験の前提として，Fingerprint による比較結果の成否は UID の比較結果と合致するかによって判定している．また，採取可能な情報が偽装されている場合は考慮しない．

4.1 識別精度

20_Fingerprints, 18_Fingerprints および 3_Fingerprints の識別精度について，それぞれ算出した．PC，iOS 端末および Android 端末の識別精度を表 8 に示す．

表 8　識別精度

	PC	iOS 端末	Android 端末
20_Fingerprints	0.966	0.905	0.966
18_Fingerprints	0.948	0.897	0.965
3_Fingerprints	0.879	0.905	0.924

表 8 より，20_Fingerrptins と 18_Fingerprints を比較したときの精度の低下は PC がスマートフォンよりも大きいことがわかる．このことから，PC において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識別に役立っていることがわかる．また，20_Fingerprints を用いた場合，Android 端末は PC や iOS 端末よりも識別精度が高いことがわかった．

PC，iOS 端末および Android 端末での 20_Fingerprints，18_Fingerprints および 3_Fingerprints の識別精度を以下にそれぞれ図 4，図 5 および図 6 として ROC 曲線で示す．20_Fingerprints において，FP 率が 5%のとき，PC と Android 端末の TP 率は約 80%であり，iOS 端末の TP 率は約 60%である．FP 率が 10%のとき，PC と Android 端末の TP 率は 90.5%であり，iOS 端末の TP 率は約 80%である．20_Fingerprints を用いた場合，iOS 端末は PC や Android 端末よりも識別精度が低いことがわかった．

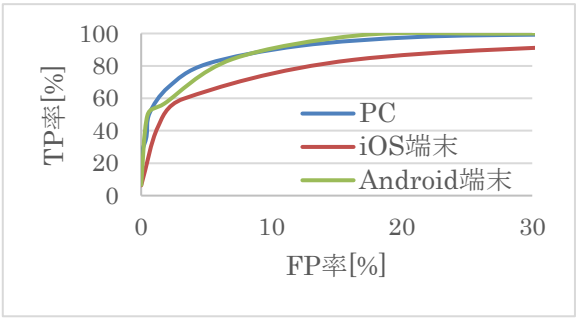

図 4　20_Fingerprints を用いた識別精度

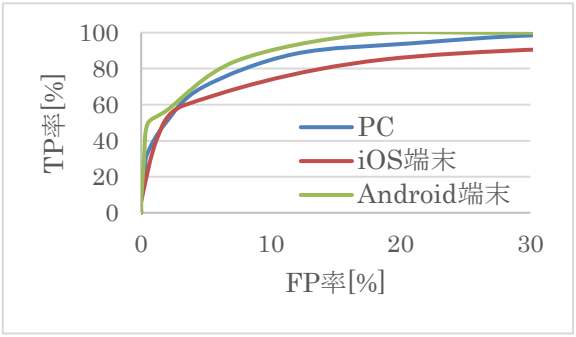

図 5　18_Fingerprints を用いた識別精度

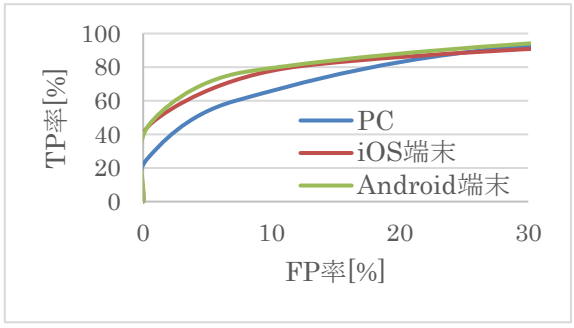

図 6　3_Fingerprints を用いた識別精度

4.2　追跡精度

追跡精度として，アクセス間隔が 21 日～28 日のサンプル間の比較結果を基にした識別精度を示す．20_Fingerprints，18_Fingerprints および 3_Fingerprints のそれぞれについて算出した．PC，iOS 端末および Android 端末の追跡精度を表 9 に示す．

表 9　追跡精度

	PC	iOS 端末	Android 端末
20_Fingerprints	0.944	0.921	0.980
18_Fingerprints	0.914	0.916	0.979
3_Fingerprints	0.798	0.915	0.955

PC，iOS 端末および Android 端末での 20_Fingerprints，18_Fingerprints および 3_Fingerprints の追跡精度を以下にそれぞれ図 7，図 8 および図 9 として ROC 曲線で示す．20_Fingerprints において，FP 率が 5%のとき，Android 端末の TP 率は約 80%であり，PC と iOS 端末の TP 率は約 60%である．FP 率が 10%のとき，Android 端末の TP 率は約 100%，PC の TP 率は 90%，iOS 端末の TP 率は約 80%である．20_Fingerprints を用いた場合，Android 端末は PC や iOS 端末よりも追跡精度が高いことがわかった．

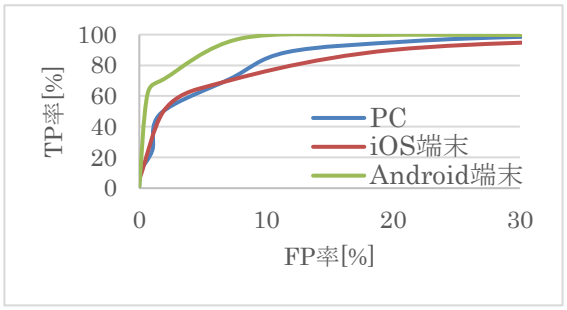

図 7　20_Fingerprints を用いた追跡精度

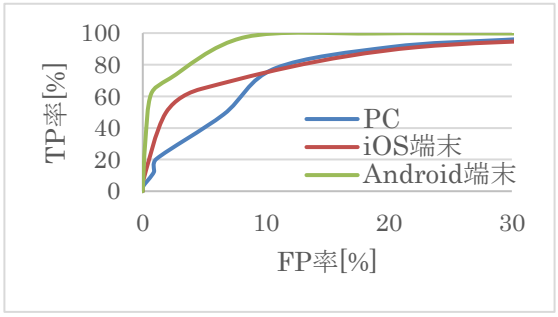

図 8　18_Fingerprints を用いた追跡精度

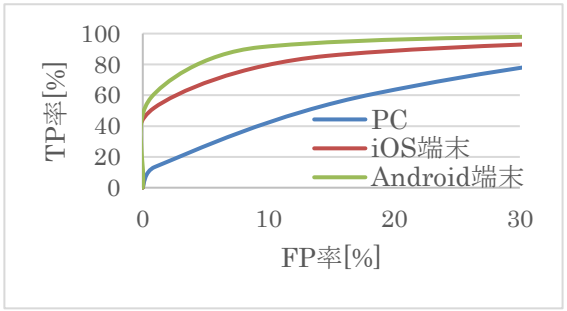

図 9　3_Fingerprints を用いた追跡精度

5. 結果の考察

PC では識別精度よりも追跡精度の方が低くなっているが，スマートフォンで追跡精度の方が高くなっている．この原因として，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短期間で変わりやすいことが関係していると考えられる．また，長期間にわたって継続的に Fingerprinting サイトにアクセスしているサンプルはスマートフォンよりも PC が多いことも原因と考えられる．

本論文で示した結果の全てにおいて，Android 端末は PC や iOS 端末よりも精度が高くなっている．この原因として，Android 端末の UA 文字列にのみ端末の機種名が記載されていることが考えられる．さらに，タッチ機能およびデバイスピクセル比の値が PC や iOS 端末のものに比べて Android 端末では多様であることも原因として考えられる．しかし，タッチ機能およびデバイスピクセル比の値は機種名によって一意に決まるものである．よって，Android 端末の精度の高さは UA 文字列にのみ端末の機種名が記載されているという原因によって説明できる．

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，追跡精度ともに 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．

iOS 端末における識別精度および追跡精度を求める際に 3_Fingerprints が最良の組み合わせであることを示すことも課題である．

7. まとめ

本論文では，iOS 端末，Android 端末の両端末において Fingerprinting を用いた識別について示した．識別精度が iOS 端末では 90.5%，Android 端末では 96.6%となり，スマートフォンにおいて Fingerprinting を用いて追跡できる可能性があることがわかった．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わかった．

参考文献

[1] https://webtransparency.cs.princeton.edu/webcensus/

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 高橋和司，石川貴之，細井理央，安田昂樹，齋藤孝道，2016，Browser Fingerprinting によるスマートフォンの識別，マルチメディア，分散，協調とモバイル(DICOMO2016)シンポジウム 論文集 CD-ROM p.659-p.665