プライベートブラウジングにおける Browser Fingerprinting を用いた Web 行動追跡

石川 貴之[†1] 細井 理央[†1] 安田 昂樹[†1] 高橋 和司[†1] 齋藤孝道[†2]

概要：プライベートブラウジングは，複数のユーザが共用のデバイスを利用する際のプライバシーを守る機能として 2005 年に Safari に導入され，その後 Firefox や Google Chrome などの主要ブラウザにも導入された．この機能はクッキーを用いた Web 行動追跡を防ぐことができる．さらに Web 行動追跡の対策機能をプライベートブラウジングに導入するブラウザベンダも現れた．本論文では，Web 行動追跡技術として利用される Browser Fingerprinting を用いて，プライベートブラウジングにおけるブラウザの識別精度を計測した．その結果，プライベートブラウジングは Web 行動追跡の対策として効果が低いことがわかった．

キーワード：プライベートブラウジング，Browser Fingerprinting，Web 行動追跡

Web Tracking of Private Browsing with Browser Fingerprinting

Takayuki Ishikawa[†1] Rio Hosoi[†1] Koki Yasuda[†1]
Kazushi Takahashi[†1] Takamichi Saito[†2]

Abstract: Major browsers implement private browsing in from 2005, it protects your privacy when you use the sharing device. Private Browsing can prevent the Web tracking using cookie. In addition, Mozilla introduced Tracking protection in private browsing. In this paper, we measure the identification accuracy of the private browsing with Browser Fingerprinting. As a result, we found that the effect as a measure of Web tracking in the private browsing is low.

Keywords: Private Browsing, Browser Fingerprinting, Web Tracking

1. はじめに

プライベートブラウジングは，複数のユーザが共用のデバイスを利用する際のプライバシーを守る機能として 2005 年から Safari に導入され，その後 Firefox や Chrome などで導入された．表 1 は主要ブラウザにおけるプライベートブラウジング機能が導入された年と名称を示している．ブラウザにより一部名称が異なるが，本論文では，これらをプライベートブラウジングとよぶ．

プライベートブラウジングは一部の Web 行動追跡を防ぐことができるとされている．他方，Web 行動追跡を実現する手法としては，タイミングアタックを用いて閲覧履歴を取得する手法や，クッキーなどを利用してクライアント側に識別子を保存する手法などが存在する．プライベートブラウジングでは Web の閲覧履歴やクッキーをストレージに保存しないので，これらの Web 行動追跡を防ぐことができる．

Internet Explorer (IE)や Firefox など主要なブラウザの多くは，プライベートブラウジングに Do Not Track のフラグをデフォルトでオンにするなど，Web プライバシーへの対応も行っている．さらにユーザのプライバシー意識の高まりから，Firefox でプライベートブラウジングにトラッキングプロテクション[1]を導入するなど，Web 行動追跡の対策機能をプライベートブラウジングに導入するブラウザベンダも現れた．2016 年 1 月に Mozilla が行ったプライバシーに関する意識調査 [2]によると，オンライントラッキングを認識しているユーザの中で，オンライントラッキングをされないためにトラッキング保護機能を利用したことのあるユーザは 22.2%に上っている．一方で，新たな Web 行動追跡技術としてブラウザから得られる複数の情報を組み合わせることでユーザを一意に識別する Browser Fingerprinting という手法が現れ，Web 広告事業者を中心に利用されつつある．

表 1 プライベートブラウジングの登場年と名称

ブラウザ	プライベートブラウジングの名称	年
Safari	Private browsing	2005
Chrome	Incognito mode シークレットモード	2008
Firefox	Private browsing	2009
IE / Edge	InPrivate mode	2009

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

本論文では，プライベートブラウジングにおける Browser Fingerprinting を用いた Web 行動追跡について，以下の 2 つの観点から調査，実験を行った．

1. 同一ブラウザ上での，プライベートブラウジングでのアクセスと通常のブラウジングでのアクセスの紐づけ
2. プライベートブラウジングのブラウザ識別と識別期間

その結果，プライベートブラウジングで収集可能な情報を用いて Browser Fingerprinting を行った場合，通常のブラウジングに近い識別精度をとることがわかった．よって，Browser Fingerprinting を用いた Web 行動追跡が行われた場合，プライベートブラウジングでは Web 行動追跡への対策としての効果が低いことがわかった．

2. Browser Fingerprinting

2.1 Browser Fingerprint

Web サーバが JavaScript や Flash などを用いることにより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よぶ．特に，UserAgent やフォントリスト，プラグインリストなど端末の特定に繋がる情報を指す．特徴点を 1 つ以上組み合わせたものを Browser Fingerprint（以降，Fingerprint とする）とよび，Fingerprint を採取し識別を行う手法を Browser Fingerprinting（以降，Fingerprinting とする）とよぶ．

2016 年 1 月に行われた最新の調査[3]によると，Alexa[4]上位 100 万サイトのうち，約 1.6%の Web サイトが Fingerprinting を行っており，上位 1000 サイトのうち約 5.1%が Fingerprinting を行っていることが判明している．

既存の Fingerprinting の研究として，Eckersley[5]は Fingerprint を収集する Web サイトを開設し，Fingerprint の分析を行った．その結果，収集したサンプルの 83.6%がユニークな Fingerprint を持ち，Flash や Java が実行できる端末に限定した場合では，94.2%がユニークな Fingerprint を持つことを示した．

Nikiforakis[6]は実際に企業が行っている Fingerprinting について調査を行い，用いられている採取手法について明らかにした．また，UserAgent の偽装を行う拡張機能を調査し，UserAgent の偽装は Fingerprinting の対策にはなりえず，新たな特徴点として利用される可能性があることを示した．

Laperdrix[7]は Eckersley[5]の研究から新たな特徴点を導入した情報収集サイトを開設し，収集したデータセットを用いてそれらの特徴点の性質を示した．

Boda[8]はフォントリストを JavaScript のみで採取する手法を提案し，Web ブラウザ間で共通のフォントリストなど一部の特徴点のみで端末を識別することによるクロスブラウザに対応した Fingerprinting を提案した．

2.2 Fingerprint 収集サイト

我々のグループでは Fingerprint 収集サイト[9]を運用し，ブラウザに関する様々な情報を収集している．また，同一ブラウザからのアクセスであることを確認するために，Fingerprint の収集に伴い，ブラウザ識別用のクッキー（以降，UID とよぶ）を生成し，Fingerprint 収集サイトのデータベースに保存している．同時に，アクセス時刻も保存している．

表 2 に我々で収集している特徴点とプライベートブラウジングでの対策状況（詳細は 3.2 節）を示す．なお，既知の特徴点についての収集方法などの詳細は先行研究[10][11][12]に記述されている．

表 2 収集可能な特徴点と各ブラウザにおける
プライベートブラウジングでの対策状況
GC : Google Chrome，FF : Firefox
T:利用可能，F:利用不可，¥N:未実装

#	特徴点	FF	GC	IE	Edge	Safari
1	JavaScript UserAgent	T	T	T	T	T
2	グローバル IP アドレス	T	T	T	T	T
3	HTTP UserAgent	T	T	T	T	T
4	プライベート IP アドレス	T	T	¥N	T	¥N
5	プラグインリスト	T	T	T	T	T
6	Canvas Fingerprint	T	T	T	T	T
	フォントリスト(Flash)	F	F	F	F	¥N
8	画面解像度と色深度	T	T	T	T	T
9	http accept language	T	T	T	T	T
10	sse2 の有無	T	T	T	T	T
11	device pixel ratio	T	T	T	T	T
12	Do Not Track	F	T	F	F	F
13	http accept	T	T	T	T	T
14	http aclscept charset	T	T	T	T	T
15	タッチ機能の有無	T	T	T	T	T
16	http origin	T	T	T	T	T
17	Refresh rate	T	T	T	T	T
18	タイムゾーン	T	T	T	T	T
19	http connection	T	T	T	T	T
20	http accept charset	T	T	T	T	T
21	local storage の利用可否	T	T	T	T	T
22	session storage の利用可否	T	T	T	T	T
23	バッテリーステータス	T	T	¥N	¥N	¥N
24	カメラとマイクの個数	T	T	¥N	T	¥N
25	物理コア数	T	T	T	T	T
26	Ad Blocker の有無	T	F	F	F	T
27	HSTS Supercookie	F	T	¥N	F	F

本論文では，表 2 の#1 から#22 までの特徴点を用いて Fingerprint とする．特に我々の先行論文[13]より Fingerprint に利用する特徴点として#12, #17 を新たに追加した. Do Not Track は，ユーザによる Web 行動追跡の拒否の申告をフラグとしてリクエストヘッダに記述するものであるが，Web サイトによってはこれを無視することも可能であり，この情報自体が特徴点となる．また，Refresh Rate はディスプレイの 1 秒間の描画回数である． 60Hz が一般的だが，一部の高性能なディスプレイと GPU を利用した場合，75Hz や 120Hz をとる．

#23 以降の特徴点は収集可能，または収集しているがサンプル数が少ない特徴点である．今回の実験では利用しないが，今後，特徴点として利用されることが予想される．

3. プライベートブラウジング

3.1 既存の関連技術

プライベートブラウジングの主要な目的はデバイスを共有して利用するユーザ間でのプライバシーを互いに守ることである. これを実現するために, クッキーや HTML5 のローカルストレージなどの保存領域に保存されたデータはプライベートブラウジングタブの終了時に削除されるようになっている.

プライベートブラウジングにおける Web のプライバシーへの対応は, 以下のようにブラウザベンダごとに異なる.

Web 行動追跡の拒否を申告する Do Not Track は，Google Chrome を除くすべての主要なブラウザで, プライベートブラウジング時にデフォルトでオンとなっている.

2012 年に考案された HSTS スーパークッキー[14]への対応もベンダごとに異なる結果となった．HSTS スーパークッキーは，Web サイトが HSTS ヘッダを送信するためのサブドメインを複数用意し，ユーザごとに異なるサブドメインへアクセスさせることで，ユニークな HSTS の識別子をブラウザに保存させる手法である．通常のブラウジング時に生成された HSTS スーパークッキーをプライベートブラウジング時に Web サイトが取得できるので, プライバシーの問題になるとされていた.

Chrome ではプライベートブラウジングでの情報が通常のブラウジングから取得できないのであれば脅威とはみなせず，HSTS を利用した際のセキュリティ上の利点が重要であるとして, 対策をとらない方針を示した[15]. Firefox や Safari では, これをプライバシーの問題であると捉え, HSTS のキャッシュをプライベートブラウジングで利用されないように実装を変更した[16][17]．なお，IE では，HSTS が実装されていないので利用できず， Edge では Firefox と同様の対策がとられた.

既存のプライベートブラウジングの研究として，Aggarwal[18]は 2010 年時点における各ブラウザのプライベートブラウジングの機能を調査し，Local DNS Cache や Flash Cookie などいくつかの対策不備を指摘し，これらを利用することで，デバイスを共有しているユーザによってアクセス履歴が取得されることを示した．また，プライベートブラウジングでの脅威について Local attacker と Web attacker の 2 つの観点で定義している．Local attacker は PC を共有するユーザの観点から，Web attacker は Web サイトから取得される情報におけるプライバシーに関する観点である.

Satvat[19]は一部のブラウザの SQLite の実装上の不備やメモリ解析によってローカルからのプライベートブラウジングのアクセス履歴の取得が可能となることを示した．また，大量のクッキーの書き込みにかかる時間の差からプライベートブラウジングからのアクセスであるかを判定する手法を提案した．Satvat の実験では，エラー率の平均である EER（Equal Error Rate）は IE では 63%と低い精度であったが，Chrome や Safari で 1%程度となり，ブラウザによっては正しく判定できる可能性がある.

3.2 プライベートブラウジングと特徴点

プライベートブラウジングでは，プライバシーを守るために機能が制限されていることがある．そこで，プライベートブラウジングで特徴点が取得できるかを主要なブラウザで確認した．ブラウザごとのプライベートブラウジングでの特徴点の対策状況を表 2 に示す．なお，ブラウザによっては特徴点を取得するための機能が実装されていない場合もあり，その実装状況も記載した.

表 2 より，プライベートブラウジングでは Fingerprinting への対策がほとんど行われていない．これは，通常のブラウジングとプライベートブラウジングの Fingerprint に差が少ないことで，高い精度の識別が行われる可能性があることを示している.

また，IE や Safari では，ブラウザとして利用できる機能が他のブラウザよりも少ないことで，利用できる特徴点が少ない.

以下，プライベートブラウジングで利用できない特徴点についてそれぞれ利用ができない理由を記述する.

- フォントリスト (Flash)

  Flash を用いて PC のフォントリストを取得するには，TextField.getFontList()と Font.enumerateFonts()を利用することができる．しかし，Adobe Flash Player 14 以降，プライベートブラウジング時にこれらの関数を用いたとき，返り値に空の文字列を返すようになった[20]．よって，プライベートブラウジング時のみフォントリストが収集できなくなった.

- Do Not Track

  Chrome 以外のブラウザでは，プライベートブラウジング時に Do Not Track をデフォルトでオンにしている．そのため，ユーザごとの差が生じなくなり，特徴点として利用できなくなる．

- Ad Blocker の有無

  Ad Blocker は AdBlock や uBlock などに代表される Web 広告をブロックする拡張機能である．Ad Blocker は一般に公開されているブロックリストを用いることが多く，ブロックされやすいファイル名を使用し，そのファイルの有無を確認することで Ad Blocker を使用しているかを判定することができる．

  Chrome，IE，Edge では，不正な拡張機能の実装によりローカルでのプライバシーの侵害が発生することへの懸念から，プライベートブラウジングで拡張機能が利用できず，AD Block の存在も確認することはできない．

4. 実験方法

本節では，Fingerprint 収集サイトで収集したサンプルを用いたプライベートブラウジング利用時のブラウザの識別に関する実験の実験方法や Fingerprinting に用いる手法，評価方法を説明する．

4.1 データセット

Fingerprint 収集サイトで 2014/9/10 から 2016/7/5 までの期間，Fingerprint の収集を行った．このうち，アクセス直後の切断など適切に収集できなかったアクセスを除いたサンプル数は 7,793，クッキーで識別したブラウザの数（UID 数）は 2,463 となった．また，複数回アクセスしたユーザは 1044，最長アクセス期間は 664 日であった．また，同一 UID 間の組み合わせ数は 44,372，異なる UID 間の組み合わせ数は 3,416,168 となった．

サンプル内でのブラウザの割合を表 3 に示す．なお，参考として StatCounter[21]が提供する日本と世界でのブラウザのシェアも記載する．

表 3 サンプル内でのブラウザの割合

ブラウザ	収集サイト	日本	世界
Chrome	30.73%	27.78%	44.83%
Safari	25.17%	27.31%	13.39%
Firefox	23.16%	9.84%	10.33%
IE	5.89%	25.70%	10.91%
Opera	0.81%	0.89%	4.88%
Edge	0.73%	0.56%	0.45%
Other	13.50%	7.91%	15.21%

4.2 Fingerprinting における識別手法

本論文では，Fingerprint は表 2 の#1 から#22 までの各特徴点を文字列として連結して生成する．また，Fingerprint の識別を行うアルゴリズムには先行研究[13]で示した Fuzzy Hashing を用いる．Fuzzy Hashing は与えられた文字列を部分文字列に分けてハッシュ化する手法である．2 つのサンプルのハッシュ値から類似度を算出することで，高速な文字列比較を行うことが可能となる．識別の実現方法として，Fuzzy Hashing を実装した ssdeep[21]の Python ラッパーを用いて Fingerprint の比較を行う．このライブラリは容易に入手することができ， Web 行動追跡に利用されることが懸念される．

4.3 識別精度の評価方法

まず，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．実験では，UID が一致しているサンプルにおいて，Fingerprint が一致とみなせたとき，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．実験では，UID が不一致となったサンプルにおいて，Fingerprint も不一致とみなせたとき，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．実験では，Fingerprint が不一致とみなせたとき，サンプルの UID も不一致となれば，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，ペナルティ値の総和が閾値より大きい場合に比較したサンプルの UID が同一であれば，比較結果は FN となる．

特に TPR（True Positive Rate）と FPR（True Positive Rate）は以下の式で求まる．

$$TPR = \frac{|TP|}{|TP + FN|} \quad (1) \qquad FPR = \frac{|FP|}{|FP + TN|} \quad (2)$$

本論文では類似度を用いて精度を算出するので，それぞれの閾値によって TPR，FPR は異なる．そこで，ROC（Receiver Operating Characteristic）曲線と AUC（Area Under the Curve）を用いて，精度の評価を行う．

ROC 曲線は FPR を x 軸，TPR を y 軸にとり，それぞれの閾値における結果をグラフに描画することで得られる曲線である．AUC は ROC 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．

4.4 実験方法

実験は LOOCV（Leave-One-Out Cross Validation）で行う．データセットの 1 つのサンプルをテストサンプルとし，それを除くすべてのサンプルに対して，それぞれ同一サンプルであるかを Fingerprint で識別する．識別の成否は UID を用いて判定する．これをすべてのサンプルで繰り返し，識別精度を求める．

プライベートブラウジングの識別について実験を行うため，3.3 節で示したプライベートブラウジングで対策されている#7，#12 の特徴点を除いた Fingerprint（以降，Private Fingerprint とする）を生成する．また，比較として通常のブラウジングで収集できる特徴点を用いた Fingerprint を Public Fingerprint とする．

プライベートブラウジングと通常のブラウジングの識別精度を算出し，これらの精度を比較することでプライベートブラウジングが Fingerprinting への対策として効果があるかを確認する．

5. 実験

5.1 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 （3）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．

$$\mathrm{NE} = \frac{\mathrm{H}(X)}{\log_2(N)} \quad (3)$$

各特徴点の NE の比較を表 4 に示す．表 4 の特徴点は NE を降順にソートしている．なお，表 2 で示されたプライベートブラウジングで対策されている特徴点については背景を灰色にしている．なお，通常のブラウジングで利用できる#1~22 の特徴点を用いた Fingerprint（以降，Public Fingerprint とする）についてもエントロピーを算出した．

プライベートブラウジングでは#7，#12 の特徴点に対して対策がとられている．特に#7 は NE が 0.41 と比較的大きな値となっているので，識別への影響が大きいと考えられる．

表 4 特徴点ごとの Normalized Shannon's Entropy
(灰色はプライベートブラウジングで対策がとられている)

#	特徴点	NE
1	JavaScript UserAgent	0.687314
2	グローバル IP アドレス	0.677292
3	HTTP UserAgent	0.669057
4	プライベート IP アドレス	0.599952
5	プラグインリスト	0.493429
6	Canvas fingerprint	0.442459
7	フォントリスト(Flash)	0.416889
8	画面解像度と色深度	0.332312
9	http accept language	0.257447
10	sse2 の有無	0.197248
11	device pixel ratio	0.17076
12	Do Not Track	0.13732
13	http accept	0.116583
14	http accept encoding	0.103371
15	タッチ機能の有無	0.07886
16	http origin	0.056785
17	Refresh rate	0.025357
18	タイムゾーン	0.017652
19	http connection	0.00388
20	http accept charset	0.003111
21	local storage の利用可否	0.862171
22	session storage の利用可否	0.332312
28	Public Fingerprint	0.954549
29	Private Fingerprint	0.949171

5.2 識別精度

通常のブラウジング時とプライベートブラウジング時のそれぞれにおける Fingerprinting の識別精度を ROC により図 1 のとおり示す．また，このときの AUC も示す．

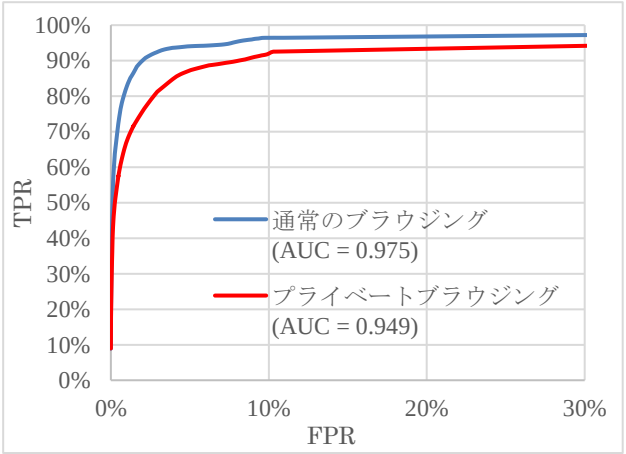

図 1 プライベートブラウジングと通常のブラウジングの ROC 曲線と AUC

図 1 より FPR が 5%のとき，TPR は通常のブラウジングでは 94.1%，プライベートブラウジングでは 87.3%となった．また，FPR を 10%許容した場合の TPR は通常のブラウジングでは 96.4%，プライベートブラウジングでは 92.5%となった．プライベートブラウジングでは，一部の特徴点が利用できないことで識別精度がわずかに低下するものの，通常のブラウジングに近い値をとることがわかる．また，AUC の値からも通常のブラウジングとプライベートブラウジングの差が少なく，プライベートブラウジングでの識別にほとんど影響がない．以上より，プライベートブラウジングで利用できる特徴点のみを用いた Fingerprinting においても，識別できる可能性が高いといえる．

5.3 期間ごとの識別精度

Fingerprint に用いられる特徴点は User Agent やプラグインリストなどバージョンアップに伴い，変化することがある．そこで，アクセス間隔によってサンプルを分割し，アクセス期間ごとの識別精度を算出した．

期間は 1 日未満，1~7 日，以降 1 週間ごとの間隔で 56 日までを分類した．期間の分類は 2 つのサンプルでの比較の際に時刻差を算出し，定めた期間ごとに分類した後，識別精度を算出した．

それぞれの期間における AUC を表 5 に示す. 表 5 より，約 1 か月となる 28 日間アクセス期間をあけた場合においても，0.969 となり，アクセス期間を開けたとしても AUC は大きく下がらないことがわかった．

5.4 プラベートブラウジングと通常のブラウジングの紐づけ

次にプライベートブラウジングでのアクセスと通常のブラウジングでのアクセスの紐づけの可能性を確認した．LOOCV でのテストサンプルで Public Fingerprint を用い，Private Fingerprint のサンプルに対して識別精度を算出することで，特徴点の有無による紐づけの可能性を確認した．紐づけの実験結果の ROC 曲線と AUC を図 2 に示す．

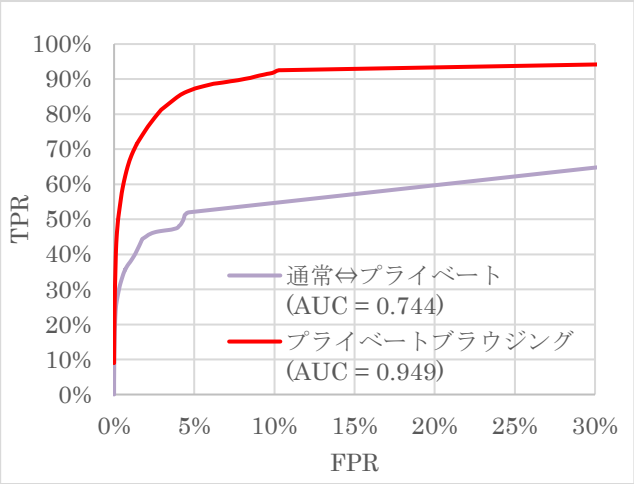

図 2　紐づけによる ROC 曲線と AUC

図 2 より，紐づけの実験では FPR が 5%のとき，TPR が 52.2%，FPR が 10%のときでも，TPR が 54.7%と比較的低い値をとり，Public Fingerprint を用いてプライベートブラウジングのサンプルと紐づけることは難しいことが分かった．

5.2 節で示したようにプライベートブラウジングで利用が可能な特徴点での Fingerprinting は高い精度で識別を行うことができる．このため，プライベートブラウジングでのアクセスと通常のブラウジングでのアクセスの紐づけは，プライベートブラウジングで利用できない特徴点を Fingerprint に含まない場合にのみ，Fingerprinting で Web 行動追跡が行われる可能性がある．

5.5 ブラウザごとの識別精度

3.3 節に示した通り，ブラウザごとに特徴点への対応状況が異なる．そこで，プライベートブラウジングにおけるブラウザごとの識別精度を確認した．ブラウザの分類は UserAgent を用いて行った．ブラウザごとの識別精度を図 2 に示す．ただし，今回の実験サンプルにおいて，Edge や Opera などのブラウザについてはサンプル数が少ないので，ここでは用いない．

表 5　プライベートブラウジング，通常のブラウジングにおける期間ごとの AUC

ブラウジング		期間(日)									
		0~1	1~7	7~14	14~21	21~28	28~35	35~42	42~49	49~56	56~
通常のブラウジング		0.9879	0.9825	0.9778	0.974	0.9707	0.9805	0.976	0.9695	0.9713	0.9541
プライベートブラウジング		0.9663	0.9033	0.9620	0.909	0.9573	0.9697	0.9727	0.9628	0.9606	0.9233
組み合わせ数	同一 UID	5671	5086	5286	4808	3510	3016	2418	2373	1876	10328
	異なる UID	47686	88375	116101	111106	107145	104148	99510	95609	90220	2556268

図 3 より，Firefox が最も高い識別精度となった．Chrome は識別精度が最も低くなった. Chrome の精度の低下はプライグインリストやブラウザのバージョンアップが他のブラウザよりも頻繁に行われることが原因であると考えられる．例えば，Chrome Stable のバージョン 51 から 52 までには，4 回のマイナーバージョンアップが行われており，最短のバージョンアップは 5 日間であった．また， Canvas Fingerprint に関してもブラウザのバージョンアップに伴い，結果が変わることがあり，精度の低下へ大きく影響していると考えられる．

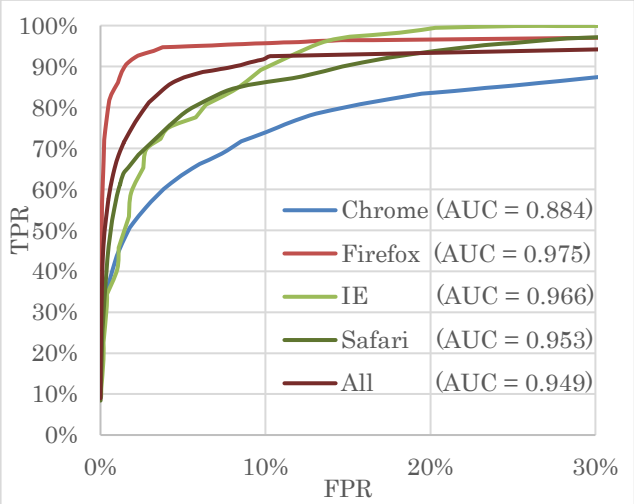

図 3 プライベートブラウジングにおける
ブラウザごとの ROC 曲線と AUC

6. 議論

6.1 考察

対策をされているフォントリストについては，JavaScript と CSS を組み合わせた手法[23]や CSS のみで収集する手法[24]が既に知られており，それらの手法を用いることで，収集方法の違いから NE 値の減少が免れないものの，類似したフォントリストを得ることができる．

本論文では収集サンプルの都合上，利用しなかった特徴点が複数ある．現状の高い識別精度に加えて，これらの特徴点を Fingerprint に含めた場合，より高い精度となることが見込まれる．また，実験ではクッキーで成否を判定していることから，クッキーを削除して再度アクセスするユーザなどにより，実際に識別する場合よりも精度の低下が生じていることが考えられる．

モバイルデバイスにおけるプライベートブラウジングでは，Do Not Track 以外の特徴点に変化は起きなかった．これは，モバイルではデフォルトで Flash が利用できず，フォントリストが収集されないため生じた．モバイルのみで同様の実験を行う場合，さらに近い精度になることが考えられる．

6.2 Web におけるプライベートブラウジングでの脅威

本節では，プライベートブラウジングの脅威について Aggarwal[18]が定義した Web attacker の観点を用いて議論する．

Aggarwal の論じた Web attacker によるプライベートブラウジングの脅威は以下の 3 点から構成されている．

1. プライベートブラウジングと通常のブラウジングのアクセスの紐づけ
2. プライベートブラウジングの行動追跡
3. プライベートブラウジングでのアクセスの判定

これらの観点から，現状で対策が行えているかをそれぞれ確認する．

6.2.1 プライベートブラウジングと通常のブラウジングの紐づけ

プライベートブラウジングで閲覧したアクセス履歴が通常のブラウジング時に意図せず漏えいすることは，Web プライバシーの観点から好ましくない．

5.4 節の実験により，収集できる特徴点をすべて用いて Fingerprinting を行った場合は，紐づけを行うことは難しい．ただし，5.2 節が示すようにプライベートブラウジングで利用可能な Fingerprint のみを用いて Fingerprinting が行われた場合には，紐づけを行える可能性が高い．

6.2.2 プライベートブラウジングの行動追跡

プライベートブラウジングはセッション終了時に識別子となる情報のほとんどを削除している．そのため，プライベートブラウジングを行う度にユーザのアクセスは異なるユーザのアクセスであると認識されるべきである．

5.3 節より，Fingerprinting を用いることで，プライベートブラウジングにおいても長期的に高い精度での識別が可能となってしまうことが示された．

6.2.3 プライベートブラウジングでのアクセスの判定

3.3 節より，プライベートブラウジング時に Flash を用いてフォントリストを取得することで，空文字が取得できる．この仕組みを用いて，フォントリストを取得する関数からの返り値が空文字のとき，プライベートブラウジングのアクセスであると特定することができる．ただし，設定で Flash を無効にしている場合，関数自体の呼び出しができないため，プライベートブラウジングの判定はできない．

プライベートブラウジングにおける Indexed DB や Local Storage などへのデータ保存への対策から，通常のブラウジングと挙動が異なり，その差異からプライベートブラウジングの判定を行う手法も存在する[25]. ただし，この手法では Tor ブラウザなどの一部ブラウザにおいてもプライベー

トブラウジングと判定される．

Satvat[19]が提案したタイミングアタックを利用した判定手法により，一部のブラウザではプライベートブラウジングでのアクセスが正しく判定される可能性がある．

7. まとめ

本論文では，Browser Fingerprinting によりプライベートブラウジングのアクセスが追跡される可能性が高いことを示した．プライベートブラウジングに Web 行動追跡への対策を導入するブラウザベンダは可能な限り Fingerprinting への対策を行うことも必要となる．

また，ユーザはプライベートブラウジングを利用していた場合においても Web 行動追跡が行われる可能性があることを認識して利用し，プライバシーへの対策には Tor Browser など，より対策が進んだブラウザを使うことが望ましい．

参考文献

[1] “Tracking Protection in Private Browsing”. https://support.mozilla.org/en-US/kb/tracking-protection-pbm, (参照 2016-08-09)

[2] “プライバシーに関する意識調査 調査結果 - Mozilla Japan”. https://www.mozilla.jp/static/docs/press/DPD2016-survey-Results.pdf, (参照 2016-08-09)

[3] “Online tracking: A 1-million-site measurement and analysis”. https://webtransparency.cs.princeton.edu/webcensus/index.html, (参照 2016-08-09)

[4] “The top 500 sites on the web”. http://www.alexa.com/topsites, (参照 2016-08-09)

[5] P. Eckersley, How Unique is Your Web Browser? In Proc. Privacy Enhancing Technologies Symposium (2010), LNCS vol.6205, 2010

[6] 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.

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

[8] K Boda, A Foldes, G Gulyas, S Imre, User Tracking on the Web via Cross-Browser Fingerprinting , in Proc. of 16th Nordic Conference on Information Security Technology for Applications, 2011.

[9] “明治大学 情報セキュリティ研究室| Fingerprinting Research”. https://www.saitolab.org/fingerprint, (参照 2016-08-09)

[10] 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 論文集 p.378-p.385

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

[12] 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 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 674-689, 2014.

[13] 石川貴之，高須航，山田智隆，武居直樹，細井理央，安田昂樹，高橋和司，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 ,2015

[14] “HSTS Super Cookies”. http://www.radicalresearch.co.uk/lab/hstssupercookies, (参照 2016-08-09)

[15] “Security: HSTS "cookies" do not obey expected policy”. https://bugs.chromium.org/p/chromium/issues/detail?id=104935, (参照 2016-08-09)

[16] “HSTS state can track users, follows them in to private browsing mode”. https://bugzilla.mozilla.org/show_bug.cgi?id=930638, (参照 2016-08-09)

[17] “Expose time-based HSTS clearing”. https://bugs.webkit.org/show_bug.cgi?id=133161, (参照 2016-08-09)

[18] G Aggarwal, E Bursztein, C Jackson, D Boneh, An analysis of private browsing modes in modern browsers. Proceedings of the 19th USENIX Security Symposium. 2010.

[19] K Satvat, M Forshaw, F Hao, E Toreini On the privacy of private browsing–a forensic approach. In Data Privacy Management and Autonomous Spontaneous Security, 2014

[20] “Release Notes | Flash Player® 14 AIR® 14”. https://helpx.adobe.com/flash-player/release-note/fp_14_air_14_release_notes.html, (参照 2016-08-09)

[21] “StatCounter Global Stats”. http://gs.statcounter.com/ , (参照 2016-08-09)

[22] “ssdeep”. http://ssdeep.sourceforge.net/, (参照 2016-08-09)

[23] “JavaScript/CSS Font Detector”. http://www.lalit.org/lab/javascript-css-font-detect/, (参照 2016-08-09)

[24] N Takei, T Saito, K Takasu, T Yamada, Web Browser Fingerprinting Using Only Cascading Style Sheets, 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA). IEEE, 2015.

[25] “Detect private browsing mode”. https://gist.github.com/cou929/7973956, (参照 2016-08-09)