

# ユーザトラッキングを防ぐための 連動して動作する Web ブラウザの提案

境 柊亮<sup>1</sup> 新城 靖<sup>1</sup>

## 1. はじめに

現在、インターネットが普及し、多くの人々が毎日様々な Web ページを閲覧している。Web サイトはユーザトラッキング、すなわち、様々な手法を用いてユーザを追跡し情報を収集している。その情報はターゲット広告などに利用されるが、情報がどのように利用されているかをユーザが確実に知る方法がなく、ユーザのプライバシーが侵害されている可能性がある。近年では、実行環境やブラウザ特有の値 (Browser Fingerprint) を使用してユーザトラッキングを行うという手法も利用されている。

本研究室では、ユーザトラッキングを検知するために双子の Web ブラウザを提案した [1]。このブラウザは親ブラウザと 2 つの子ブラウザで構成され、2 つの子ブラウザにおいて親ブラウザ上で行われた操作と同一の操作が行われる。このブラウザでは、2 つの子ブラウザから発生する HTTP リクエストを保存し比較することでユーザを識別するための情報 (識別子や属性) を検出する。しかし、この研究ではトラッキングの防止を行うことができなかった。

そこで本研究では、双子の Web ブラウザの考え方を発展させ、Web ページから得られる情報を維持したままユーザトラッキングを防止することを目指す。本研究では双子の Web ブラウザと異なり、子ブラウザを 1 つだけ使用する。そして、親ブラウザと子ブラウザの HTTP リクエストを比較し、差が存在した場合には削除もしくはランダム化を行う。これにより、ユーザトラッキングに使用される情報が送信されることを防ぎ、トラッキングを防止する。

## 2. ユーザトラッキングの手法

現在、利用されているユーザトラッキングの手法として Cookie を利用したものと Fingerprint を利用したものがある。Cookie を利用したものでは Cookie に識別子を保存しそれをもとにユーザを追跡する。Fingerprint を利用したものでは、ブラウザを実行している環境やブラウザに特有の属性を組み合わせることで Fingerprint としユーザを追跡する。組み合わせる属性として、30 種類以上知られており例として利用可能なフォント、画面解像度、ユーザエー

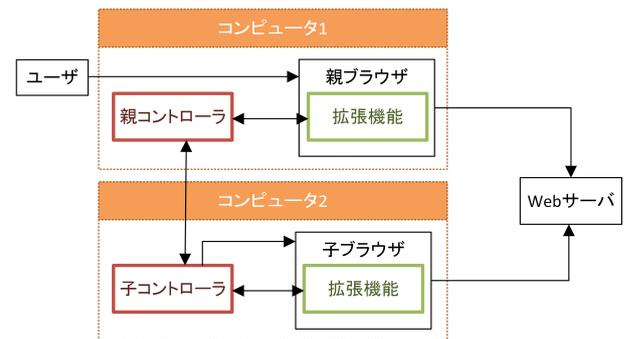


図 1 連動して動作する Web ブラウザの構成

ジェントなどが存在する [2]。

## 3. 連動して動作する Web ブラウザの設計と実装

図 1 に本研究で実装するブラウザの構成を示す。本研究で実装するブラウザでは図 1 で示されている通り親ブラウザと子ブラウザを異なるコンピュータ (環境) で動作させる。Web サーバに送信されるリクエストの中から親子で異なる情報を削除もしくはランダム化する。また、同一のブラウザを基にした場合や OS が同じ場合、ブラウザや OS に関する情報が送信される可能性があるため異なる OS、ブラウザを組み合わせる。

### 3.1 子ブラウザにおける操作の再現

親ブラウザでは、行われた操作を取得するために、Web ページの Document Object Model ノードにイベントハンドラを登録する。子ブラウザでは、親ブラウザでなされた操作を、Selenium<sup>\*1</sup>を利用して再現する。再現する操作には、クリック、キーボードからの入力、ウィンドウのリサイズなどがある。基本的に親ブラウザで行われた操作と同じ操作を子ブラウザで行うが、ウィンドウのリサイズのみ、親ブラウザのウィンドウのサイズから 1px 引いたサイズにリサイズを行う。これにより、親子で差が生まれるようにし差の削除やランダム化を可能にする。

<sup>1</sup> 筑波大学

<sup>\*1</sup> <https://www.selenium.dev/>

### 3.2 リクエストの書き換え

親子の各ブラウザではユーザが Web ページを閲覧しようとしても、すぐには Web サーバにリクエストを送らせず、それをコントローラに通知する。そして、コントローラからのリクエスト修正の指示を待つ。

親コントローラでは各ブラウザから送信されてきたリクエストの以下の項目を比較する。

- Request URL の query string
- ヘッダ (Cookie と Content-Length は別に処理)
- ボディ (x-www-form-urlencoded と JSON)

親コントローラは、差を見つけると、それを削除するか、または、ランダム化することを決定し、各ブラウザに指示する。また、リクエストボディが対応していない形式の場合には比較のみを行い、異なる場合にはリクエストをブロックするように各ブラウザに指示する。

各ブラウザはコントローラから指示を受け取ると、リクエストのブロック、およびリクエストの内容の書き換えを行う。

## 4. 実験

提案手法を Web ブラウザ Firefox<sup>\*2</sup>で実装した。これを 2 台のコンピュータで実行し実験を行った。コンピュータ 1 は Linux Ubuntu 20.04LTS, コンピュータ 2 は Oracle VM Virtual Box<sup>\*3</sup>を利用した仮想環境で Windows 10 を実行した。差を処理する方法としては削除かランダム化が選択できるが今回は削除を行った。

### 4.1 Google での検索

多くの人が利用しており、ユーザトラッキングが行われていると考えられる <https://www.google.com/>へのアクセスを行った。検索を行ったところ通常とユーザインタフェースは異なるが、同じ検索結果が表示された。このことから得られる情報は同じであったと言える。また、このアクセスを行った際に発生する [https://www.google.com/client\\_204](https://www.google.com/client_204) へのリクエストを記録し、通常の Firefox を用いてアクセスした場合の結果と比較を行った。client\_204 へのリクエストを比較しているのは、最初のリクエストではブラウザ側にデータが保存されておらず、通常の場合とのリクエストの差が HTTP ヘッダの User-Agent に限られるためである。

送信されたリクエストを比較すると、文献 [2] で列挙された属性の中で HTTP ヘッダの User-Agent が削除されていることが確認された。また、文献 [2] で列挙された属性の中には存在しないが query string 中からユーザトラッキングでしばしば利用される、ウィンドウサイズであると考えられる biw や bih が削除された。文献 [2] で列挙された

属性のうち、削除されなかった属性としては HTTP ヘッダの Accept, Accept-Language, Accept-Encoding がある。Accept は親子のブラウザの種類を変更すると削除される。Accept-Language は、親子のブラウザで言語設定を変更すると削除できる。しかしそれで表示内容まで変化してしまう。

### 4.2 Youtube での動画視聴

Youtube で動画を再生する実験を行った。その結果、動画を再生することはできなかった。この原因としてこのサイトではアクセスごとに動画の URL を変化させていることが考えられる。

## 5. 関連研究

PriVaricator は Browser Fingerprint を利用したユーザトラッキングを防止するために、Fingerprint に使われる属性の値を変化させる [3]。PriVaricator では Fingerprint に利用される属性が増えることに対応する必要があるが、本研究ではそのような必要はない。

## 6. まとめ

本研究では、得られる情報を維持したままユーザトラッキングを防止することを目的として、親ブラウザと子ブラウザという連動する 2 つの Web ブラウザを利用したユーザトラッキングの防止を提案し、それを実装する。

現在までに各コントローラと Firefox をもとにした各ブラウザの実装を完了し実験を行った。今後、異なる種類のブラウザで提案手法を実装し、評価を行う。

## 参考文献

- [1] 張 世申, 新城 靖, 三村賢次郎: 個人情報及び個人識別子を含むファイルと通信を検出するための双子の環境, 情報処理学会第 30 回コンピュータシステム・シンポジウム (ComSys2018), pp. 29–36 (2018).
- [2] Datta, A., Lu, J. and Tschantz, M. C.: Evaluating Anti-Fingerprinting Privacy Enhancing Technologies, *The World Wide Web Conference, WWW '19*, p. 351–362 (online), DOI: 10.1145/3308558.3313703 (2019).
- [3] Nikiforakis, N., Joosen, W. and Livshits, B.: PriVaricator: Deceiving Fingerprinters with Little White Lies, *the 24th International Conference on World Wide Web, WWW '15*, p. 820–830 (online), DOI: 10.1145/2736277.2741090 (2015).

<sup>\*2</sup> <https://www.mozilla.org/en-US/firefox/>

<sup>\*3</sup> <https://www.virtualbox.org/>