# Survey of Existing Fingerprint Countermeasures

Sakchan Luangmaneerote
Electronic and Computer Science
University of Southampton
Southampton, SO16 7FB, UK
sl8e14@soton.ac.uk

Ed Zaluska and Leslie Carr
Electronic and Computer Science
University of Southampton
Southampton, SO16 7FB, UK
ejz@ecs.soton.ac.uk

***Abstract*— A number of fingerprint countermeasures have claimed that their countermeasures can prevent the user away from fingerprint tracking. The research attempts to prove their claims and requires to study the effectiveness of their fingerprint prevention and observe the results of side effects after defence in order to assist the attentive user to know the limitation of current approaches. Under investigation, all countermeasures will be installed on the web browser and visit the developed hybrid fingerprint website in order to know the efficiency of the fingerprint resistance of all types. The research shows that all fingerprint countermeasures nowadays are unable to obstruct all kinds of the fingerprint tracking and countermeasures that use the blocking technique appear more side effects to the web browser than other techniques. Also, the increasing number of fingerprint attributes are increasing cause of unusual combination inside the Internet browser.**

***Keywords; Browser Fingerprinting, Privacy, Web Tracking, Fingerprint Countermeasure, Hybrid Fingerprint***

## I. INTRODUCTION

This paper investigates how the existing fingerprinting countermeasures work and what is the primary limitation of current countermeasures. The initial investigation is divided into five substantial parts. Firstly, the overall efficiency of current prevention techniques is studied and then evaluate how many types of fingerprinting are blocked from fingerprint tracking. This section has developed the hybrid fingerprint website which consists of JavaScript object, plugin, font and canvas fingerprint to observe the experience of fingerprint resistance. Secondly, it studies the side-effects of fingerprinting prevention by visiting sites and then find the overall satisfaction of the use of users. The results of side-effects are shown in this section. Thirdly, it studies in-depth of fingerprinting attributes to find how the value of fingerprinting attributes is changed after rendering the web page. This part discusses that why the changed fingerprinting attributes have influenced to the display of the internet browser. Fourthly, this section trials the behaviour of the web browser if the fingerprinting attributes are modified while rendering a web page in order to find the cause of adverse effects of the internet browser. This part builds a Chrome extension to intercept web page before loading and then inject modified fingerprinting attributes into the Internet page and observing the result of side-effects. Fifthly, the existing countermeasures will be studied that they have introduced the web browser stand out or not. Finally, all results will be concluded which countermeasure probably used and propose a future solution.

## II. RELATED WORK

Identifying the individual web browser being used is a real possibility [1] as the browser's attributes that usually utilized in the web browser can be applied for identification of the Internet web browser. The various browser's attributes that often use while surfing websites (e.g., navigator and screen object, etc.) will be concatenated into the string and then calculated with the hash functions. The hash ID created will perform as a tracker to track the user across the Internet. This created tracker is unnecessary to leave any files on the user computer which most users are unaware that they are being monitored with fingerprint technique as they cannot notice any files on their computers. Later, it can apparently claim that this approach could identify the web browser with an accuracy of 96% by Eckersley [2], a man who built the Panopticlick project. He had invited general users to test the ability of fingerprinting by visiting his website. Also, he found that list of fonts provides the influential factor to assist the fingerprinting easily (obtained fonts through using plugins). He has calculated the entropy of fonts 13.9 bits of entropy, which mean that if he only uses one attribute to create the fingerprint tracker, it has only one in 15,286 similar browsers will be found on the web Internet. It can be seen that only one list of fonts is distinctive enough to create a unique tracker. Also, he had demonstrated that a simple heuristic program could predict the changing of browser's attributes with a 94% accuracy. From these results, it can be clearly seen that fingerprinting technique can track users in practice without the doubt. The significant impact of the Panopticlick project has inspired several subsequent papers. Later on, Boda [4] demonstrated that a list of fonts could be gained using JavaScript without using of the browser's plugins. They combined a list of fonts, the IP address, time zone, and screen resolution which can identify the user's web browser with a high level of success. This means that fingerprinting is not only relying on the browser's plugin but can also use JavaScript like the alternative plan to obtain the list of fonts. Later, the canvas element in HTML5 is one attribute to be used fingerprinting [5], called the "canvas fingerprinting". This attribute will write and read an image while rendering the web page. As the value of a retrieved image provides a unique

characteristic of the user operating system that is sufficient enough to be used for identification of a web browser. In the same year, the browsing history on the Internet browser is implemented like a new tracker to track the web browser [6], which can show detection accuracy of 69%. However, this technique is not practical for the latest version of the modern web browser. Other methods are not mentioned as there are several ways to fingerprint the web browser such as packet ordering information [7], combining IP address and UserAgent [8] and mobile fingerprinting [9] audio fingerprinting [10] which almost all come from the inspiration of Panopticlick project.

## III. COUNTERMEASURES

Several approaches have proposed the possible solutions to inhibit the fingerprint tracking. The "Do not Track" (DNT) header is proven [13] that user cannot trust this technique as most websites can fingerprint a web browser by regardless of the DNT value. Tor [14] is widely acknowledged as it helps some users who require to remain anonymous online. Tor has modified various attributes in the Firefox web browser in order to protect users away from the fingerprint tracking. Tor not only is used against fingerprint technology but also is designed for different objectives such as protecting source online, keeping secret information of the company and so on. The research had collected the different defensive fingerprint countermeasures as Table 1 below.

TABLE 1 EXISTING COUNTERMEASURES

| Countermeasure | Platform |
|---|---|
| Tor [13] | Running on Tor browser |
| Rubberglove [15] | Running on Chrome |
| Chameleon [16] | Running on Chrome |
| CanvasFingerprintBlock [18] | Running on Chrome |
| ChromeDust [21] | Running on Chrome |
| StopFingerprinting [20] | Running on Chrome |
| UserAgent Switcher for Chrome [21] | Running on Chrome |
| Canvas Fingerprinting Blocker [22] | Running on Firefox |
| FireGloves [23] | Running on FireFox |
| FP-Block [17] | Running on FireFox |
| Stop fingerprinting [24] | Running on FireFox |

While few papers (e.g., FPGuard [11] and Privaricator [12]) claims that their countermeasures can keep user away from fingerprint tracking, these unproven allegations are more difficult to evaluate whether their model can prevent better or not as the status of software has not been made available. Therefore, this research had mainly focused only on existing approaches which can now download through the web Internet.

## IV. TESTING

In this section, the current fingerprint countermeasures had been investigated whether they could stop collecting user data from the fingerprint companies. In order to estimate the potential impact of using existing countermeasures, the hybrid fingerprinting site [3] had been developed specifically for investigating the efficiency of fingerprinting prevention. This site had also studied the value of fingerprinting attributes of the web browser while rendering a web page as a real fingerprint website by using font, plugin, canvas and JavaScript object fingerprint. Also, the proper fingerprint prevention is not sufficient enough for the user who needs the potential countermeasure to protect their privacy. Therefore, all spectrum of user experience had been raised to consider the user satisfaction while they are browsing a web page. This section will be organised into five substantial portions, testing prevention, testing side-effects of prevention, studying fingerprinting attributes, studying the behaviour of fingerprinting attributes and studying effects of information paradox.

### *A. Testing prevention*

As for this part, it will verify the existing fingerprint countermeasures in Table 1, mostly available for free on the web Internet. All fingerprint countermeasures had been installed in the web browser, except Tor browser, and then visit the hybrid fingerprint website in order to learn from experience of visiting website, results as Table 2 below.

TABLE 2 TYPE OF PREVENTION

| Countermeasure | Object JavaScript (navigator, screen) | List of fonts | List of plugins | Canvas |
|---|---|---|---|---|
| Tor | √ | * | √ | - |
| RubberGlove | √ | - | √ | - |
| Chameleon | ≈ | - | - | ≈ |
| CanvasFingerprintBlock | - | - | - | ≈ |
| ChromeDust | - | - | - | - |
| StopFingerprinting | - | - | - | - |
| Canvas Fingerprinting blocker | - | - | - | √ |
| FireGloves | √ | √ | √ | - |
| FP-Block | √ | - | √ | - |
| Stop Fingerprinting | - | √ | √ | - |
| UserAgent Switcher to Chrome | - | - | - | - |

( √ ) The countermeasure can prevent fingerprint by blocking, spoofing or randomization
(*) The countermeasure prevents by entropy limitation
(-) The countermeasure allows fingerprinting
≈ The countermeasure can detect fingerprint but cannot prevent fingerprint tracking.

Considering from the table 2 above, Tor and Fireglove can prevent three types of fingerprinting when compared with other countermeasures. RubberGlove, Stop fingerprinting and FP-

block can prevent two types of fingerprinting. As CanvasFingerprintBlock and canvas fingerprinting can only prevent one fingerprinting (because they are designed to address only one type of fingerprinting). The remaining countermeasures (Chameleon, ChromeDust and StopFingerprinting) do not appear to protect fingerprint tracking as they claim. Thus, it can observe that most countermeasures on the web Internet are unable to prevent all types of fingerprinting, and in particular, some countermeasures cannot inhibit user tracking as they claim.

### *B. Side effects of prevention*

This section studies adverse impacts of existing countermeasure to the popular web browser by selecting only existing countermeasure that can prevent user tracking from the result of Table 2. The study will use each current countermeasure to visit five websites in order to observe the effects of fingerprint prevention, namely: facebook.com, youtube.com, google.com, and bbc.com. Note that some popular sites in the top 100 list do not use English languages which make it difficult to determine whether the site is checking it works correctly. For this reason, the research only selects the English-language website.

TABLE 3 NEGATIVE EFFECTS TO THE WEB BROWSER

| Countermeasure | Problem of display | Problem of functionality | Difficult to use | Login problem |
|---|---|---|---|---|
| Tor | √ | √ | √ | √ |
| RubberGlove | √ | √ | √ | - |
| CanvasFingerprintBlock | - | - | - | - |
| Canvas Fingerprinting | - | - | - | - |
| FireGloves | √ | √ | √ | - |
| FP-Block | - | √ | - | - |
| Stop Fingerprinting | - | √ | - | - |

**Problems of display:** Content, fonts or screen size are changed.
**Problem of functionality**: The video and music do not play, or some functionalities are unavailable.
**Difficult to use**: Rendering a webpage is a slow process, makes user annoying, and is unsmooth.
**Login Problem**: The user faces the login problems.

Considering the results from Table 3, Tor seemed to have several problems, particularly the login problem and challenges of the Internet speed. As RubberGlove and Fireglove had the same challenges as well as Tor, except for login problems due to as unchanged location of a user. CanvasFingerprintBlock and Canvas Fingerprinting appeared to have insignificant adverse effects. FP-block and Stop Fingerprinting seemed the slightest problem. However, the fingerprinter might know that only list of fonts and plugins were modified. They might use remaining attributes to fingerprint (e.g., userAgent and canvas). Thus, FP-Block and Stop Fingerprinting appeared few adverse effects which test result might attract the attentive users who are looking for a suitable approach to block fingerprint technique.

### *C. Study fingerprinting attributes*

The result of the previous section is the great benefit to investigate further whether which fingerprinting attributes modified probably effects to the Internet browser. The further investigation had been done in this chapter for analysis in-depth in order to find which attribute introduce the problems of the user browsing experience. The research had selected the fingerprint attributes which were mainly used from many fingerprintes to show how many attributes are handled in each fingerprint countermeasure. Each countermeasure was arranged to visit hybrid fingerprint website once in order to check all fingerprinting attributes which the result is shown in Table 4. The result of Table 4 was analysed along with the table 3 so as to find the answer why the web browser that was installed why they cannot perform correctly. Also, this section will not consider the canvas fingerprint countermeasure as this attribute does not introduce any side-effects to the Internet browser.

As for the result from table 4, each countermeasure handled fingerprinting attributes differently. They only use three basic techniques to prevent fingerprint tracking, blocking, randomising, and spoofing technique. In addition, they do not handle all fingerprinting attributes as they were unaware of which browser attributes were used by fingerprinters. Thus, which of most mainly focus on the high entropy characteristics (userAgent, the list of fonts, plugins) which are sufficient enough to deceive the tracking of fingerprinting technique. Considering Tor, Rubberglove and FP-Block, they appear to deal with more fingerprinting attributes than other methods; they use spoofing and blocking technic. Even though they controlled fingerprinting attributes alike, but the adverse effects of Tor browser was opposite with FP-Block. It might be that Tor modifies more browser's attributes than the general fingerprinting attributes shown in Table 4. FP-block only altered fingerprinting attributes in Table 4 that it was not showing that they had produced the profound impact to the web browser. RubberGlove dealt with 18 fingerprinting attributes which mainly use blocking technique. As for table 3 and table 4, it can be seen that blocking technology is the profound impact to the modern web browser. As FireGloves handled eight fingerprinting attributes and mainly used blocking technique, the result of adverse effect is similar to the RubberGlove, rather the significant impact to the web browser. Stop Fingerprinting deal with two fingerprinting attributes which it shows the insignificant effect on the internet browser. Dealing with few attributes might be a good idea, but remaining of fingerprint attributes might be fingerprinted.

To sum up, the existing countermeasures that use randomization and spoofing technique seem to produce minimum effects to the internet browser. The increasing number of fingerprinting attributes is not confirmed that it associated with the adverse effects to the web browser while blocking technique is the profound impact to the web browser.

TABLE 4 METHOD TO HANDLE FINGERPRINTING ATTRIBUTES

| | Fingerprinters | | | | |
|---|---|---|---|---|---|
| **Properties** | **Tor** | **Stop fingerprinting** | **Fireglove** | **Rubberglove** | **FP-Block** |
| List of plugins | Blocking | Blocking | Blocking | Blocking | Blocking |
| List of fonts | Spoofing | randomizing | Blocking | - | - |
| User-Agent | Spoofing | | Spoofing | Blocking | Spoofing |
| HTTP header Accept-Language | Spoofing | - | Spoofing | - | Spoofing |
| appCodeName | Spoofing | - | - | Blocking | Spoofing |
| Product | Spoofing | - | - | Blocking | Spoofing |
| Product-Sub | Spoofing | - | - | Blocking | - |
| Vender | Spoofing | - | - | Blocking | Spoofing |
| Vendersub | Spoofing | - | - | Blocking | - |
| Online | - | - | - | Blocking | Spoofing |
| appVersion | Spoofing | - | - | Blocking | Spoofing |
| cookiesEnabled() | - | - | - | Blocking | Spoofing |
| javaEnable() | - | - | - | Blocking | spoofing |
| Navigator.mimeType () | blocking | - | Blocking | Blocking | Blocking |
| Screen color and pixel depth | spoofing | - | - | - | Spoofing |
| Screen width and height | Spoofing | - | - | - | Spoofing |
| Screen availLeft, availTop, availHeight and availWidth | Spoofing | - | - | - | - |
| Screen hosizontalDPI, verticalDPI | Spoofing | - | - | - | - |
| Canvas fingerprinting | - | - | - | - | - |
| Do not track | - | - | - | Blocking | Spoofing |
| Timezone | Spoofing | - | - | - | Spoofing |
| OS & Kernel Version | Spoofing | - | - | Blocking | Spoofing |
| JS: Flash Enabled | Blocking | - | Blocking | Blocking | Blocking |
| CPU | Spoofing | - | - | Blocking | Spoofing |
| Language | Spoofing | - | Spoofing | Blocking | Spoofing |
| Languages | Spoofing | - | Spoofing | Blocking | Spoofing |

### *D. Studying fingerprinting attributes*

This section had studied that if the fingerprinting attributes were modified, what are side-effects to the displaying of the web browser. This section will build the Chrome extension by injecting modified fingerprinting attributes before rendering a web page. The research has initially spoofed the low entropy of fingerprinting and then observe the problems of modified the low entropy attributes by visiting the regular website. Then, the research has changed the high entropy attributes and did the same process like the previous experiment.

- **Navigator (16 properties):**
  *appCodeName, appVersion, doNotTrack, product, productSub, cookieEnabled, vendor, vendorSub, online, platform, online, language, languages, JavaEnabled;*
- ***Navigator.mimeTypes (4 properties)***
  *enablePlugin, description, suffix, type;*
- **Screen (11 properties ):**
  *horizontalDPI, verticalDPI, availLeft, availTop, availHeight, availWidth, colorDepth, pixelDepth, width, height, bufferDepth;*

The high entropy consists of as follows:

- **List of plugins deriving from Navigator.plugins (4 properties):**
  *Name, filename, description, length;*
- **List of fonts**
- **UserAgent**

The web browser will be used to visit 30 favorite websites of alexa.com in order to study that if the fingerprinting attributes are changed before loading the web page, what is the consequence of modification should be considered.

TABLE 5 THE RESULT OF MODIFYING FINGERPRINTING ATTRIBUTES

| Attributes | Result |
|---|---|
| Low entropy | Not any effect to the web browser |
| High entropy | The effects to functionality of the web browser |
| Low and high entropy | The effects to functionality of the web browser |

Giving Table 5, it can be seen that changing little entropy attributes is not any effects to render of a web page as the high entropy of fingerprinting attributes has the influence of rendering the web browser.

### *E. The effects of information paradox*

Both Eckersley [2] and Nikiforakis [13] found that some existing fingerprint countermeasures had intended to prevent fingerprint tracking, but it turned out to be made the risk of being tracked or monitored by fingerprinters due to the unusual combination of browser-related information. Thus, it should consider this problem along with fingerprinting prevention. This section had tested the results of existing approaches which the results are shown in Table 6.

TABLE 6 INFORMATION PARADOX

| Countermeasure | Paradox |
|---|---|
| Tor | - |
| UserAgent Spoofing | Inconsistency between operating system and navigator.plugins |
| Fireglove | Inconsistency between CPU and userAgent |
| FP-Block | Inconsistency of DNT, CookieEnabled, JavaEnabled() |
| CanvasFingerprint Block | - |
| Stop fingerprinting | - |
| RubberGlove | Inconsistency between navigator object and HTTP header request |

From the result in Table 6, the number of attributes had increased the risk of abnormal combination implicitly. Therefore, handling few attributes had the lower risk of unusual combination than dealing with more attributes. Among experiment the research had learned that some attributes cannot control them. For instance, cookiedEnable and Do not Track (DNT) shown in HTTP header had shown the contradictory result on values of JavaScript object. However, dealing with few fingerprint attributes had increased the chance of being fingerprinted as the remaining attribute was not handled.

## V. CONCLUSION

This paper has attempted to discover the effectiveness of currently available fingerprint approaches. The result will be more useful for attentive users who are looking for some methods to protect their privacy. In term of efficiency of prevention, most countermeasures cannot prevent all types of fingerprinting which cause the web browser to be unable to avoid from being tracked. Regarding the user's experience, the existing countermeasure that uses blocking technique appears to generate more user dissatisfaction than other techniques. For studying fingerprint attributes, almost countermeasures use blocking and spoofing technique to prevent fingerprint tracking, and all countermeasures only handle the high entropy attributes to stop fingerprinting. In addition, the issue of anticipation that which fingerprint attribute might be selected by fingerprinters still be a challenge for fingerprint countermeasure in the future. Also, modifying fingerprint attribute with code injection shows that the high entropy attributes more side effects than the low entropy attribute. In term of the information paradox, some countermeasures cannot conceal the modified fingerprinting attributes, and the remaining attributes might be used to fingerprint the user computer.

## REFERENCES


[1] J. R. Mayer, "Any person... a pamphleteer": Internet Anonymity in the Age of Web 2.0," Undergraduate Senior Thesis, Princeton University, 2009.

[2] P. Eckersley, "How unique is your web browser?." pp. 1-18.

[3] S. Luangmaneerote. "hybrid fingerprinting website," http://www.pleasefingerprintme.org

[4] K. Boda, Á. M. Földes, G. G. Gulyás, and S. Imre, "User tracking on the web via cross-browser fingerprinting," Information Security Technology for Applications, vol. 7161, pp. 31-46, 2012.

[5] K. Mowery, and H. Shacham, "Pixel perfect: Fingerprinting canvas in HTML5," Proceedings of W2SP 2012 IEEE Computerc Society, pp. pp. 1-12, May 2012, 2012.

[6] L. Olejnik, C. Castelluccia, and A. Janc, "Why johnny can't browse in peace: On the uniqueness of web browsing history patterns." p. 16 pages.

[7] L. Lu, E.-C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences." pp. 199-214.

[8] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host Fingerprinting and Tracking on the Web: Privacy and Security Implications." p. 16 pages.

[9] T. Hupperich, D. Maiorca, M. Kührer, T. Holz, and G. Giacinto, "On the Robustness of Mobile Device Fingerprinting: Can Mobile Users Escape Modern Web-Tracking Mechanisms?." pp. 191-200.

[10] P. W. T. A. Project. "AudioContext Fingerprint Test Page," 12/06/2016, 2019; https://audiofingerprint.openwpm.com/.

[11] K. Weldemariam, "FPGuard: Detection and Prevention of Browser Fingerprinting." p. 293.

[12] N. Nikiforakis, W. Joosen, and B. Livshits. "Privaricator: Deceiving fingerprinters with little white lies," 25/06/2015; http://research.microsoft.com/pubs/209989/tr1.pdf.

[13] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless monster: Exploring the ecosystem of web-based device fingerprinting." pp. 541-555.

[14] Tor. "Tor Project: Anonymity Online," 27/04/2015; https://www.torproject.org/.

[15] J. S. Clary. "RubberGlove " 19/06/2016; https://chrome.google.com/webstore/detail/rubberglove/koabfojebhfdjnligkcihoeekimoekpg?hl=en.

[16] Mozilla_Public_License. "Chameleon," 20/06/2016; https://github.com/ghostwords/chameleon.

[17] C. F. Torres, H. Jonker, and S. Mauw, "FP-Block: Usable Web Privacy by Controlling Browser Fingerprinting." pp. 3-19.

[18] Addidrime.net. "CanvasFingerprintBlock," 15/05/2015; https://chrome.google.com/webstore/detail/canvasfingerprintblock/ipmjngkmngdcdpmgmiebdmfbkcecdndc.

[19] R. T. a. T. G. Ram Bhaskar. "Combatting Browser Fingerprinting with ChromeDust," 20/06/2016; https://css.csail.mit.edu/6.858/2013/projects/rambhask-tgalvin-rrt.pdf.

[20] stopfingerprinting. "StopFingerprinting," 20/06/2016, 2016; https://chrome.google.com/webstore/detail/stopfingerprinting/kfhlgmfkolojpnmhgggilmillpcokmnb?hl=en-US.

[21] G. Wilson. "User-Agent Switcher for Chrome," 15/05/2015; https://chrome.google.com/webstore/detail/user-agent-switcher-for-c/djflhoibgkdhkhhcedjiklpkjnoahfmg.

[22] askeing. "Canvas Fingerprint Blocker," 20/06/2016; https://addons.mozilla.org/en-US/firefox/addon/canvas-fingerprint-blocker/?src=api.

[23] K. Boda. "Firegloves," 23/04/2015; http://fingerprint.pet-portal.eu/?menu=6.

[24] NiklasG. "Stop Fingerprinting," 20/06/2016; https://addons.mozilla.org/en-US/firefox/addon/stop-fingerprinting/?src=api.