Timezone associated with the web browser language (navigator.language and navigator.languages). Therefore, randomising Timezone must also consider the web browser language. For example, if randomising that the web browser uses in the UK, it must display of en-US because it is impossible to support another language.

In order to reduce the information paradox, the consistency between JavaScript object (navigator) and HTTP header field must be coordinated. Usually, the communication from the Web browser to the website is done through HTTP protocol. The web browser conveys the HTTP request to the website and then the website will acknowledge the web browser with HTTP response as detailed in Table 8 and Table 9 below.

Table 8. Detail of request header

Request Header	**Data**
Accept	text/html,application/xhtml+xml,application/xml;q=0.9 ,image/webp,*/*;q=0.8
Accept-Encoding	gzip, deflate, sdch
Accept-Language	en-US,en;q=0.8
Cache-Control	max-age=0
Connection	keep-alive
Cookie	ga=GA1.2.700983430.1460133764;_cbclose=1; _cbclose22799=1;_uid22799=9D5FEB32.11; verify=test
DNT	1
Host	dict.longdo.com
Referer	http://dict.longdo.com/search/unseeing
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94 Safari/537.36

Table 9. Detail of response header

Request Header	**Data**
Connection	keep-alive
Content-Encoding	gzip
Content-Length	24688
Content-Type	text/html
Date	Fri, 13 May 2016 15:29:20 GMT
Server	nginx/1.8.0
Vary	Accept-Encoding
X-Powered-By	PHP/5.3.10-1ubuntu3.19

The FP-prevention model will embed a proxy in the browser extension. The communication between the web browser and the web server will be intercepted every time consistent with randomised UserAgent. Therefore, HTTP requests filed that are conveyed to the web server will not be the authentic HTTP field.

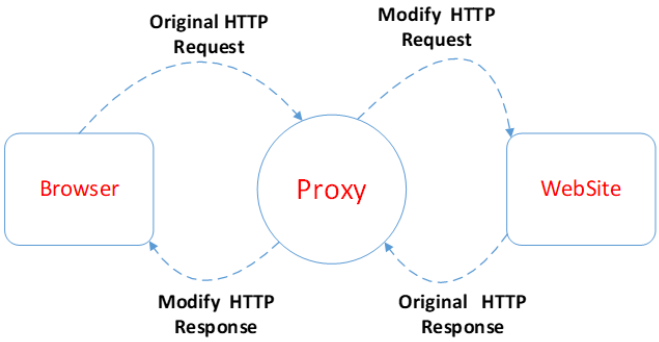

Figure 3. Workflow of embed proxy

To considering the Figure 3, the concept of operation will listen to HTTP-on-modify-request and notify browser extension while HTTP request is ready to send. The browser extension will intercept HTTP request and then replace the original HTTP request by spoofing UserAgent and Accept-Language which obtain from a defined list of UserAgent. This mechanism will assist problem of abnormal combination between HTTP Header and JavaScript object.

5.1. Features of FP-prevention

The purpose of FP-prevention aims to change the identity of the web browser. The result of changing browser's identity can inhibit fingerprinting tracking, and the web page is able to execute web pages smoothly without annoyance to the user.

FP-Prevention model differs from the current existing method as following. Firstly, this model uses only randomization technique (not blocking) by considering the relationship of attributes. Reducing the number of fingerprinting attributes unrelated is to mitigate the browser stand out which could cause easier fingerprinting. Secondly, this model considers sufficient fingerprinting attributes to prevent effectively and could reduction of side-effects to the web browser. Also, this model does not randomise all fingerprinting attributes. Some attributes are ignored (e.g., Do Not Track (DNT) or cookieEnabled) because these attributes are set to the user preference which randomising these attributes probably make browser stand out. Thirdly, this model considers the consistency between HTTP header and JavaScript object. Many existing randomization techniques (UserAgent spoofing, Fireglove, FP-block and RubberGlove) focus only on JavaScript objects which produce the biggest mistake because some JavaScript objects related to HTTP header are detected by the fingerprinting companies. Thus, considering the consistency between HTTP header and JavaScript object would reduce the problem of browser stand out as well. Fourthly, this model has improved the workflow of randomization technique. It is unnecessary to randomise all attributes. This model starts to randomise from UserAgent to other attributes because it contains many attributes inside which UserAgent randomised at once can link automatically

to other attributes. Fifthly, this model requires inhibiting hybrid fingerprinting because fingerprinters currently attempt to fingerprint browser with multiple ways. The information is shown below in Table 10.

Table 10. Feature of FP-prevention

	Prevention Features					
Countermeasure	**Object fingerprinting**	**Font fingerprinting**	**Plugin fingerprinting**	**Canvas fingerprinting**	**Reducing browser stand out**	**Consistency between JavaScript and HTTP header**
Fireglove	√	√	√			
FP-Block	√		√	√	√	√
Stop fingerprinting		√	√			
RubberGlove	√	√	√	√		
FP-prevention	√	√	√	√	√	√

6. Conclusion

This paper desires to realize the capability and limitation of existing countermeasures. The given results indicate that the existing countermeasures still have a shortcoming. The number of selected attributes and method to inhibit the fingerprinting tracking contribute to the browsing experiences. In addition, some existing countermeasures remain creation of information paradox which leads to easier fingerprinting. The countermeasure that prevent well does not mean the great countermeasure. The great fingerprinting countermeasure should regard the part of fingerprinting prevention, side-effects and information paradox simultaneously. The paper proposes a new model, FP-prevention, to address the weakness of existing countermeasures.

7. 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?," in PETS'10 Proceedings of the 10th international conference on Privacy enhancing technologies Springer-Verlag Berlin, Heidelberg, 2010, 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," in *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, 2012, p. 16 pages.

[7] L. Lu, E.-C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences," in In Proceedings of the European Symposium on Research in Computer Security, 2010, 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?," in Proceedings of the 31st Annual Computer Security Applications Conference, 2015, 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," in *Data and Applications Security and Privacy XXIX: 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings*, 2015, 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," in *Security and Privacy (SP), 2013 IEEE Symposium on*, 2013, pp. 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.

