My Site Knows Where You Are: A Novel Browser Fingerprint to Track User Position

Tianqi Wu[†*§], Yubo Song[†*§], Fan Zhang[‡], Shang Gao[¶], Bin Chen[†*§]
[†]*School of Cyber Science and Engineering, Southeast University*
[*]*Key Laboratory of Computer Network Techonology of Jiangsu Province*
[§]*Purple Mountain Laboratories*
[‡]*Telecommunications Software & Systems Group, Waterford Institute of Technology*
[¶]*Department of Computing, The Hong Kong Polytechnic University*
220205042@seu.edu.cn, songyubo@seu.edu.cn, fzhang@tssg.org,
goldensaintgao@qq.com, chenbing9961@163.com

***Abstract*—Utilizing browsers to identify and track users has become a routine on the Web in recent years. It is easy for the browser to collect sensitive information and construct comprehensive user profiles while the users are still unaware. As the problem mentioned above, several anti-fingerprint mechanisms have been adopted to protect user privacy. However, our research finds a novel method based on localization fingerprints that may still threaten user privacy. The location fingerprint obtains the response delay of data transmission over the link between the users and the third-party sites. Since the physical link state information between the host and the remote website is distinct and steady, it can be used to extract statistical features and construct user profiles. We implement a multilateration cross-site image resource request scheme to collect link-state information of users and develop a prototype called PingLoc to evaluate the effectiveness. About 1,093 users from all over the world are involved in our experiment. The evaluation shows that the delay features collected are stable, and the accuracy of the localization fingerprint is up to 98%. Pressure testing shows that the PingLoc is robust against various anti-fingerprint mechanisms and achieves 93.5% accuracy for browser switching, 80.6% accuracy for virtual machine disguising, and 88.2% accuracy for IP rotation.**

***Index Terms*—Browser fingerprint, user tracking, cross-domain resource request**

I. Introduction

In recent years, it has become widespread for websites to identify and track users on the Web. Sites can use scripts to accurately identify users and track them, collecting their interests and hobbies to provide more smart and personalized service. For example, recommend ads precisely based on users' interests[1]. However, it might lead to the leak of sensitive information and threaten user privacy, such as phone number, email number, and social account.

Browser fingerprint is a standard mechanism used to identify and track users through configuration information and browser information visible to the website, such as fingerprint-based on Cookie or Evercookie and fingerprint-based on browser plugins/extensions[2]. The abuse of browser fingerprints may challenge the security of user privacy. For example, the websites might collect information about your social account while you are still unaware. To solve the problem, the browsers adopt a stricter security policy to limit the execution of scripts and install specific browser plugins to poison the data collected[3].

Although the browsers apply several mechanisms against browser fingerprint and limit the information collection, but not thoroughly abandon information collection. Therefore, the users' fingerprints can still be obtained through the information permitted to collect. This paper introduces a novel method, utilizing cross-domain image resource requests to get the physical location fingerprint.

The physical location fingerprint on the browser utilizes the physical time delay, from the users' browsers to one or more web servers, to extract users' location fingerprint and determine their identity. Unlike other browser fingerprints, physical location fingerprint does not collect information from users' browser itself but collects physical position information to construct user profiles. Compared to other browser fingerprints, the physical location fingerprint has the advantages as follows:

- The users' browser might add noise into the information collected to interfere with the websites or manipulate the information to pretend to be different users. However, it is difficult to manipulate the physical position information.
- Since the physical location fingerprint utilizes the position information rather than browser information, it has less relevance to users' browsers. As a result, the physical location fingerprint can be more robust to identify the users using more than one browser platform. However, It is hard for the existing browser fingerprint to identify and track users who frequently change browsers.
- The browsers have applied several mechanisms to limit the collection of position information, such as GPS. Therefore, websites cannot achieve position information directly. However, the method introduced in this paper utilizes the response of a cross-domain image resource request to get the delay time of users' browsers, which does not need authorization. So it can get the position information while users are unaware.

The physical location fingerprint can be applied to various scenarios to identify and track users. Furthermore, it can obtain more realistic user profiles. It can be used to track the users covering their information maliciously, such as the fake users on the social network. It also can be used to track attackers and find their position on the network. When supported by a huge position database, it can be used to realize user localization as well.

However, there is still a challenge to face. It is not easy to obtain position information of users on the browser. As mentioned above, to bypass the detection of the browser, GPS and IP addresses will not be used in physical location fingerprint since such information is protected. To solve the problem, we have found that the link-state information of the user's access to the remote web servers is stable. Meanwhile, the link-state information of different users varies greatly. Therefore, the user's position information can be inferred by detecting the user's network link-state information, and the distinction may be helpful to construct user profiles.

Packet Internet Grouper is a classic method for detecting the link-state information of the network[10]. However, due to the limitation of the web API, it is difficult and unsafe to ask the users to execute such commands on the browser. But it is possible to achieve the link state information from users to third-party websites, such as amazon, Facebook, Twitter, and so on. Then the link state information can be utilized to get the position information of users. However, the Same Origin Policy (SOP) of the browser limits the users to access the resource on third-party websites in order to protect user's privacy. However, in this paper, we will introduce the response of cross-domain image resource requests to get the delay time of users' browsers. It is similar to the ping command but implemented on the Web, which can detect the connection status of users to third-party websites on the browser. Then, the user's network link-state information can be obtained on the Web and infer the user's login position to detect the user disguised as different users from the same position.

The main contributions in this article are as follows:

- For the first time, we propose a multilateration cross-site image resource request scheme to obtain the physical location fingerprint of users according to time delay, which can be used to identify users because the delay is relative to physical link state information from user hosts to web servers.
- We develop a prototype system PingLoc, which is made up of a web application for user fingerprint extraction and a classification model for user identification. PingLoc utilizes the response of cross-domain image resource requests to obtain the link state information and bypass the limitation of SOP.
- We propose a position detection algorithm and extract the statistical characteristics of the delay time sequences. We also compared the performance of different machine learning algorithms to track users in PingLoc.
- To verify the performance of PingLoc, we collect more than 1,300,000 records from 1093 browsers worldwide and obtain a classification accuracy of 98%. Besides, we test the robustness of PingLoc by imitating several disguising methods commonly used and achieve 93.5% accuracy for browser switching, 80.6% accuracy for virtual machine disguising, and 88.2% accuracy for IP rotation.

II. Related Work

To identify the users on the browser, various kinds of browser fingerprints have been attracting growing attention in the last few years. Abouollo [4] proposes a scheme using fingerprints based on canvas to detect fake users on Online Social Networks (OSN). Laperdrix [5] verifies the validity of several browser fingerprint schemes and proposes that canvas fingerprints can provide the greatest user discrimination. Besides, Laperdrix analyzes the selection of suitable canvas parameters to obtain the maximum discrimination for each user [3]. Alswiti [6] puts forward a different user identification scheme. They propose that users have unique behaviors. They can construct user profiles based on the user's behavior on the website and identify users. In addition to user behavior, browser extensions can also be used to identify users. Gulyas [7] discusses the scheme for user identification by establishing user profiles with unique browser extensions installed by each user.

However, the solutions above are difficult to deal with anti-fingerprint and anti-tracking methods because they still focus on the information collected from the user browser itself, so users may frequently change their browser fingerprint easily to make the detection fail and protect their privacy. To solve this problem, Li [8] proposes a solution to improve the stability of the detection by linking the new fingerprint to the old one. But their solution can only deal with the users who frequently use the same browser.

With the problem above, we propose a physical location fingerprint. It is inspired by Mirsky [9]. Mirsky detects man-in-the-middle attacks through machine learning by analyzing the ping response in the LAN. This paper makes use of multilateration ping on the HTTP layer and uses several learning algorithms to "localize" users and, according to the unique location fingerprint, to identity and track the user browsers.

III. The physical location fingerprint

The existing browser fingerprint schemes take focus on the collection of information on user browsers by extracting features from the information collected, create user profiles, and construct user fingerprints. But usually, users have the authority to modify the parameters of their browsers and change the features of their browsers, pretending to different users. For example, poison the data collected with browser plugins, confuse the detection of websites, and protect the users' privacy.

However, the physical location fingerprint does not focus on the information of the browser itself but the link-state

information of the user browser to remote third-party websites. This link-state information of user browsers is opened, easy to collect, and hard to control and change for users.

Here we will introduce a cross-domain image resource request scheme to obtain the link-state information. It is achieved by using browsers to request picture resources to third-party websites without being restricted by the Same Origin Policy (SOP) and Cross-Origin Resource Sharing (CORS). The browser restricts scripts to access resources on cross-origin webpages by SOP and CORS, which prevents the attackers from obtaining the private information of users when visiting other websites[10]. However, some resources are not limited by SOP and CORS, such as image resources. Therefore, the time of delay from the user's browsers to the target website can be obtained with the request to image resources of cross-domain websites. The image resources loaded can be a favicon of the website, and the time of delay can be calculated by capturing the onload event. Or load a non-existent image resource, and the user's browser will get an error response. Then the time of delay can be calculated by capturing the onerror event.

We find that the cross-domain image resource scheme can get the delay of time like the real ping command. Figure 1 shows the value of two browsers from different positions to the same website, the horizontal axis is the times of tests, and the vertical axis is the value of time delay. It is obvious that within a period of time, the value of time delay from the same browser to the website is stable, while there is a big difference between the two browsers. Therefore, the cross-domain image resource scheme can also be used to discriminate different browsers.

To create user profiles and build up physical location fingerprints, it is needed to extract appropriate features to build a training set for subsequent machine learning algorithms. In this paper, we select 11 well-known websites and extract 7 statistical features for each website. Therefore, for each sampling point of each user, a 77-dimensional feature vector is created for subsequent model training.

The statistical features we selected in this paper are followed. n is the window size and x is the sampled data value

- Max: Maximum value of the window.

$$Max = max\{x\} \tag{1}$$

- Min: Minimum value of the window.

$$Min = min\{x\} \tag{2}$$

- Mean: Average value of the window.

$$Mean = \frac{1}{n}\sum x \tag{3}$$

- Var: Variance of the window.

$$Var = \frac{1}{n-1}\sum (x - Mean)^2 \tag{4}$$

- RMS: Root-Mean-Square of the window. RMS reflects the noise of the data window.

$$RMS = \sqrt{\frac{\sum x^2}{n}} \tag{5}$$

Because the data collected is affected by many factors and does not completely follow the normal distribution, It is needed to discuss the skewness and kurtosis of the data.

- Skew: Asymmetry of the window. Skewness reflects the skew direction and degree of data distribution.

$$Skew = \frac{\sqrt{n(n-1)}}{n-2}\left[\frac{\frac{1}{n}\sum (x - Mean)^3}{(\frac{1}{n}\sum (x - Mean)^2)^{\frac{3}{2}}}\right] \tag{6}$$

- Kurt: Peakedness of the window. Kurtosis reflects the steepness of the data distribution.

$$Kurt = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\frac{\sum (x - Mean)^4}{(\sum (x - Mean)^2)^2} - 3\frac{(n-1)^2}{(n-2)(n-3)} \tag{7}$$

Since the meanings of the statistical features selected are different, the actual values of the features also vary greatly. In order to eliminate the influence of the huge difference in the value of different statistical features on the subsequent model training and ensure the reliability of the results, it is necessary to standardize the data during data preprocessing. In this paper, min-max standardization method is used to process the data so that the data falls within the interval of [0,1].

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{8}$$

IV. System Design

To prove that the physical location fingerprint may threaten the security of users' privacy, we developed a prototype named PingLoc to identify user devices. In this section, we will introduce the architecture of PingLoc and the operation process of PingLoc.

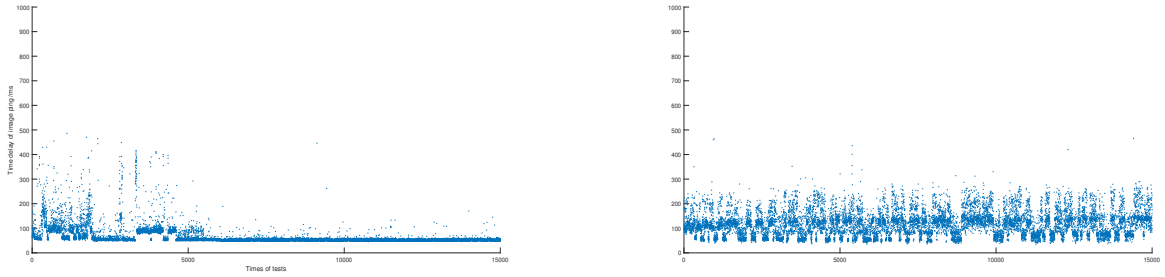
A. Architecture

PingLoc utilizes web applications to collect time delay of the user browser to remote third-party websites and pass it to the learning algorithms for model training through the server. Then, the model trained can be applied to localize user browsers.

PingLoc makes use of the scripts to collect the user's time delay information and sends it back to the website server. The website server is responsible for data preprocessing and extracting statistical features. After that, the server passes the data to the learning algorithms for model training and applies the final training results into user localization.

PingLoc can be roughly divided into the following steps: user browser sampling, server data processing, feature extraction, and model training.

The scripts on the webpages are usually handed out by the web server and executed by the client browsers. Therefore,



(a) The time delay value of position a

(b) The time delay value of position b

Fig. 1. The time delay of two position to a same website

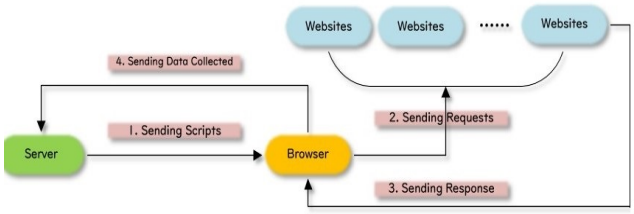


Fig. 2. The process of PingLoc

the scripts can be embedded on the webpage of our website and send it to the user's browser, which will execute it and will feedback the obtained data to our server.

B. Data Collection

For further work, it is needed to implement a multilateration cross-domain image resource request scheme to collect the link-state information of the user browser to a group of third-party sites and send them back to the server.

In PingLoc, the timestamp is used as the relative path of the image resources. In this way, different paths can be generated each time, and indeed there is no such image resource path named by the timestamp on the target website. Therefore, the target website always returns an error, and the time of delay can be obtained by catching the onerror events.

However, it is not enough to collect responses from just one site. More responses from different websites can reflect more differences among browsers. In order to improve classification accuracy and reduce the impact of complex network environment noise, it is advised to collect as many time delays from users to different websites as possible. In this paper, the time delay from users to 11 well-known websites is collected.

C. Data preprocessing and features extraction

The collected data cannot be directly used for model training. It is needed to preprocess the data collected and extract appropriate features before it can be used for training.

First of all, similar to the ping command in ICMP, the request sent might lose as well, and the lost packet should be removed during data preprocessing. The cause of the packet loss may be the transmission error of the data stream in the network, or an error that occurred during the HTTP packing or unpacking process, or it may be because the network propagation delay is higher than the delay packet loss threshold. In order to improve the efficiency of sampling, a packet loss threshold is set. After the browser sends the request, if there is still no response from the target website after a packet loss delay, it is considered that the packet loss has occurred and prepare to resend the request. The packet loss rate of the experimental data used in this paper is about 3% and can be ignored.

In order to improve the accuracy of the classification of our model, it is not appropriate to directly use the data collected as training data because the existence of extreme values since the noise of a complex network environment may affect the training results and decrease the accuracy of the trained model. In this paper, a window is used to extract the statistical features of a group of data as training set data, which can ensure that the data used in model training is less affected by extreme values. We will discuss in Section 5 how to select the appropriate window size to extract statistical features.

D. Model training

In order to improve the accuracy of model training as much as possible, several common machine learning algorithms are used to train our classification model and evaluate their performance, including Support Vector Machine (SVM), Decision Tree, Bayes classifier, K-Nearest-Neighbor (KNN), and Random Forest.

The algorithms are used to perform model training and cross-validation on the extracted feature vectors. We will discuss the efficiency of each algorithm in Section 5.

V. Experiment and Evaluation

In this section, we will discuss how to select appropriate window size to extract statistical features and discuss the performance of various learning algorithms in the PingLoc system to classify different users.

We collect data from 1093 browsers with different hardware browser types, different operating systems, and different browser platforms worldwide. We extract their time delay to 11 well-known websites. These websites involve different fields such as search engines, news, economy, academy, and

entertainment. Throughout the whole process, we have collected more than 1300,000 user browser time delay records. Our data used for the experiment in this paper can be found at https://github.com/1362860831/PingLoc.

A. Select the size of the window

The size of the window affects the effect of feature extraction and training. If the selected window is too small, it is not easy to reflect the statistic features of the data; but if the selected window is too large, too many similar neutral data frames will be generated to affect the training result. In order to find the most appropriate size of the window, we change the size of the window to obtain greater classification accuracy.

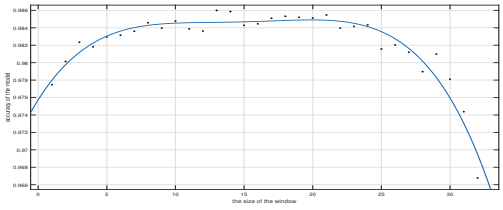


Fig. 3. The prediction accuracy under different window sizes

The horizontal axis in figure 3 is the window, and the vertical axis is the accuracy of the model obtained by training. The curve in the figure is drawn by polynomial fitting according to the experimental results. It can be found that accuracy will decrease when the selected window is too small or too large, and the appropriate window size should be ranged from 20 to 30. In this paper, the window size for feature extraction equals 25.

B. Performance of each learning algorithm

Make use of the algorithm mentioned in Sec4.3 to train our classifier models. 10-fold cross-validation was performed during the training to more accurately estimate the prediction accuracy of each algorithm. The accuracy of each algorithm is shown in figure 4(a). It is found that the accuracy of the KNN and Random Forest algorithms is optimal and more than 90%, and performs better than the other algorithms in the accuracy of the classification.

Receiver Operating Characteristic Curve (ROC) is also important for evaluating training performance. Figure 4(b) is the ROC of the five learning algorithms mentioned in Section 4. The horizontal axis is the false positive rate (FPR), and the vertical axis is the true positive rate (TPR). The closer to the upper left corner of the curve means a higher TPR and a lower FPR, and that is to say, the classification will get fewer misjudgments, and the classification is more accurate. The area formed by the ROC and the horizontal axis is usually used to measure the accuracy of classification, which is called Area Under ROC (AUC). The larger the value of AUC, the better the classification is. It is found that random forest and KNN have a better performance in the figure.

Since the PingLoc system needs to be applied to actual web applications, it is needed to consider the time consumed by the classification process. In order to ensure the normal use of ordinary users, it should not spend too much time in the detection process. Otherwise, it will destroy the user's experience and increase the load of the verification server. Figure 6(c) shows the time consumption of each algorithm under the same hardware conditions. The time consumption of SVM and KNN is significantly higher than the other three types of algorithms. The time consumption of SVM and KNN is significantly higher than the other three types of algorithms but still far less than the average time users stay on the webpage.

VI. The discussion on robustness and privacy

In this section, we imitate several disguising methods commonly used to pretend to be real users by tampering with the system parameters, including modifying IPs, switching browser platforms, using virtual machines, and using these methods to test the performance of PingLoc to identify users.

We simulate that the user controls a device in the network and can freely switch the operating system such as Windows7, Windows10, Ubuntu16.04, and Debian7 through a virtual machine. Furthermore, the user can freely choose different browsers such as Chrome, Firefox, IE and Edge to perform as different users.

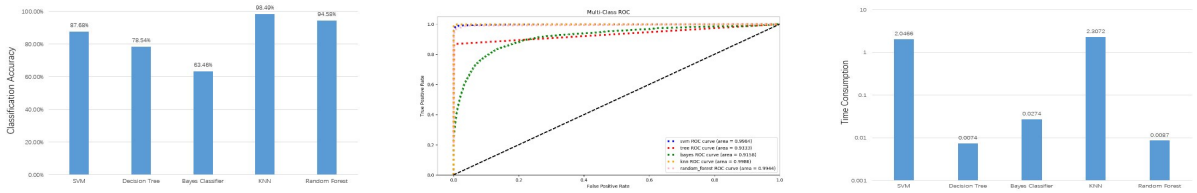
A. The robustness discussion

1) Switch different browser: In the related research of browser fingerprint, user identity authentication across browser platforms has always been a hot topic. The common way is difficult to achieve cross-browser identity authentication, such as extracting cookies or canvas fingerprints[11]. In this paper, we use PingLoc to test the users using different browsers to masquerade as real users.

In our experiment, the user used the Chrome, Firefox, IE and Edge browser platform and used a different version of the browsers plugins to increase differentiation. Among the 524 user records collected, the prediction accuracy of PingLoc in the KNN-based model was as high as 93.5%, while the prediction accuracy of the model based on the random forest reached 86.1%.

2) Use virtual machine: Using a virtual machine is another method commonly used to bypass authentication. Under different operating systems, the browsers share different encoding methods and different hardware drivers, which may cause the existing detection scheme to fail to construct the correct user profiles.[12] Furthermore, the virtual machine can change the hardware configuration of the browser easily or install different browser platforms on the operating system to increase the distinction.

In this paper, we use different operating systems on virtual machines such as Windows7, Windows10, Ubuntu16.04 and Debian7, and try to change the hardware configuration of the virtual machine. For example: allocate different memory sizes and CPU cores to virtual machines. Among the



(a) The accuracy of different learning algorithms

(b) The ROC and AUC of different learning algorithms

(c) The time consumption of different learning algorithms

Fig. 4. The evaluation of different learning algorithms

8245 user records collected, PingLoc's prediction accuracy of KNNbased models is as high as 80.6%, while the prediction accuracy of random forest-based models reaches 68.8%.

3) IP rotation: It is also possible for the user to rotate IP to increase the difference between the fake users[13]. This paper switches 100 random IP address and collect 1048 user records. The prediction accuracy of PingLoc based on the KNN model reaches 67.6%, while the prediction accuracy of the model using random forest reaches 88.2%.

B. The privacy discusion

To deal with the potential threat of the novel location fingerprint since the abuse of our work, we think several countermeasures against position detecting is needed:

- A stricter Same Origin Policy is needed. The physical location fingerprint provided in this paper is realized since the loose restriction of SOP on cross-domain image resources. The link-state information of user browser to remote websites can be achieved with access to cross-domain resources, then the browser fingerprint of user can be calculated.
- Browsers have to follow a stricter privacy policy. Although many browsers have limited the scripts collecting the privacy information of users, there is still a way to get the link-state information to the third-party websites, which may provide convenience for attackers.

VII. Conclusion

In this paper, we introduce a novel physical location fingerprint to identify and track users on the browser. This scheme utilizes multilateration response delay of data transmission over the link between the users and the third-party site to obtain the link-state information of the user browser, which is highly relevant to users' physical location and can be used to build location fingerprint. We discuss the implementation and introduce the prototype system PingLoc. Finally, we test and evaluate PingLoc, and the experiments show that the identification accuracy of PingLoc is up to 98% and has a 93.5% accuracy for browser platforms switching, 80.6% accuracy for virtual machine disguising, and 88.2% accuracy for IP spoofing, proving that PingLoc is robust against several common disguising methods.

Acknowledgement

This work is supported by Frontiers Science Center for Mobile Information Communication and Security, Southeast University, Nanjing, China. Yubo Song is the corresponding author.

References

[1] S. Englehardt and A. Narayanan. 2016. Online Tracking: A 1-million-site Measurement and Analysis. *2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1388–1401.

[2] Cao Y, S. Li , and E. Wijmans . "(Cross-)Browser Fingerprinting via OS and Hardware Level Features." *Network and Distributed System Security Symposium 2017.*

[3] P. Laperdrix, G. Avoine, B. Baudry, and N. Nikiforakis, "Morellian Analysis for Browsers: Making Web Authentication Stronger with Canvas Fingerprinting," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 11543, R. Perdisci, C. Maurice, G. Giacinto, and M. Almgren, Eds. Cham: Springer International Publishing, 2019, pp. 43–66.

[4] A. Abouollo and S. Almuhammadi, "Detecting malicious user accounts using Canvas Fingerprint," *2017 8th International Conference on Information and Communication Systems (ICICS)*, Irbid, 2017, pp. 358-361.

[5] P. Laperdrix, W. Rudametkin and B. Baudry, "Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints," *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, 2016, pp. 878-894.

[6] W. Alswiti, J. Alqatawna, B. Al-Shboul, H. Faris and H. Hakh, "Users Profiling Using Clickstream Data Analysis and Classification," *2016 Cybersecurity and Cyberforensics Conference (CCC)*, Amman, 2016, pp. 96-99.

[7] G. G. Gulyas, D. F. Some, N. Bielova, and C. Castelluccia, "To Extend or not to Extend: On the Uniqueness of Browser Extensions and Web Logins," in *Proceedings of the 2018 Workshop on Privacy in the Electronic Society* - WPES'18, Toronto, Canada, 2018, pp. 14–27

[8] X. Li, X. Cui, L. Shi, C. Liu and X. Wang, "Constructing Browser Fingerprint Tracking Chain Based on LSTM Model," *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, Guangzhou, 2018, pp. 213-218.

[9] Y. Mirsky, N. Kalbo, Y. Elovici and A. Shabtai, "Vesper: Using Echo Analysis to Detect Man-in-the-Middle Attacks in LANs," in *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1638-1653, June 2019.

[10] D. F. Somé, "EmPoWeb: Empowering Web Applications with Browser Extensions," *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2019, pp. 227-245

[11] Károly Boda, dám Máté Flde s, Gábor Gyrgy Gulyás, & Sándor Imre. (2011). "User Tracking on the Web via Cross-Browser Fingerprinting." *Nordic Conference on Information Security Technology for Applications. Springer-Verlag.*

[12] Solomos, K. , Ilia, P. , Ioannidis, S. , & Kourtellis, N. . (2018)."Cross-device tracking: systematic method to detect and measure cdt."

[13] N. Vlajic, P. Madani and E. Nguyen, "Anonymity of TOR Users Demystified," *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, 2017, pp. 109-114