A Novel Website Fingerprinting Attack against Multi-tab Browsing Behavior

Xiaodan Gu, Ming Yang, Junzhou Luo
School of Computer Science and Engineering
Southeast University, Nanjing, P.R. China
{guxiaodan; yangming2002; jluo}@seu.edu.cn

***Abstract*—Website Fingerprinting (WF) attacks have posed a serious threat to users' privacy, which allow an adversary to infer the anonymous communication content by using traffic analysis. Recent studies have demonstrated the effectiveness of WF attacks through a large number of experiments. However, some researchers believe that the assumptions of WF attacks vastly simplify the problem and are critical in the practical scenarios. In this paper, we assess the threat model of WF and relax the assumptions about browsing behavior to improve the practical feasibility. To deal with the multi-tab browsing scenario, we propose a novel WF attack and identify webpages respectively. The main idea resides in the fact that the user visits the second page with a short delay after opening the first page due to the think time. We analyze the anonymous traffic transmitted in the delay and select fine-grained features to identify the first page. Furthermore, we exclude the first page's traffic and utilize coarse features to identify the second page. We deploy our attack in real word environment and the experiment lasts for two months. The Naive Bayes classifier is then applied on the collected datasets to classify the visited websites among 50 top ranked websites in Alexa. When the delay is set to 2 seconds, our attack can classify the first page with 75.9% accuracy, and the second page is 40.5%. The results show that the WF attack is still effective in the practical scenarios and we can't dismiss WF as a threat.**

Keywords—Website Fingerprinting; traffic analysis; pattern recognition; privacy;

I. Introduction

Privacy leakage on the web is a hot topic. On account of the increasingly deliveries of web applications, people reveal lots of sensitive data online, including social security numbers, credit card numbers, health reports, etc. If the privacy data are transmitted in clear text between the client and web server, adversaries can get them by sniffing the network traffic. Even though users encrypt the content, adversaries still can infer their browsing activates according to the communication relationships with web servers. As a remedy, several low-latency anonymous communication systems, such as Tor, are proposed to protect users from eavesdroppers by encrypting the content and hiding the communication relationships. Among them, SSH proxy is a widely used single-hop system, which can get higher performance than the multi-hop systems. By establishing an encrypted tunnel between the proxy and its client, SSH proxy encapsulates and forwards all the packets through the tunnel. So the identities of users' destination websites can be hid.

Although SSH proxy obscures the content and recipient, rich information still can be inferred using traffic analysis. The nature of the problem is that the single-hop system never drastically transforms the sizes, timing, frequency, and order of packets. Utilizing these features, the attacker can construct a fingerprint and identify the real communication partner. In short, the Website Fingerprinting (WF) attack is a passive fingerprinting technology to infer the communication content based on the traffic pattern. Recently, several works have implemented numerous experiments and demonstrated the effectiveness of the WF attacks. However, more and more people [1, 2] criticize this technology for its overestimating the attacker's abilities. They believe that challenging assumptions in the threat model will vastly simplify the problem. For example, all researchers assume that the attacker can separate the overlapped anonymous traffic of different pages. But in practice, there is no method to achieve this goal. As a result, criticizers believe that these unrealistic assumptions have a significant impact on the efficacy of WF attack.

In this paper, we study the multi-tab browsing behavior and propose a novel WF attack to identify webpages respectively. To the best of our knowledge, we are the first to carry out a WF attack aiming at the overlapped anonymous traffic. First, we evaluate the practical feasibility of the threat model of WF attacks and relax some assumptions by allowing the victim to load another page in the background. Second, we perform the data preprocessing to determine whether the traffic is overlapped. Then we extract appropriate features to create fingerprints according to the users' retrieval pattern. Finally, we identify two pages with the Naive Bayes Classifier [17] respectively. The results show that the WF attack is still effective in the practical scenarios.

The rest of this paper is organized as follows. Section II presents the related work. In Section III and IV, we assess the threat model and propose a novel WF method. We provide the results of our experiment in Section V. Section VI concludes this paper and discusses the future work.

II. Related Work

When HTTP /1.0 was the dominant protocol deployed in the World Wide Web, it clearly specified that each web object should be transmitted through a new TCP connection. What's

978-1-4799-2002-0/15/$31 ©2015 IEEE

more, the immature anonymous technologies merely encrypted the content without transferring other features. Based on this finding, the number and sizes of objects were easily obtained by examining TCP connections. Both features were good enough to distinguish websites. Hintz [3] introduced the idea of WF attack to disclose the security risks in SafeWeb. Coincidentally, Sun et al. [4] also presented a similar attack against the SSL encrypted connection at the same time. They used the Jaccard's coefficient to measure similarity and achieved the detection rate of about 75%. As the HTTP /1.1 protocol began to adopt persistent connections and pipelining, there was no need to open a new TCP connection for each web object. Furthermore, the emergence of practical anonymous systems completely removed the information about objects. They established an encrypted tunnel and mapped TCP connections to different logical channels. As a result, the aforementioned attacks based on objects' sizes and number were no longer feasible.

Researchers began to pay more attention to other traffic characteristics and similarity algorithms for identifying websites. Bissias et al. [5] were the first to utilize some fine-grained features including the distribution of packet sizes and inter-arrival times to create fingerprints. Then they evaluated the detection accuracy on several different combinations of features by using the cross correlation technology. They found that longer delays between training and testing data only slightly decreased accuracy. Liberatore and Levine [6] carried out a more extensive investigation with an interesting list of 2,000 distinct websites. In the evaluation, they creatively applied a mature pattern classification algorithm, namely the Naive Bayes classifier, to achieve the success rate of up to 70%. Herrmann et al. [7] indicated that applying the common text mining techniques to attributes could improve accuracy. They extracted the direction, packet sizes and frequency to represent fingerprints, which were identified with the Multinomial Naive Bayes classifier. By visiting some websites repeatedly, Lu et al. [8] found that the sequences of incoming and outgoing packet sizes were very similar if they belonged to the same site. They utilized the Edit Distance to measure the similarity between sequences and the identification accuracy reached 81%. Panchenko et al. [9] extracted the fingerprints based on traffic volume, time, and direction of packets, and then utilized the Support Vector Machine (SVM) as the classifier. The result is significantly improved from 3% to 54% in Tor. Recently, Cai et al. [10] replaced the Radial Basis Function in SVM with the Damerau-Levebshtein Distance, which made a great success with over 83% accuracy. Wang et al. [11] focused their attention on Tor cells rather than TCP/IP packets and provided an experimental method to remove Tor SENDME cells. They also used several distance-based metrics to improve SVM classification. Furthermore, based on the multi-modal property of web pages, they [20] applied a K-Nearest Neighbor classifier on a large feature set with weight adjustment. While WF attacks emerged as a serious threat to users' privacy, people began to research on defense mechanisms. For example, Wright et al. [12] proposed traffic morphing, which optimally morphed a page's traffic to similar to another. From the point of view of the browser side, Luo et al. [13] proposed a novel system, namely HTTPOS, which provided a comprehensive and configurable suite of traffic transformation techniques. All these countermeasures were aimed to thwart statistical traffic analysis algorithms. However, in 2012, Dyer et al. [14] carried out numerous experiments and showed that nine known countermeasures were vulnerable considering coarse traffic features. The result of Cai et al. [15] also proved it. In the latest research, they presented a mathematical framework to evaluate attacks and countermeasures. They argued that current WF defenses were not as hopeless as people think. With some appropriate defenses users also could protect their privacy. In addition, at CCS 2014, Juarez et al. [1] believed that WF attacks oversimplified the threat model and criticized the practical feasibility in many aspects, including users' browsing habits, differences in location, version of Tor Browser Bundle, etc.

As we can observe, most researchers make unrealistic assumptions during WF attacks and get high detection rates, which can't be achieved in the practical scenarios. Even though Juarez et al. have proved the lack of practicality of previous works, they don't provide any solution. To address this problem, we relax the assumptions about browsing behavior and propose a novel WF attack to improve the practical feasibility.

III. Threat Model

In this section, we first assess the assumptions of the general threat model proposed in previous works of WF attacks. Then we relax some assumptions which require strong ability of the adversary to make the attack more realistic.

A. General Threat model and Assumptions

Fig. 1 illustrates a basic attack scenario of WF attacks. In it, the victim establishes an encrypted tunnel to the SSH proxy, which encrypts and encapsulates all web traffic. The adversary only can sniffer and record the encrypted traffic between the victim and proxy. In order to distinguish whether the encrypted traffic is generated by a targeted page, the adversary needs to compare it with his known traffic data. If some pair is matched, he can infer the victim's retrieving activities.

The adversary implements a WF attack in two phases: in the training phase, he creates a large fingerprint database of targeted websites through collecting encrypted traffic in a similar network environment with the victim. Then he trains an appropriate classifier with the labeled fingerprint data. In the testing phase, the adversary eavesdrops on the victim's traffic and identifies it with the trained classifier.

To realize the above attack, the adversary is always assumed to have some capabilities as follows:

- The adversary has enough background knowledge about the victim. For example, he can get details about the anonymous technology used by the victim, including the software version, physical location and so on. Besides, he can infer an interesting list of websites which the victim will visit.
- Some functions of the victim's browser are limited, e.g. turn off caching and automatic updating.

- The adversary is able to configure a similar network access like the victim and stores a large number of traffic fingerprints for the interesting websites.
- The adversary is able to extract all traffic belonging to an individual page. That is to say, even though some background traffic is produced, the adversary also can separate the overlapped traffic and get traces respectively.
- The adversary is able to discover the beginning and ending of one page's load so that he can get an integral trace to draw the fingerprint.

B. Assumptions Evaluation

Obviously, the aforementioned assumptions shape an extremely powerful character for the adversary except that he can decrypt the traffic. Consequently, many researchers criticize the practical feasibility of the WF attacks. From a practical point of view, we divide these assumptions into two classes.

Reasonable Assumptions. We consider the first three assumptions to be reasonable. In the real world, by utilizing some out-of-band methods, an adversary can get a lot of background knowledge about the victim. For instance, through physical contact or a long-term observation, the adversary can obtain the victim's browsing history and infer the interesting list with a certain probability. As for the limitations of turning off the browser's cache and automatic updates, they are not the default options and may violate the victim's preferences. However, if the victim wants to visit some sensitive websites or uses a public computer, he may set his browser in incognito mode, which will disable these functions. To configure a similar network access, the adversary can select the same SSH proxy used by the victim. Considering the prices of storage devices and free cloud storage services, the adversary can create a large fingerprint database with low cost.

Unreasonable Assumptions. We consider the last two assumptions to be unreasonable. When the victim opens multiple tabs simultaneously, all packets generated by different pages are encrypted and forwarded through the same tunnel. There is no mature technology can help people to decide whether a packet belongs to the targeted page. Besides, the adversary also can't discover the beginning and ending of one page's load. Only when all packets are transmitted, can he determine that one page has been completely loaded. To evade these issues during the experiments, researchers must promise that only one page is visited each time and no background traffic is generated. But the findings of Mozilla [16] indicate that vast majority of users open two tabs at a time in practice. So in the practical scenarios, the adversary can't separate the overlapped traffic and get a high detection rate.

Given the last two unreasonable assumptions, we relax them by allowing the victim to load another page in the background. In this new threat model, traffic belongs to different pages may be overlapped. The adversary can't extract all traffic belonging to an individual page to draw an integral fingerprint.

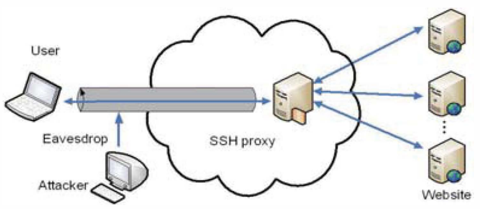

Figure 1. Threat Model of the WF attack

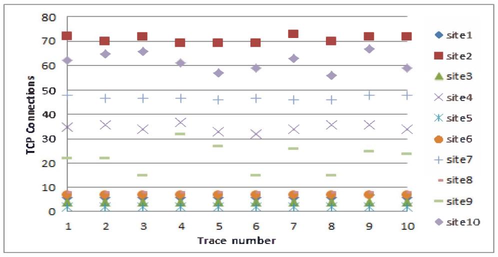

Figure 2. Distribution of Tcp connections of Top 10 sites

IV. Website Fingerprinting Attack

Since the victim is allowed to open two pages simultaneously in our threat model, we need to perform the data preprocessing to determine whether the traffic is overlapped, and then extract appropriate features to create fingerprints. Finally, we identify two pages with the Naive Bayes Classifier respectively.

A. Data Pre-procession

The essence of the WF attack is a pattern matching problem, which will be affected by noise data. So in the data pre-procession, we filter broken train samples to create integrated traffic templates. Since the overlapped traffic considered in our model is generated by two pages and many features are obscured, we can't achieve satisfying classification accuracy with previous works. We need to determine whether the unknown traffic is overlapped before extracting appropriate features. With regard to the availability and efficiency, we select the following features to discover overlapped traffic:

- **TCP Connections**. According to the SSH Connection Protocol, when the client wants to establish a new TCP connection, he sends the SSH_MSG_CHANNEL_OPEN message to open a new channel. So we can get the number of TCP connections by counting the open channel message whose size is 96 in Linux. By accessing top ten targeted websites repeatedly, we find that the number of TCP connections is a stable and discrimination feature, as shown in Fig. 2.
- **Total Per-direction Bandwidth**. We add all the sizes of incoming and outgoing packets separately to count the total bandwidth consumed per-direction. If the structure and embedded objects of one page are not dramatically changed, the feature is always stable. In

addition, due to the trend of content-rich websites, the value is far from other pages.

- **Inter-packet Time**. We obtain the Inter-packet time by computing the delays between the adjacent packets with payload. If the inter-packet time exceeds a certain value, we will believe that the traffic of the first page is completely transmitted and the long think time causes this idle time. So we assign this feature with value 1, otherwise 0.

If we get all the features from the traffic sample, we can denote them as:

$$\overrightarrow{X} =< c, b_1, b_2, t > \tag{1}$$

In the above, c is the number of TCP connections; b_1 is the outgoing bandwidth while b_2 is the incoming bandwidth; t represents the feature of inter-packet time. When t is assigned with value 1, we can determine that the unknown traffic is overlapped. Otherwise, we need to use the Mahalanobis Distance to calculate the similarity between the unknown traffic sample $\overrightarrow{X}$ and training instances:

$$MD(\overrightarrow{X}) = \min_i(\sqrt{\frac{(X_c - Y_{ic})^2}{S_c^{\,2}} + \frac{(X_{b_1} - Y_{ib_1})^2}{S_{b_1}^{\,2}} \frac{(X_{b_2} - Y_{ib_2})^2}{S_{b_2}^{\,2}}}) \tag{2}$$

S is the corresponding standard deviation over the training set. If the maximum similarity is less than the threshold, $\overrightarrow{X}$ is determined to be overlapped traffic.

B. Feature Extraction and Classification

When we confirm that the unknown traffic is generated by two pages, we extract different features and identify fingerprints respectively.

1) The First Loaded Page

According to the users' retrieval pattern, the victim visits the second page after some period of idle. This idle is called think time, which is consumed to digest the content of the first loaded page or select (or enter) a new link. In this idle the transmitted traffic of first page is not affected, so we can extract a few fine-grained features to do classification:

- **Without Packets Sized Zero**. The sized zero packets are TCP ACK packets which are present in all HTTP traffic. These packets can't provide useful information for identification and may confuse other features. So they should be filtered in the experiment.

- **RTT**. We calculate the delay between the first Get packet and the first Response packet. This feature can reflect some location information of the web server.

- **First Get packet Size**. The first Get packet is a request to the html document and its size is partly decided by the length of URL. It won't be dramatically changed unless the URL of website is altered.

- **Html Document Size**. The html document is used to describe the structure and content of the web page. Its size is a stable value even though some objects are changed. In order to get this feature, we add all the sizes of packets responding to the first GET packet. Assuming the second GET packet is sent at time *t*, we will calculate all the incoming packets received before (*t* + RTT).

- **Per-direction Packet Ordering**. The structure of the page induces a logical order in its packet sequences. Taking into account that the overlapped traffic may disrupt the ordering information, we just extract the per-direction ordered packet sequences transmitted in the idle.

2) The Second Loaded Page

We choose coarse features to classify the second loaded page. The basic idea is excluding the first page's traffic from the overlapped instance and extracting statistics information to identify the remaining traffic.

- **TCP Connections.** This feature is the same as described above.

- **Total Per-direction Number of Packets.** We count the packets with payload per-direction and use them as the classification features.

- **Total Per-direction Bandwidth.** This feature is the same as described above.

3) Classification

Based on the selected features above, we further apply the Naive Bayes Classifier to identify two pages respectively. The Naïve Bayes Classifier assumes independence between all attributes and estimates the probability of a vector $\overrightarrow{f} =< f_1, f_2 ... f_n >$ belonging to a particular class C_i as:

$$\begin{aligned} &p_{C_i}(f) = p(C_i|f) \\ &= \frac{p(C_i)p(f|C_i)}{p(f)} \propto p(c_i)\prod_{j=1}^{n} p(f_j|C_i) \end{aligned} \tag{3}$$

V. Evaluation

In this section, we mainly describe the experiment methodology used in the real world environment to evaluate the performance. According to the previous works, we use the

TABLE I. DATASETS USED FOR DIFFERENT EXPERIMENT SCENARIOS

Experiment scenario	**Usage of datasets**	
	Training Dataset	**Testing Dataset**
Closed-world	*one_page*	*two_pages*
Open-world	*one_page*	*first_false , second_false*

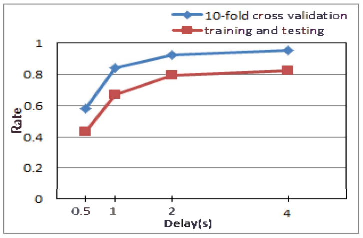

Figure 3. Detection rates of different values of the delay

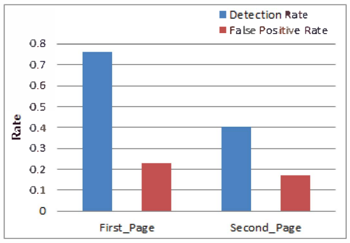

Figure 4. Detection rates and false positive rates of different patterns

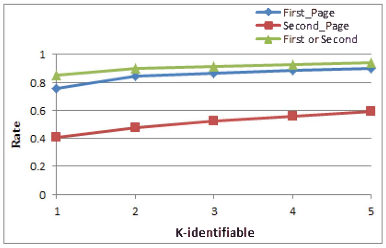

Figure 5. Detection rates of different values of K

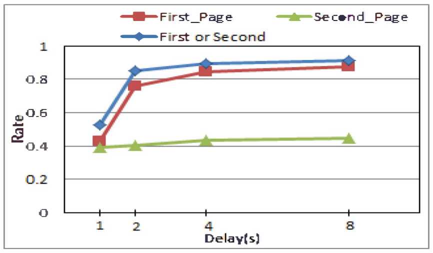

Figure 6. Detection rates of different values of the delay

detection rate (accuracy) and false positive rate as evaluation criteria.

A. Experimental Setup and Data Collection

According to the Zipf distribution, we choose 50 top ranked websites in Alexa to construct the interesting list. In the Closed-world scenario, we construct two datasets based on different crawl pattern with tcpdump, including the *one_page* dataset and the *two_pages* dataset. The *one_page* dataset is generated by accessing 50 targeted websites repeatedly without any background traffic. But when we build the *two_pages* dataset, we access two targeted pages simultaneously and the background page is loaded with a delay. So as to cover all the combinations of two pages, we generate 2,500 records in each batch. In the Open-world scenario, to evaluate the false positive rate, we construct an uninteresting list with 50 other sites and build two more datasets called *first_false* and *second_false*. During the construction of the *first_false* dataset, we access an uninteresting page and then load an interesting page. The *second_false* dataset is built in the reverse order. The usage of different datasets is shown in Table 1.

The experimental setup is shown in Fig. 1. We deploy our attack in real word environment and the experiment lasts for two months. All the traffic is generated by using Mozilla Firefox 18.02 running on Ubuntu 12.04 to retrieve websites via an Openssh proxy. We utilize a browser extension called Pagestates to realize the automatic website accessing. Similar to previous works, we turn off the browser's cache and disable the automatic update function. However, the active contents such as Flash, Java and JavaScript are allowed in our experiment, which made the browser's configuration closer to the normal ones. In the classification phase, we use Weka[18] toolkit to implement the Naive Bayes Classifier. Besides, we modify the source code to calculate the accuracy with several guesses.

B. Experiment Results

To correctly identify two pages visited by a user, the key point is that we need to obtain enough information from traffic transmitted in the short delay and infer the identity of the first page. Intuitively, the longer delay the higher accuracy. However, too long delay is not realistic considering the user's think time. To get an appropriate vale of this parameter, we respectively use 10-fold cross validation and training and testing pattern on the *one_page* dataset. Fig. 3 illustrates the detection rates in terms of the length of the delay. We can see that the detection rate of 10-fold cross validation is up to 92.4% when the delay is set to 2 seconds. In addition, according the distribution of users' think time [19], we set the delay to 2 seconds in the following experiment. To simulate the practical scenarios, we train the classifier on the *one_page* dataset and test data from the *two_pages* dataset. As shown in Fig. 4, when the delay is set to 2 seconds, the accuracy of the first page is 75.9%. If we know the real identity of the first page, we can classify the second page with the accuracy of 40.5%. And the false positive rates are 22.9% and 17.4% respectively.

Fig. 5 illustrates the detection rates of three patterns with *K* guesses. The pattern called "First or Second" means that we consider the classifier successful if it can identify either the first page or the second page. The value of *K* is increased from 1 to 5, which represents the size of the ranking list of candidates. In other words, if the real identity of a page is in the ranking list of candidates, we consider the classifier successful. As we can see form Fig. 5, the trend of two curves in the upper portion of the figure are so similar. It means that correctly classifying the first page is necessary for identifying the second page. When *K* is set to 1, the accuracy of the "First or Second" pattern is 84.9%. Similarly, Juarez et al. [1] carried out several WF attacks on multi-tab traces collected in Tor. When the size of the interesting list is set to 32, the highest accuracy is only 10%. When it is raised to 64, the highest accuracy is decreased to 7%.

As mentioned above, we set the delay to 2 seconds in our automatic data collection procedure. But when we artificially view the traffic records from the *two_pages* dataset, we find that the exact value of the delay between two pages is always less than 2 seconds. There may be many reasons, e.g. the response delay of the browser and tcpdump. This finding means that the detection rates of our attack may be underestimated. In order to verify this conjecture, we change the value of the delay in the procedure. But in the feature extraction, we still analysis packets in 2 seconds. Fig. 6 illustrates the detection rates with different values of the delay. When the delay is set to 4, we can see an increase in the accuracy.

VI. Conclusions

The WF attack is always criticized for its unrealistic assumptions, especially restricting the user browsing without any background traffic. Consequently, people describe it as a paper tiger. To address this problem, we relax the assumptions and propose a novel WF attack aiming at the multi-tab browsing behavior in this paper. When the delay is set to 2 seconds, our attack can classify the first page with 75.9% accuracy, and the second page is 40.5%. Based on the results, we believe that the WF attacks still pose a significant threat to users' privacy and should be carefully considered without contempt. For future research, we intend to look for some countermeasures to effectively defend against the WF attack.

Acknowledgment

This work is supported by National Natural Science Foundation of China under Grants No. 61272054, No. 61320106007, National High Technology Research and Development Program of China under Grants No.2013AA013503, China Specialized Research Fund for the Doctoral Program of Higher Education under Grants No. 20110092130002, State Grid Corporation of China-Research and Application of Security Isolation and Active Defense Technology for Information Network Border, the Fundamental Research Funds for the Central Universities (No. 2242014R30010), Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

References

[1] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A Critical Evaluation of Website Fingerprinting Attacks", in Proc. of the 21st ACM Conference on Computer and Communications Security (CCS 2014), Scottsdale, Arizona, USA, 2014.

[2] M. Perry, "A Critique of Website Fingerprinting Attacks", https://blog.torproject.org/blog/critique-website-traffic-fingerprinting-attacks, November 2013.

[3] A. Hintz, "Fingerprinting Websites Using Traffic Analysis", in Proc. of Privacy Enhancing Technologies Workshop (PET 2002), LNCS 2482, 2002, pp. 171-178.

[4] Q. Sun, D. R. Simon, Y.M. Wang, W. Russell, V. N. Padmanabhan and L. Qiu, "Statistical identification of encrypted web browsing traffic", in Proc. of the 2002 IEEE Symposium on Security and Privacy (IEEE S&P), 2002, pp. 19-30.

[5] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy vulnerabilities in encrypted HTTP streams", in Proc. of Privacy Enhancing Technologies Workshop (PET 2005), LNCS 3856, 2005, pp. 1-11.

[6] M. Liberatore and B. N. Levine, "Inferring the source of encrypted HTTP connections", in Proc. of the 13th ACM Conference on Computer and Communications Security (CCS 2006), Alexandria, Virginia, USA, 2006, pp. 255-263.

[7] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier", in Proc. of the 2009 ACM CCS Workshop on Cloud Computing Security, Chicago, Illinois, USA, 2009, pp. 31-42.

[8] L. M. Lu, E. C. Chang and M. Chan, "Website fingerprinting and identification using ordered feature sequences", in Proc. of the 15th European conference on Research in computer security, LNCS 6345, 2010, pp. 199-214.

[9] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing-based anonymization networks", in Proc. of the ACM CCS Workshop on Privacy in the Electronic Society, Chicago, Illinois, USA, 2011, pp. 103-114.

[10] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a Distance: Website Fingerprinting Attacks and Defenses", in Proc. Of the 19th ACM Conference on Computer and Communications Security (CCS 2012), Raleigh, NC, USA, 2012, pp. 605-616.

[11] T. Wang and I. Goldberg, "Improved Website Fingerprinting on Tor", in Proc. of the 13th WPES, Berlin, Germany, 2013.

[12] C. Wright, S. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense against Statistical Traffic Analysis", in Proc. of the 14th Annual Network and Distributed Systems Symposium (NDSS), San Diego, California, USA, 2009, pp. 1-14.

[13] X. Luo, P. Zhou, E. Chan, W. Lee, R. Chang, and R. Perdisci, "HTTPOS: Sealing information leaks with browser-side obfuscation of encrypted flows", in Proc. of the Network and Distributed Security Symposium (NDSS), San Diego, California, USA, 2011, pp. 1-20.

[14] K. P. Dyer, S. E. Coull, T.Ristenpart and T. Shrimpton, "Peek-a-Boo, I Still See You: Why efficient traffic analysis countermeasures fail", in Proc. of the 2012 IEEE Symposium on Security and Privacy (IEEE S&P), 2012.

[15] X. Cai, R. Nithyanand, T. Wang, R. Johnson and I. Goldberg, "A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses", in Proc. of the 21st ACM Conference on Computer and Communications Security (CCS 2014), Scottsdale, Arizona, USA, 2014.

[16] Mozilla Labs, "Test Pilot: Tab Open/Close Study:Results", https://testpilot.mozillalabs.com/testcases/tab-open-close/results.html#minmax.

[17] D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval", in Proc. of the 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 4-15.

[18] I. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques", Second Edition (Morgan Kaufmann Series in Data Management System). Morgan Kaufmann, USA, 2005.

[19] F. D. Smith, F. H. Campos, K. Jeffay and D. Ott, "What TCP/IP Protocol Headers Can Tell Us About the Web", in Proc. of the ACM SIGMETRICS 2001/ Performance 2001, Cambridge, MA, 2001, pp 245-256.

[20] T. Wang, X. Cai, R. Nithyanand and I. Goldberg, "Effective Attacks and Provable Defenses for Website Fingerprinting", in Proc. of the 23rd USENIX Security Symposium, San Diego, CA, 2014.