BBM461 Research Report – Spring 2021

Name: Koray KARA

Student ID: 21803682

Report Subject: Website Fingerprinting

List of the reported papers:

1) Hayes, J., & Danezis, G. (2016). k-fingerprinting: a robust scalable website fingerprinting technique. In 25th USENIX Security Symposium (pp. 1187-1203).

2) Wang, T., & Goldberg, I. (2017). Walkie-Talkie: an efficient defense against passive website fingerprinting attacks. In 26th USENIX Security Symposium (pp. 1375-1390).

3) Vastel, A., Laperdrix, P., Rudametkin, W., & Rouvoy, R. (2018). Fp-Scanner: the privacy implications of browser fingerprint inconsistencies. In 27th USENIX Security Symposium (pp. 135-150).

4) Shusterman, A., Kang, L., Haskal, Y., Meltser, Y., Mittal, P., Oren, Y., & Yarom, Y. (2019). Robust website fingerprinting through the cache occupancy channel. In 28th USENIX Security Symposium (pp. 639-656).

5) Gong, J., & Wang, T. (2020). Zero-delay lightweight defenses against website fingerprinting. In 29th USENIX Security Symposium (pp. 717-734).

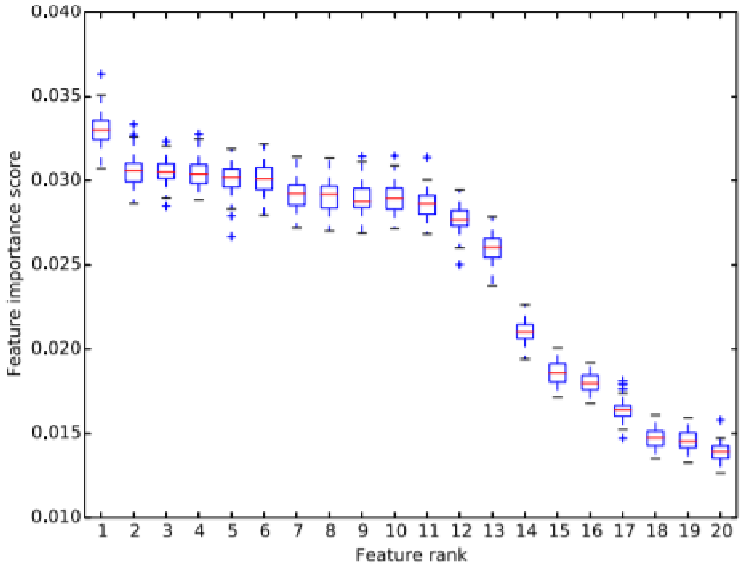
1) k-fingerprinting: A Robust Scalable Website Fingerprinting Technique

In this paper, they introduce the website fingerprinting attack and how the website fingerprinting work. Unlike the classical fingerprint method, they found a method called k-fingerprinting. This method is based on "Random Decision Tree" algorithm, which is one of the machine learning algorithms. At the same time, they offer a stronger attack than existing attacks with this method. Website fingerprinting attack is carried out simply as follows. The attacker first traces the data traffic paths to understand which of the specific sites a user has entered. It extracts and trains the features of these sites using machine learning algorithms. In this way, the attacker understands which sites a client has visited. They explain some of the weak features of this method as follows. The user only needs to access a limited number of websites, but this is unlikely in real life.

As mentioned in Paper, with the "Random Forest" algorithm, features were extracted from the websites that a user entered, and this attack was achieved with a high success. They tried this method on many websites and thought to train only part of the data to achieve an attack with a high success rate and obtain a low-cost attack. They state that searching through Tor would not be of much use to reduce this attack to a minimum level of success, and that k-fingerprint attack was highly successful on these services. They have also mentioned about some studies against this attack. Some of these are a defense mechanism called "traffic morphing" in 2009. This mechanism was highly successful over the first fingerprinting attacks. Other defense mechanisms have been developed in a way that forces these attacks by increasing the number of layers within the website.

In their work, they created a feature using the "Random Forest" algorithm for each data traffic on their website. These are called fingerprint vectors. They have calculated the distance between these vectors with a metric called Hamming distance. By using the closest k of these distances, a classification can be made as desired by the attacker. They performed the K-fingerprinting attack in 2 stages. First, the attacker chooses a few websites to monitor the network traffic, and these are trained by the "Random Forest" algorithm. Using this trained model, the user collects session information. Fingerprint vectors are created with these collected data. The attacker then calculates the Hamming Distance of the vectors that were trained with these fingerprint vectors and understands whether the website is visited by the client according to the distribution of the nearest k fingerprint.

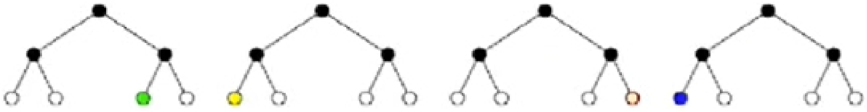
They have used approximately one-hundred fifty features, but they have found about twenty or thirty features of them is relatively quite important because these are determinative for training phase. The related feature importance graph is shown in the picture below.



They created this graph to observe the "best" features and determined that these features were the number of packets (inbound and outbound) and information leaked a few seconds after a web page was loaded. This feature has enough information to achieve an attack. In order to achieve the k-fingerprinting attack, they convert the problem into a classification task. For example, given a loaded website, they have recorded the network traffic that is generated by that load so they can record the time and sent/received packets. Then, they convert it into feature set and pass it to the Random Forest algorithm. This algorithm is ensemble decision trees and tries to find the best splitted data. The leaf of that tree will be a category of the data. Random Forest has a bunch of decision trees and there is a mechanism inside itself that determine how the input will be classified. Instead of just using Random Forest for classification, they also used it to create fingerprints. They have basically looked at the position of the leaf nodes and take this position as the fingerprints for the input. After all of the creation, they have used K-NN algorithm to classify the input. For example, when they have a website and they want to classify it, they basically look at the 5 nearest neighbors to the input and If all of the nearest neighbors also agree on how to classify the input, the classification of the website is done successfully. Otherwise, they have labeled this website as the unmonitored website.

The usage of Random Forest with its leaf nodes as shown in the figure below.

The output of this random forest will be used as the fingerprint of the website load.



In order to evaluate the attack, they have used some accuracy metrics, such as TPR, FPR, and BDR. The explanation of the first two is correctly, and falsely classified. BDR is based on some conditional probability metrics. It considers both TPR and FPR and gives a real idea that how the attack is actually performing. It tells the probability of the correct prediction of the classification.

They have used parameter tuning to increase the model accuracy and they have observed that the attack does not really benefit from adding more trees after they hit the certain number of trees. So, this attempt does not affect very much to increase the accuracy after adding one hundred twenty trees.

While they are testing their results with this attack, they firstly worked on Alexa monitored set and it is observed that the results are not so great. After that, they have looked at the Tor hidden service monitored set results, they have observed that the true positive rate drops about ten percent regardless of the number of unmonitored web sites. For the false positive rate case, it drops dramatically to about 0.3%. That means if they test 200 or 300 web pages, they will only have false positive in the order of ten instead of hundreds or thousands which the attacker will be able to attack on these pages.

They have also mentioned the limitations of this work. One of these limitations is that there are many unmonitored and monitored websites, and they understand with a random probability which of these sites a specific person has accessed. So they have to decide on the web sites that the people may visit then remove some of these sites from the dataset to have a much better results. Another limitation is that given some stream of the data, they assumed the attacker can perfectly know the start and stop time of the web site loading and the client does not listen to music or download something in the background that may create noise. However, in the real-life scenario, people do multiple tab browsing in the background. By considering these realistic scenarios it is right to say that the website fingerprinting attack might not reflect practical risks.

In conclusion, this attack is a highly successful type of an attack on many web pages. In order to get much better results, the deep learning methods can be used instead of just using the basic machine learning algorithms. It is also found that there is a difference between Tor Hidden Services and normal websites. That means that for a given network traffic, it is easily told whether the Tor hidden service is browsed or not.

2) Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks

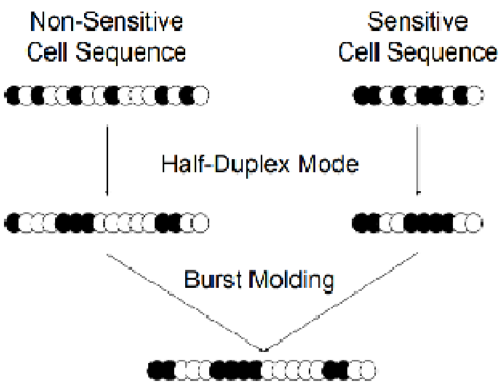
In this paper, the new defense against website fingerprinting is introduced that is called Walkie-Talkie defense. They state that WF attacks will be successful even if some hidden technologies are used. For example, they say that Tor's defense mechanism is not very effective in reducing the success of the WF attack. For this reason, the researchers proposed alternative defense methods. However, these methods did not show much success against the new attacks developed. The presented Walkie-Talkie (WT) defense method has been developed to be successful in this regard. Some of the features of this defense are that it is effective against many newly developed WF attacks and the network is protected efficiently by reducing the band with overhead. The WT defense mechanism is also very easy to use. They implemented the system only by changing the application layer so that it would not affect the server performance much. In addition, while this system is running in the background, other users have been made independent from each other. In this way, more than one client can use this mechanism simultaneously in a reliable manner.

In the website fingerprinting attack, it is stated that the attacker tries to perform the attack by using web pages trained package sequences and some machine learning algorithms. In a scenario that is far from the real-life scenario, these packet sequences come from where the attacker knows and monitors, and the attacker can easily distinguish these packets. However, in a real-life scenario, these packets can come from web pages that are unknown by the attacker. In that sense, it is stated that the attacker should be able to distinguish the web pages in terms of sensitivity. Many researchers have developed noise-resistant attacks on this issue and have developed attacks that can deal with real-life scenarios and pose many threats to privacy.

In this work, it is also stated that one of the studies for these attacks is the "traffic morphing" defense. In this defense, the package length was changed randomly to create the impression that these packages came from another website. This paper concluded that this defense was successful only against the first attempted attacks because this method is based on changing the packet length only, regardless of the sequence order of the packets.

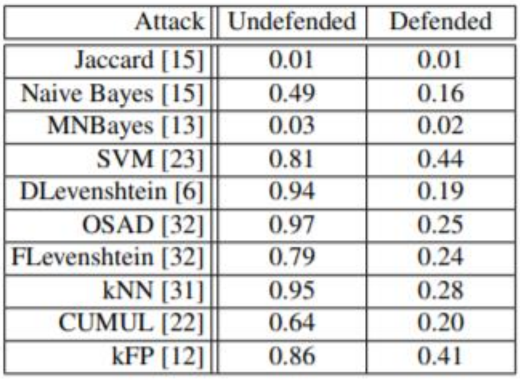
Another defense is HTTPOS defense. In this defense, the client adds a new field called "Range" to the header section. This field is used to divide the network traffic into random packets. However, it has been observed that this defense is not also very effective on newly developed attacks. One of the defenses implemented against WF attack is Tor’s defense. This mechanism uses HTTP pipelining and thus changes the order of requests in the pipeline. This mechanism is also stated to be less effective against known attacks.

Walkie-talkie mechanism were designed to be an agnostic defense and they have tried to minimize the bandwidth and time values. The web browsers are normally work with full-duplex page loading but instead of using this, they have changed the web browsers for Walkie-talkie model, so that they instead enforce a half-duplex policy in that model. The start of these duplexes is same but the difference of these two is that they wait for the entire page to come before the request that is made. That means they do not talk while the page is still loading. After that they request for the page and queue up all the resources and send all of the requests at the same time when the whole page is done. In this way they ensure that each side does not talk while the other side was talking. So, it is kind of like a walkie-talkie. By enforcing half duplex mode on self-sequences, they basically grouped all of the packets (incoming and outgoing) into bursts.



As shown in the picture above, in the half-duplex mode, the first cell is a request for the page and then the following cells are bunch of cells that represents the page, the following cells are a bunch of cells that represent the resource of the page. By grouping all of these bursts, they have the model that make sense. In order to implement and enforce this half-duplex mode, they have changed the Tor client, or the client can add fake cells to make herself/himself look like some dummy pages.

For the evaluation of the Walkie-Talkie, they compare the accuracies before and after the attack attempt on many web pages. So, they dived the pages into 100 classes, and they have evaluated the scheme against the known attacks for both undefended and defended case.



Attack	Undefended	Defended
Jaccard [15]	0.01	0.01
Naive Bayes [15]	0.49	0.16
MNBayes [13]	0.03	0.02
SVM [23]	0.81	0.44
DLevenshtein [6]	0.94	0.19
OSAD [32]	0.97	0.25
FLevenshtein [32]	0.79	0.24
kNN [31]	0.95	0.28
CUMUL [22]	0.64	0.20
kFP [12]	0.86	0.41

As seen in the picture above, they have observed that the maximum attack accuracy for such a scheme is approximately 50%. This is actually a proof of work that a scheme is really working. They have interpreted the table like this. There is no attack in the defended column that achieves a accuracy over %50.

They have also compared their defense against known defenses and found that by using Walkie-Talkie, the overhead is much smaller and stable as well. The reason of this they have used half-duplex mode and it ensures that the sequences are identical.

In conclusion, the new defense is created that is called Walkie-Talkie by enforcing half duplex mode. It was achieved by adding a structure to how a page is loaded. It is also a good defense that works against all of the passive website fingerprinting attacks and also it has very low overhead.

3) Fp-Scanner: The Privacy Implications of Browser Fingerprint Inconsistencies

In this paper, they have researched how user privacy is affected with changed fingerprints and how methods should be developed in order to prevent inconsistencies that may occur. In order to do this research, they offer a test package called FP-SCANNER. With this package, they can easily find fingerprint inconsistencies in the browser and their negative effects. In addition, thanks to this test suite, they can reveal the origin of the modified fingerprint features. According to some researches, user tracking has become very popular among websites. One of the techniques that is currently developed is the Unique User Identifier (UUID) designed to store the user's information such as cookies. However, some hidden modes have been developed to protect the user. These modes automatically delete the cookie information of the user and reduce the efficiency of some standard user tracking methods. Because of the browser fingerprinting is stateless, some unwanted vulnerabilities may arise. Researchers have developed some methods for this. These methods mostly based on changing the fingerprint to hide the user's original identity. However, this method causes an inconsistent number of fingerprints in the system. Some of the researches in this study have been expanded to reveal these inconsistencies. One of them is called Nikiforakis. This work is mostly relying on finding some irregularities by considering big number of browser fingerprint precautions.

Due to the privacy problems that is caused by browser fingerprint tracking methods, many precautionary methods have been developed on this issue. This study focuses on 5 of these methods. One of these methods is the script blocking method which is developed to block some fingerprint scripts. The goal of this strategy is breaking the collection of the fingerprint. One of the other method is attribute blocking. That method is used for blocking access to a specific function that can get the value of a canvas in order to decrease the entropy. To switch the values of some attributes by changing their values and adding some noise to data, attribute switching method can be used. The aim of this method is simply breaking the stability. There is also similar method that aim the same purpose called Attribute Blurring. It is also used for breaking stability that is required to track using fingerprinting.

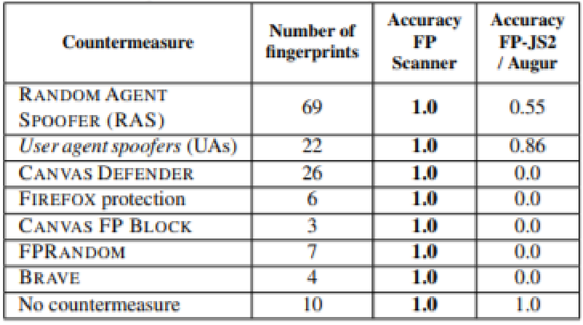
There are also some kind of tools that are very convenient for browsers such as browser extensions and forked browsers. The problem of the countermeasures is that the finger printers might be able to detect them. For example one of these finger printers that is called “Augur”. The usage of this can be a problem because it is used generally to track people. There are some researches to detect the presence of user agent spoofers. For a real example, the user agent can say the used platform is Windows although the platform says Linux. This issue is called fingerprint inconsistency and it can be used to show up the presence of countermeasures.

In order to understand whether the attributes of a fingerprint have been modified or not, they have introduced FP-Scanner by expanding all kinds of countermeasures. The inconsistencies are splitted into four components and all of these components are discussed separately. One of these components is OS inconsistencies. They verify the system by using Navigator Platform. Also WebGL can be used to generate 3d shapes in the browser. These values can be used to verify if the OS that is displayed in the user agent is consistent or not. Second component is Browser inconsistencies. In this case, the error is simply thrown like stack overflow error. The browser features can also be used to test whether the version has been modified or not. The third component is Device Inconsistencies. In this case, the goal to be achieved is to verify a device that claimed to be a smartphone or computer. In order to achieve this goal, some events and sensors are used for testing. Finally, the fourth component is canvas inconsistencies. It is an attribute that has a high entropy, and that entropy depends on the device or browser on the OS. It also an important attribute for tracking so it has high stability. This component is used mostly to detect different countermeasure as modifying the canvas. It is also used to verify if the background is transparent, whether it is an isolated pixel, and what this number of pixels should be per color.

For the evaluation case, seven countermeasures are evaluated. Some of them are the extensions like Canvas Defender, Canvas FP block. Also FP-Random is used to modify the canvas in a more consistent way. In order to consistently make an evaluation, the Firefox protection and Brave are tested for fingerprinting protection.

For the results of this study is shown in the picture below. The countermeasures are presented in each line. It is observed that the presence of countermeasures is always be detected however it can be seen that Random Agent Spoofer is outperformed because it tries to lie in a more consistent way but it can still be detected with high accuracy.

Table 8: Comparison of accuracies per countermeasures



Countermeasure	Number of fingerprints	Accuracy FP Scanner	Accuracy FP-JS2 / Augur
Random Agent Spoofer (RAS)	69	**1.0**	0.55
User agent spoofers (UAs)	22	**1.0**	0.86
Canvas Defender	26	**1.0**	0.0
Firefox protection	6	**1.0**	0.0
Canvas FP Block	3	**1.0**	0.0
FPRandom	7	**1.0**	0.0
Brave	4	**1.0**	0.0
No countermeasure	10	**1.0**	1.0

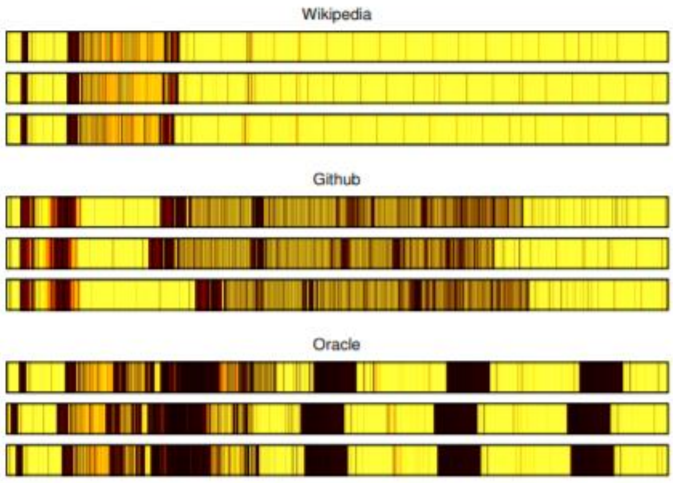
The privacy implications are also observed. The most important implication is trackability. Having fingerprinting countermeasure make the user very easily trackable. Since detecting a countermeasure is not enough for making tracking easier, this implication depends on different factors. The first factor is the ability to identify the countermeasure more accurately. The second factor can be number of users because in some cases identifying the countermeasures may not be used by many users. So, detecting with them is highly discriminating. The third factor is the ability to recover original values for easily tracking. The final factor is quantity of information leaked by countermeasure.

In conclusion, in this study, fingerprinting countermeasures and some techniques to detect extensions are clearly explained. Countermeasures can be detected by fingerprinters by using inconsistencies. In terms of privacy implications, discrimination and tracking have explained and what they depend on have also been explained.

4) Robust Website Fingerprinting Through the Cache Occupancy Channel

This paper focuses on the attacks that allow malicious users to track all traffic information of a normal user's computer, even if tools are used to protect users' privacy. It is stated that in the network, any of the eavesdropper can easily monitor the source and the destination of the network traffic. After that he/she can reveal some secret information about the victim. So, there are several privacy protecting tools are introduced such as Tor browser. In that browser, the source and the destination are not in the network traffic. So, the "on-path adversary" cannot identify which web site the victim user surfs. There are many researches that are trying to find unique website fingerprinting. In that regard, some of the website objects like pictures, scripts and videos are translated into some features such as packet length, timing, and direction. These features will be used for identifying which website the user have been visited. In order to achieve this identification, some of the machine learning tools and statistics are used with these extracted features. To make a planned remote attack successful, a new model is created and proposed in this work. This model assumes that the attacker can inject some advertisement in the browser of the users. These advertisements may be located in the background window or the same page. It runs on the same architecture with the target computer. The mentioned attack works as follows. The attacker sends an advertisement to the target computer. Then the victim user surf on this private website. While the user surfing in that website, he/she leaves the footprints on the cache. The attacker waits for a while to make sure that the victim user is really executing. Then by using this malicious advertisement, the attacker does the "Prime and Probe attack" and reads the cache. After that, sends the cache information to the adversary. By using this attack, the attacker has collected more than 20 datasets and 100 traces for each of the websites. These collected data have been used to train on deep learning model. The obtained accuracies are 87% accuracy in Firefox and 47% accuracy in Tor. After all of these works, they have observed that reducing time resolution does not eliminate the threat.

In the Prime and Probe method, there is a memory buffer that the adversary allocates. When it accesses every cache in the loop, then the victims use his processes and adversary uses the timing attack and the adversary can identify some features about these processes. In the picture below, traces of the cache are observed while the user surfing different websites.



The darker points represent the largest cache contention in the time slot. As seen the various websites above, the differences between them can be easily identified. It has been done by using some Machine Learning models.

It is mentioned that these works have some problems. JavaScript timer resolution has been dropping rapidly over the years. These low time resolutions make the cash attack (Prime and Probe) really useless. In order to find a solution to this problem, they have measured the contention over the whole last level cache over time instead of measuring the contentions on each cash set. This solution makes the traces more usable for the model. For example, they have obtained the low resolution in Tor as 100 ms just by counting the number of whole cache probes.

To create evaluation model, they have collected lots of data from different websites in different countries. Then, they have trained a deep learning model on this traces and evaluated the accuracy on different test set. In the data collection set up, the browser agent visits several websites and they have collected the cash attack trace in every one of those visits of websites. After that they have applied many machine learning models.

For the attacks, they have used Native Application & JavaScript attack, cache attack. Also, they used different browsers like Chrome, Safari, Firefox and different OS's like Linux, Windows, MacOS. Closed and Open World datasets have been used as a data set types. The closed one have a set of sensitive websites that the attacker want to identify. On the other hand, the open world dataset has more websites in the real world. An attacker can use it if he/she want to determine whether they are sensitive or insensitive websites. Another dataset type that is used is Network fingerprinting countermeasures. They have just tried to minimize the leak from the features of packet size, and direction of communication by using packet padding and traffic molding to drop the classifier accuracy to base rate.

For the results, they have increased the accuracy from 72% to 90% for the closed world dataset in regular browsers. For the open world one, the accuracy is increased %72 to %87. For the Tor browser, approximately %46 accuracy is obtained for the closed word dataset, and approximately %62 accuracy is obtained for the open world dataset. They have proposed some countermeasures. One of them is cache activity masking. It has an impact of about %5 on the performance of computer. It is achieved by allocating the buffer that is cache sized and accessing all of the cache line in the loop. Although it seems to be attack, it is actually a countermeasure. By using this countermeasure, Tor classifier drops the accuracy to base rate but the accuracy reduction rate for regular browser classifiers is about 10%.

In conclusion, they have created unprivileged code that can run on the machine of victim user. They can beat the low resolution and they have also introduced an effective countermeasure for this work.

5) Zero-delay Lightweight Defenses against Website Fingerprinting

In this paper, they have proposed and also evaluated some website fingerprinting defenses against the attacks. As an introduction, it is stated that network surveillance is one of the major threats to privacy of the people. There are some anonymous networks developed to prevent this threatening problem. One of them is Tor. This network provides defenses to ensure the privacy of users browsing the internet. In order to implement these defense mechanisms, it sends user packets through a number of proxies. In this way, it prevents the surveillance of the visibility of target and source packets in the network. However, some of the researches have realized that a attacker that performs his/her attack locally is able to violate the users' privacy by observing the traffic of the network passively. Since different web pages can have various loading times and so many different numbers of packets, this information can be used to fingerprint the websites by breaking the privacy. This attack is generally known as website fingerprinting. That means encrypting the packets does not provide much benefit. Any person on the same network or Internet Service Provider (ISP) both can be a website fingerprinting attacker. In order to perform this attack, the attacker will collect some data of the traces of the network as a first step. Then he/she extract some features and train the classifier that is chosen by the attacker. After that the attacker will use this trained classifier to make a prediction. Some of the attacks have been putting forward and they are all able to achieve the accuracy of more than 90% which is a huge threat for the privacy of the users.

To overcome all of these attacks, there are some defense mechanisms have been proposed. The first one is WTF-PAD. The key idea of this defense mechanism is to insert some dummy packets while detecting unusual time gaps in the network traffic. This defense is deployed on Tor. However, it has been shown that it can be broken by deep fingerprinting with approximately 90% accuracy. The second defense mechanism that is proposed is Tamaraw. This defense forces the packets from both directions in a constant duration. However, that defense will incur a lot of additional burden so it is relatively more expensive.

To evaluate these defenses, they have examined them in terms of privacy and overhead. In order to evaluate, they have referred it to data overhead which is the additional data that is sent and time overhead which is the extra delay that is caused by the defense. The browsing experience is affected both on the time and also data overhead.

In order to find a better defense than all these defenses, they have proposed the two Zero-delay lightweight defenses that are called FRONT and GLUE. In this work, By saying zero delay, they have tried to explain that the defense has zero time overhead and the lightweight means they have required a small data overhead. The key idea of FRONT is similar as WTF-PAD. It surprises the network traffic by inserting additional meaningless packets in different directions. To decide when they have needed to send a dummy packet, instructions of some research are followed. These researches has been showing that the trace front leaks more information. So they have followed some intuitions of it. The first one is trace front obfuscating. In order to achieve it, they have sampled n timestamps where n in a random variable. Second intuition is to have trace to trace randomness. In that intuition, the same page is loaded each time and quite different traces can be collected. To evaluate FRONT, they have collected a dataset that includes approximately one hundred web pages. Each of them has been visited 100 times. They have also visited 10.000 webpages. The attacker will split 90% of this data set for training and 20% for testing. The aim of the attacker is identifying whether the client is visiting a monitor page and also to determine which monitor pages the user has visited. As a result of the experiment, FRONT has the same overhead as WTF-PAD, that is %33. That means, FRONT has been shown to outperform WTF-PAD. By comparing these with Tamaraw, it is observed that it incurs five times more overhead than FRONT but it is found also that kNN classifier has an even lower score against the FRONT. For the other attacks, FRONT can lower their f1 score to less than 0.46.

Another defense is proposed in this work that is called GLUE. The key idea of it is to force the attacker to solve some difficult problems about splitting. Some of the researches has showed that it is very hard to divide two web page loadings when they have overlapped each other. So, they have sent some dummy traces between the loading gaps and make them look like one single trace. In order to classify those pages, the attacker has to find the method of splitting. However, they have realized that the first trace is very easily identified. That means, the attacker can know the start of the loading. Therefore, they have covered the trace with some additional FRONT noise data. To sum up, when the loading has been starting, GLUE will add an additional noise. In reality, they can randomly choose a trace that is loaded before as a GLUE trace.

In order to make an evolution of GLUE, they have observed two cases. They have firstly assumed that the attacker always knows how many web pages are in the trace. So attacker should only find some fixed number of splits. They have first tested the attack on undefended traces without adding any extra noise. It is observed that the attacker can achieve approximately 92% split accuracy. It is relatively high accuracy for prediction although there is any noise. Then they have tested the attack by using the defense mechanism of GLUE. The accuracy record is significantly dropped to 20%. The precision is also decreased. If they have used more glued traces, it would be more difficult to perform the attack. As a second assumption, they took a more realistic assumption where the attacker does not know the page number in the trace. The attacker should determine the number of l first and find the corresponding splits. This case is harder than the previous one. So, even in the undefended traces, the attacker performs 45% to 75% accuracy for recall, and 41% to 77% accuracy for precision. By using the GLUE dataset, it was seen that the result of the attack performed even worse this time. It only achieves 3% to 46% recall and 1% to 16%

accuracy for precision. The time overhead of GLUE is always zero because it does not delay any of the packets of the users. However, its data overhead is dependent on the user behavior.

In conclusion, in this work, they have proposed 2 defense mechanism against the website fingerprinting attack in Tor. FRONT focuses on confusing the front of the traces and ensures randomness of trace to trace. On the other hand, The GLUE forces the attacker to solve some split problem that is hard to solve. Most of them are lightweight.