International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 6 Volume No.6 October - November – 2022
DOI:10.46647/ijetms.2022.v06i06.071 ISSN: 2581-4621

RBP: a website fingerprinting obfuscation method against intelligent fingerprintingattacks

Thipparthi Upendra Chari[1], B Srinivasa S P Kumar[2]
[1]*Department of MCA, Chaitanya Bharati Institute of Technology, India.*
[2]*Ascend Author, Assistant Professor, Department of MCA, Chaitanya Bharati Institute of Technology, India.*

Abstract
A significant danger to website privacy and web security is posed by website fingerprinting (WF), a passive traffic analysis attack. It gathers the network packets created when a user accesses a website and then employs a number of approaches to identify patterns in the network packets that can be used to determine the kind of website the user is accessing. Numerous anonymous networks, like Tor, can satisfy the need to conceal identify from users while participating in network activities, but they are also vulnerable to WF assaults. In this research, we present Random Bidirectional Padding, a website fingerprinting obfuscation technique against sophisticated fingerprinting techniques (RBP). It is a cutting-edge website fingerprinting defence technology built on time sampling and random bidirectional packet padding that can change real packet distribution to destroy Inter-Arrival Time (IAT) features in the traffic sequence and increase the difference between datasets with random bidirectional virtual packet padding. We test the efficiency of thedefence against cutting-edge website fingerprinting attacks in actual circumstances.

Keywords: Packet padding, Website fingerprinting defence, Traffic analysis, and Traffic obfuscation

1. INTRODUCTION

As the Mobile Internet has grown, a sizable number of mobile smart devices have joined the network, placing unprecedented strain on the backbone network. The data produced by edge devicescannot be processed effectively by cloud computing's centralised processing approach. Researchers proposed edge computing, a newcomputing model that performs computation at the network's edge, as a solution to this issue [1]. Edge computing may handle a significant volume of temporary data close to the network's edge and minimise communication with cloud computing infrastructure, which drastically reduces bandwidth usage and system latency.

However, users who are situated in the edge computing architectureare susceptible to traffic analysis attacks. As shown in Fig. 1, an attacker can discover users' identities by carrying out a traffic analysis attack in the manner shown in the following four steps:

Step 1: To run a train set, the attacker gathers traffic data from thecore cloud.
Step 2: Using the gathered training traces and accompanying labels,he trains the machine learning-based classifier.
Step 3: To collect unlabeled traces, the attacker eavesdrops onmobile users.
Step 4: The attacker utilises the taught classifier to predict theidentities of users during the prediction phase.

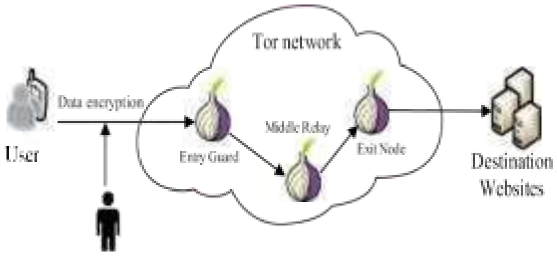
Due to the serious privacy risk posed by traffic analysis attacks, more and more users want to remain anonymous when participatingin network activities. As a well-known anonymity, Tor [2].

1.1 MOTIVATION

An unprecedented amount of mobile smart devices have joined to the network as a result of the rise in popularity of the mobile internet, placing a heavy burden on the backbone network. The centralised processing approach of cloud computing will be unable to effectively handle the size of the data produced by edge devices. [11] Researchers proposed edge computing, a new computing model that executes computation at the network's edge, asa remedy for this issue. Edge computing lessens interactions with cloud computing facilities and bandwidth overhead by processing huge amounts of transient data at the network's edge. On the other hand, users in the edge computing architecture are vulnerable totraffic analysis attacks.

1.2 TERMINOLOGIES

BuFLO - Buffered Fixed Length ObfuscationCNN - Convolutional Neural Network
IAT - Inter-Arrival Time
KNN - K-Nearest Neighbor
Mbps - Megabits per second
RBP - Random Bidirectional PaddingTOR - The Onion Router
UML - Unified Modeling LanguageWF - Website Fingerprinting
WTFPAD - Website Traffic FingerprintingAdaptive Defense



2. LITERATURE SURVEY

Since finger-printing can reveal a tonne of information about the user, including the network protocols [9], operating systems, hardware, and software, among other things, it has been a serious issue for everyone in recent years. To learn as much as they can about their targets, hackers start their attacks using this technique. Website fingerprinting, browser fingerprinting [7], and device fingerprinting are only a few examples of the various forms of fingerprinting attacks. In this study, we concentrate more on a type of passive traffic analysis attack called [8] Website Fingerprinting. numerous types of websites Machine learning and deep learning models can be used in fingerprinting attacks, and evaluating them isa difficult undertaking because we are unsure of the exact methods the adversary will employ to identify the website. Automated website fingerprinting [10], triplet fingerprinting [13], and deepfingerprinting using CNN [12] are only a few examples of website fingerprinting attacks that make use of machine learning and deep learning-based methods. The information and the users' privacy are at risk because, when tested in the closed world scenario, they only provide an accuracy of 90–98% [14] on the website the user has accessed.

Anonymous networks like Tor [4] can let individuals browsing a website remain anonymous. Users get protection from network eavesdroppers thanks to Tor. However, numerous recent research have revealed that Tor is similarly at risk from website fingerprinting attacks [1]. To safeguard privacy and uphold security, numerous website fingerprinting protection methods were established. By utilising the properties and patterns of the network packets created, the machine learning system guesses the website. The more distinctive the elements, the more precisely one can identify the website. It will be significantly more difficult for the trained models to recognise the website if we mask the traffic patterns [6] and factors that mislead them. As a result, BuFLO, a padding mechanism invented by Dyer et al. [3], injects false packetsbetween actual packets, confusing the algorithm that has been trained to recognise the website. Similar to this, a variety of website fingerprinting mechanisms were invented, and the field of study in this area has expanded quickly. In addition to BuFLO, there are numerous other protection mechanisms, such as WTFPAD, TAMAWRAW, and BiMorphing [2] [15], but even these are vulnerable to website fingerprinting attacks with a deep learning model that has a closed-world accuracy of nearly 98%. Using the same technique of dummy packet padding and overcoming the shortcomings of the previous website fingerprinting defence mechanisms—to destroy the Inter-arrival Time (IAT) Feature, which is the main feature used in the recent website fingerprinting attacks—represents a significant novel contribution to our work.

3. Random Bidirectional Padding

In our defense, the *padding* phase apply randomization to thepacket padding without any delay and moderate bandwidth overhead. The algorithm of padding as follows.

Algorithm 1: Random Bidirectional Padding **Input**: Original traffic *Toriginal*, target traffic *Ttarget*

Output: The traffic sequence padded by RBP

```
// Define the size of padding packet ;
length ← Definition()
while trace(Toriginal) do
    t ← Sampleing(Ttarget ) ;
    if t expires then
        flow ← flowdirection(Toriginal) ;
        // Generate dummy packet ;
        dummy ← generatedummy(flow, t, length) ;
        // Inject the dummy packet ;
        insert(dummy) ;
    end
end
```

The length of all dummy packets padded by RBP is specified by the algorithm at the start of the padding phase to be MTU (Maximum Transmission Unit). The position of the traffic trace for padding fake packets is then determined using the time t collected during the startup phase. When the time t expires and the client has not received the real packet, the RBP algorithm will inject a dummy packet to trace. The timestamp, direction, and length of the genuine packet serve as the parameters for the created dummy packets. For instance,the timestamp of the dummy packet would be equal to the timestamp of the present traffic plus the time t. The two preceding packet directions determine the direction of the dummy packet. If the directions of the first two packets are the same, the dummy packet will be in the opposite direction; if they are the opposite, it will be in the same direction as the first packet.

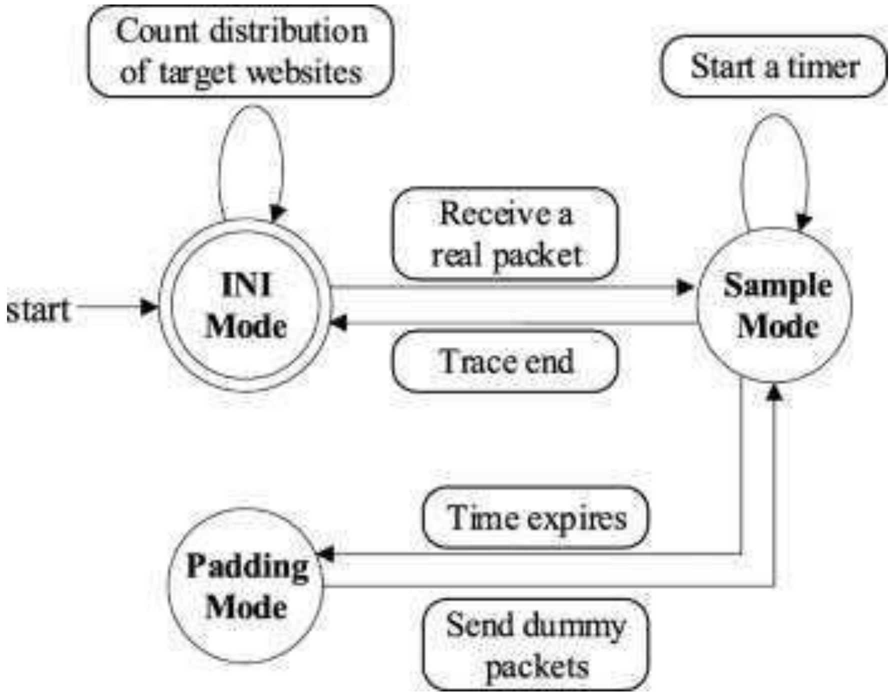
Experimental evaluation and discussion

We demonstrate the potency of the suggested WF defencein this section. We test our defence against a dataset from Tor. When no defence is used and when a defence mechanism is present, we look at closed-world and open- world scenarios.

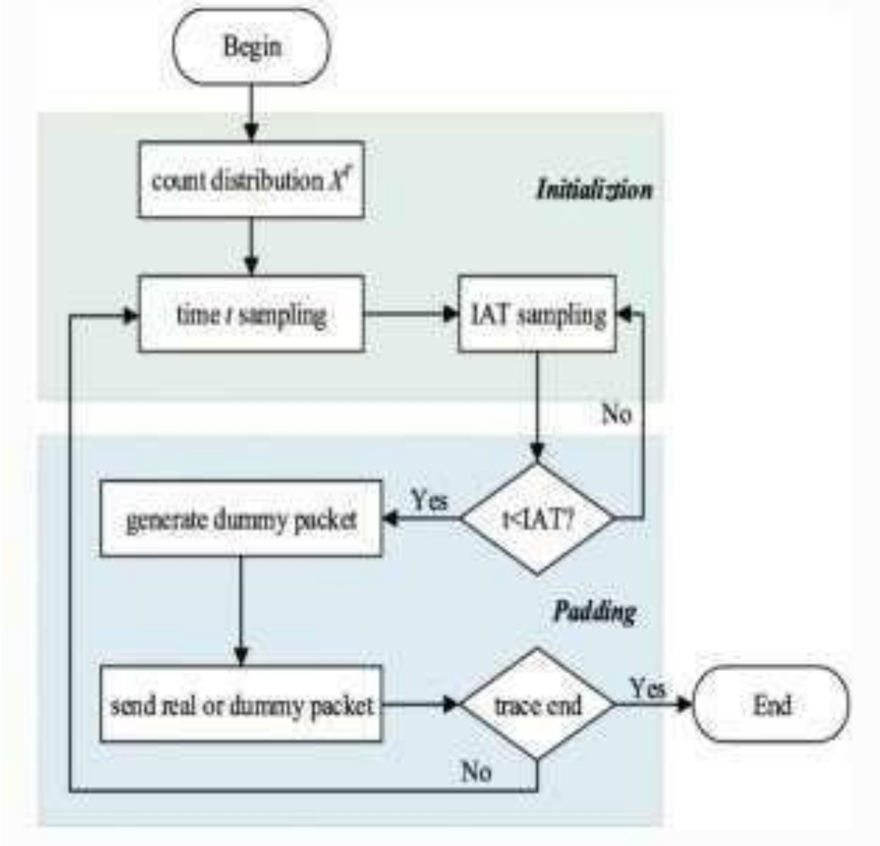
Dataset

The dataset we employ includes the open-world and closed-world traffic data mentioned in 2.1. We assessthe efficiency and overhead of our defence in a real- world setting using the Tor dataset. The information is made up of encrypted packets that the browser creates when it connects to the Tor network. Additionally, we employ the dataset in a closed-world scenario as a comparison in an experiment to further assess how well our protection functions in other datasets. There are two groups of collections in the dataset. The first one is a collection of the websites being watched; there are 100 websites in total, each with 90 traces. For the tests in a closed world, we use this set of data. The second group includes 9000 websites, each of which has a trace. Thesewebsites were chosen among the top websites according to Amazon Alexa. We refer to the first group of 100 websites in the open-world setting as the monitored set and the second group of 9000 websites as the unmonitored set. The top 100 websites according to Amazon Alexa, which are made up of 100 websites with40 traces apiece, are also used to pick the dataset for the literature.

System Architecture:



Flow Chart:

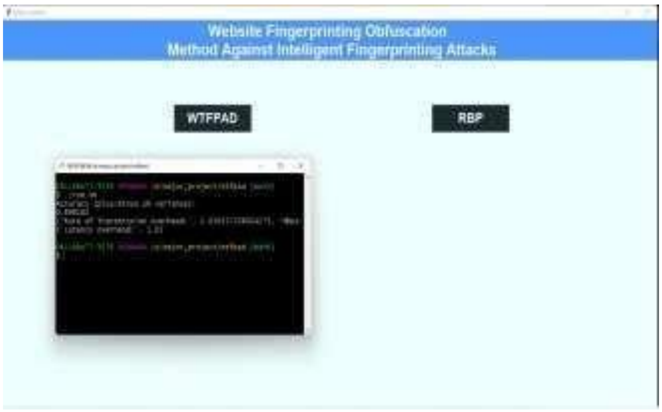


Results

Using the Tor dataset, we evaluate the RBP approach in the closed-world and open-world settings. We show the results when no morphing is applied (normal traffic) and compare them to the morphed data (when packets are morphed).

RBP in closed-world.OUTPUT

Website Fingerprinting Obfuscation
Method Against Intelligent Fingerprinting Attacks
WTFPAD
RBP


Website Fingerprinting Obfuscation
Method Against Intelligent Fingerprinting Attacks
WTFPAD
RBP

4. CONCLUSION

In this paper, we presented a novel WF defence that can thwart WF attacks based on deep learning models. This defence combines direct time sampling and random bidirectional padding, which ensures a modest bandwidth overhead and results in zero delay for real packets that are sent between the client and the server. Our defence can extend the difference in traffic distribution caused by a user accessing the same websiteat different times while hiding the inter-arrival time feature inthe traffic sequence. As a result, an adversary using a deeplearning model that was trained on an earlier dataset is unable to accurately identify traffic, increasing the cost of the adversary's assault. By studying the defence against passive attacks andcontrasting it with cutting-edge techniques, we empirically demonstrated the usefulness of the suggested methodology. The positive outcomes, reasonable bandwidth usage, and genuinepackets with zero delay provide up new possibilities for a more useful WF protection.

5. REFERENCES

1. Shi W, Sun H, Cao J, Zhang Q, Liu W (2017) Edge computing-an emerging computing model for the internet of everything era. J Comput Res Dev 54(5):907–924.

2. Dingledine R, Mathewson N, Syverson P (2004) Tor: The second-generation onion router In: 13th Usenix Security Symposium. Usenix.

3. Juarez M, Imani M, Perry M, Diaz C, Wright M (2016) Toward an efficient website fingerprinting defense In: European Symposium on Research in Computer Security, 27–46.. Springer, Switzerland.

4. Sirinam P, Imani M, Juarez M, Wright M (2018) Deep fingerprinting: Undermining website

fingerprinting defenses with deep learning In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 1928–1943.

5. Sirinam P, Mathews N, Rahman MS, Wright M (2019) Triplet fingerprinting: More practical and portablewebsite fingerprinting with n-shot learning In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 1131–1148.

6. Rescorla E, Modadugu N, et al. (2006) Datagram transport layer security. RFC 4347, April.

7. Rescorla E, Schiffman A (1999) The secure hypertext transfer protocol. IETF Request for Comments, RFC 2660.

8. Wang T, Cai X, Nithyanand R, Johnson R, GoldbergI (2014) Effective attacks and provable defenses for website fingerprinting In: 23rd {*USENIX*} Security Symposium ({*USENIX*} Security 14), 143–157.

9. Panchenko A, Lanze F, Pennekamp J, Engel T, Zinnen A, Henze M, Wehrle K (2016) Website fingerprinting at internet scale In: NDSS, 1–15.

10. Rimmer V, Preuveneers D, Juarez M, Van Goethem T, Joosen W (2018) Automated website fingerprinting through deep learning In: Proceedings of the 25nd Network and Distributed System SecuritySymposium.