Journal of Intelligent Information Systems (2020) 54:341–362
https://doi.org/10.1007/s10844-019-00553-0

Effective defense against fingerprinting attack based on autocorrelation property minimization approach

Hojjat Jahani[1] · **Saeed Jalili**[1]

Received: 4 June 2018 / Revised: 28 February 2019 / Accepted: 1 March 2019 /
Published online: 13 March 2019

Abstract
The website fingerprinting attack is one of the most important traffic analysis attacks that is able to identify a visited website in an anonymizing network such as Tor. It is shown that the existing defense methods against website fingerprinting attacks are inappropriate. In addition, they use large bandwidth and time overhead. In this study, we show that the autocorrelation property is the most important success factor of the website fingerprinting attack. We offer a new effective defense model to resolve this security vulnerability of the Tor anonymity network. The proposed defense model prevents information leakage from the passing traffic. In this regard, a novel mechanism is developed to make the traffic analysis a hard task. This mechanism is based on decreasing the entropy of instances by minimizing the autocorrelation property of them. By applying the proposed defense model, the accuracy of the most effective website fingerprinting attack reduces from 98% to the lowest success rate of the website fingerprinting attack, while the maximum bandwidth overhead of the network traffic remains on about 8%. Recall that the current best defense mechanisms reduce the accuracy of the attack to 23% with a minimum bandwidth overhead of more than 44%. Hence, the proposed defense model significantly reduces the accuracy of the website fingerprinting attack, while the bandwidth overhead increases very slightly (i.e., up to 8%).

Keywords Anonymity network · Tor · Fingerprinting attack · Defense model · Autocorrelation property

1 Introduction

Today, it is necessary to construct secure networks for data transmission with high privacy and availability. The main goal of introducing anonymity networks is providing secure networks with high privacy and availability features to prevent information leakage and reduce

✉ Saeed Jalili
sjalili@modares.ac.ir

Hojjat Jahani
h.jahani@modares.ac.ir

[1] Computer Engineering Department, Electrical and Computer Engineering Faculty, Tarbiat Modares University, Tehran, Iran

the possibility of traffic analysis. Tor network is the most popular anonymity network based on the onion routing such that a large number of users employ it to benefit from the advantages of a secure network (Dingledine et al. 2004, 2008). The website fingerprinting (WF) attack is one of the most effective attacks based on traffic monitoring and analysis to detect a visited website which violates the user privacy of an anonymity network such as Tor (Hintz 2002; Sun et al. 2002; Zhu et al. 2005; Murdoch and Zieliski 2007). In this type of attack, an analyst (the attacker) is able to categorize websites merely by extracting sequence of packets within the passing traffic of websites and apply machine learning techniques in parallel. With the development of approaches for the extraction of the similarity distances between instances in order to increase the accuracy of detecting the visited websites, some defense methods were presented to improve the security and privacy of the Tor anonymity network. However, current defense methods do not have an effective impact in preventing the WF-attack as they impose a high bandwidth and time overhead.

In this study, we show that the autocorrelation property is the most important success factor for the WF-attack. We also show that none of previous defense methods can control the impact of this factor effectively. Thus, anonymity networks are still prone to the WF-attack. We propose a novel model based on a devised traffic distribution filter to control the autocorrelation property of instances by increasing the artificial similarity between them. The proposed defense model is a general defense method that is robust against any classification attacks since it boosts the correlation and similarity between different instances. Finally, the proposed defense model is shown to be decentralized and user-friendly.

The paper is organized as follows: Section 2 presents the background of the Tor anonymity network. Section 3 discusses the literature of WF-attacks and the countermeasure methods. The Proposed defense model and its implementation are thoroughly explained in Section 4. In Section 5, the performance of the proposed defense model is examined and the results are compared with previous defense models. The prominent properties of the proposed defense model are analyzed and discussed in Section 6. Finally, a conclusion is given in Section 7.

2 Background

2.1 Tor anonymity network

Tor is one of the pioneering anonymizing networks with millions of users a day. Tor's design is based on onion routing consisted of three relays (i.e., Entry guard, Middle and Exit relay). To set up an anonymous communication between a user and a webserver, Tor establishes three distinct secure connections with aforementioned relays. Each relay knows the identity of the previous and the next relay relative to itself. Thus each relay recognizes either the user's identity or the web server's identity but not the both at the same time. As a result, the user privacy is achieved in the anonymizing network. Figure 1, shows the use of Tor anonymity network by a user.

2.2 Threat model

As shown in Fig. 1, the passive attacker can only monitor the traffic flow passing between a user and the first relay (i.e. Entry guard) of the anonymizing network. By analyzing the information collected through eavesdropping of the traffic, the attacker tries to violate the

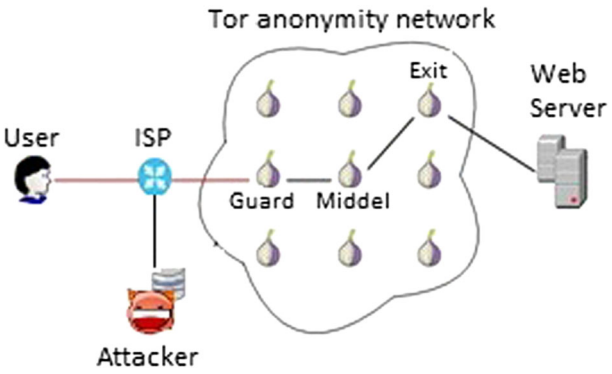

Fig. 1 User connection to a webserver through Tor anonymity network and the location of a local passive attacker between a user and the first relay

user's privacy and identify the visited website by the user. There are two distinct scenarios for the WF-attack: 1) The open-world scenario and 2) The closed-world scenario. In the open-world scenario, the attacker knows only a few monitored websites and learns the pattern of their instances. The target of this scenario is to detect the monitored websites through the traffic of thousands of non-monitored websites which are linked by the user. In the closed-world scenario, it is assumed that the attacker knows all websites and has collected all instances of them. Thus, in the learning stage, the patterns of all instances are learned. Finally, in the test stage the intruder identifies the user's visited websites.

3 Related works

The WF-attack is a serious threat to security of anonymity networks such as Tor. In this section, we first review some of the most important WF-attacks and then introduce effective defenses against them.

3.1 Attacks

In recent years, many researchers have tried to develop the WF-attack by using more suitable classification techniques and providing more advanced feature extraction methods (Herrmann et al. 2009; Shi and Matsuura 2009). The most successful attack was introduced by Panchinko in 2011 based on statistical properties of instances (Panchenko et al. 2011). Later, Cai et al. (2012) and Wang and Goldberg (2013) applied the similarity distance approach to extract optimal features. Then Wang et al introduced a new attack based on the K-NN[1] in the open-world scenario. This attack is able to detect 100 monitored websites through thousands of non-monitored websites at an 85% TP with FP of 0.6% (Wang et al. 2014). In 2016, Panchenko et al., gathered the large-scale datasets of websites over the Internet connection that included realistic background noise for the first time. They were able to improve the accuracy of the WF-attack in both scenarios. They also used the K-NN algorithm in the Closed-world scenario to enhance the accuracy rate to 89% for 100 websites (Panchenko et al. 2016).

[1] k-Nearest Neighbors

For most of the aforementioned WF-attacks, the packet rate or website loading time are seen as features that improve the accuracy of WF-attacks. However these features shouldn't be regarded as security vulnerabilities of the Tor anonymity network, but something that depends on user's equipment and network characteristics (e.g. speed, bandwidthc). In 2016, a novel algorithm based on the FFT[2] was introduced to calculate the similarity distances (Jahani and Jalili 2016). In this approach, they discovered the existence of autocorrelation property in instances as the most important security vulnerability of the Tor anonymity network. As a result, privacy breaches in anonymity network is possible without utilizing loading time. A broad range of approaches of WF-attack are also proposed in Juarez et al. (2014), Wang and Goldberg (2014), and Hayes and Danezis (2016).

He et al used active WF-attack against Tor anonymity network and showed that it is vulnerable to this type of attack (He et al. 2014). They achieved a 65% detection accuracy rate for 100 websites despite adding a one second delay. In 2015, Gu et al introduced the active WF-attack based on the Mahalanobis distance. This attack not only shows a serious vulnerability of the Tor anonymity network, but also it threatens the user's privacy. They could achieve the detection accuracy rate of 76% and 40.5% for the first and second websites, respectively (Gu et al. 2015).

3.2 Defenses

Initial defense approaches against WF-attack was introduced based on packet padding. Wright et al introduced two defense techniques, DTS[3] and TM,[4] based on traffic distribution, in which they tried to show that the transmitting traffic of one website becomes similar to another Specific website (i.e., target website) by changing packet lengths (Wright et al. 2009). In 2012, Dyer et al. showed that all defense methods based on traffic distribution are not efficient against new WF-attack (Dyer et al. 2012). Luo et al. proposed HTTPOS method (i.e., HTTP obfuscation) in the application layer as a platform to deal with WF-attack (Luo et al. 2011). Despite many other defense approaches, HTTPOS is implemented in the user side and does not require any support in the server side. This method operates with the intelligent use of HTTP pipelining which changes the length and the number of outgoing packets. This defense approach is not useful against the WF-attack that is based on the similarity distance (Cai et al. 2012; Wang and Goldberg 2013; Jahani and Jalili 2016). Tor developers have proposed a new defense approach with no overhead to Panchinko's method (Perry 2011). This approach is implemented in the application layer which tries to obfuscate traffic by using the pipelining technique and randomizing order of the requests. Similar to the HTTPOS method, this approach has no effect on the WF-attack that is based on the similarity distance (Cai et al. 2012; Wang and Goldberg 2013; Jahani and Jalili 2016).

In 2011, the BUFLO[5] method was introduced by Dyer, He made a comprehensive evaluation and analysis of all defense methods and showed their failure against WF-attack (Dyer et al. 2012). The BUFLO method achieved a relative success against WF-attacks. BUFLO is an obfuscation method that changes the length of packets by padding and fragmenting and sending new packets in each specific time interval. If there are no data to forward, this method sends dummy packets instead of actual ones. They reduced the accuracy rate of only

[2]Fast Fourier Transformation
[3]Direct Target Sampling
[4]Traffic Morphing
[5]Buffered Fixed-Length Obfuscator

those WF-attacks that are based on statistical properties (not WF-attacks that are based on similarity distance) to 18% with a 193% overhead for 128 websites. Then, Juarez et al used BUFLO against WF-attack based on the similarity distance, such as Pa-SVM and DL-SVM and could reduce the accuracy rate to 14% and 18%, respectively (Juarez et al. 2016).

In 2014, Cai et al. introduced Tamaraw model to prove their claim about the possibility of having a defense with less overhead (Cai et al. 2014). They developed this method to overcome BUFLO shortcomings such as equivalence of transmitting rate (ρ) in both communication directions. This method reduces the accuracy rate of WF-attacks based on similarity distance to 41%, while the bandwidth and time overhead of this method are 140% and 50%, respectively. Later, they proposed CS-BUFLO[6] which determines data transmission rate by considering network traffic congestion. By evaluating CS-BUFLO for 200 websites, they could reduce the accuracy rate of WF-attacks based on the statistical-properties to 18% with the bandwidth and time overhead of 279% and 327% respectively (Cai et al. 2014). Moreover, for 128 websites, they reduced the accuracy rate of WF-attacks based on the similarity distance to 40.5% with the bandwidth and time overhead of 230% and 273% respectively. The review of the above works reveals that the efficient defense methods against WF-attacks severely reduce the performance of the anonymity networks due to high bandwidth and time overhead (i.e., the delay time to load a web page is about half a minute). In 2015, Wang et al. proposed Walkie-Talkie model; they changed the connection link of the browser from Full-duplex to Half-duplex to limit the attacker's ability. Moreover, they reduced the attacker's ability to correctly identify a page of the target website by inserting dummy packets to the link. They could reduce the accuracy rate of WF-attacks based on the similarity distance to 48% in 100 websites, while the bandwidth and time overhead are 23% and 15% respectively (Wang and Goldberg 2015).

In 2016, Juarez et al, offered a novel defense based on AP[7] against WF-attack in Tor. In a closed-world scenario, they decreased the accuracy of WF-attack from 91% to 23%. although unlike previous defense models, the latency overhead is about zero and bandwidth overhead have always been less than 60% (Juarez et al. 2016)]. Cherubin et al, proposed a defense method at application layer which is implemented on both sides of the communication (i.e. user side and the server side). The user-side defense is a lightweight one which is employed as a browser add-on (Cherubin et al. 2017).

4 The proposed defense model and its principles

In this section, our proposed defense model is presented and explained in detail. This model aims to resolve security vulnerabilities of anonymizing networks such as Tor and also addresses the challenges and limitations of previous defense models including: 1) High bandwidth and time overhead, 2) being prone to the DOS attack (i.e. fixed length packet, and the possibility of dropping packets in the first relay), 3) the packet rate needs to be controlled to reduce the accuracy of WF attacks, such intervention increases the bandwidth and website access delay and 4) the existence of the autocorrelation property in instances as the most important security vulnerability and main cause of WF-attack effectiveness. These limitations are among the most important factors that make previous models impractical.

[6]Congestion Sensitive BUFLO
[7]Adaptive Padding

In this study, the proposed defense model minimizes the autocorrelation property to overcome the aforementioned challenges and completely vanishes the required features for WF-attacks (i.e., fixed packet length, packet order, etc.). The proposed defense model randomizes the length of original packets with minimum dummy packets insertion (minimum bandwidth overhead), such that the required features to mount WF-attacks, are tangibly removed and the length of packets are changed randomly in loading each website. In the following, we first present the principles of the proposed defense model, and then describe the proposed defense model in detail.

4.1 Autocorrelation property

An important technique to find out the correlation between the components of two sequences is to determine the amount of autocorrelation in each sequence. Autocorrelation is the similarity between a sequence and its shifted vectors with different shift values. The amount of autocorrelation is obtained by calculating the Hamming distance between the original sequence and its shifted sequence. On the other hand, one of the main randomness criteria of a sequence is the autocorrelation property of that sequence. In order to explain the relationship between autocorrelation and the randomness of a sequence, it is necessary to point out the randomness postulates proposed by Golomb (1967). They suggested that the two main randomness postulates are as follow:

- The number of occurrence of elements in a pseudo-random sequence is approximately equal.
- The out-of-phase autocorrelation of a pseudo-random sequence is a negligible constant

To further explain the second randomness postulates, suppose that s_t is a sequence of period p (i.e., $s_{m+p} = s_m$ for every m). For any fixed τ defined as shifted value, it should satisfy $0 \leq \tau < p$ such that $s(t+\tau)$ is shifted sequence. Suppose that A is the number of positions in which the two sequences (i.e., original sequence and shifted sequence) are equal and D=(p -A) is the number of positions in which they are unequal. Then autocorrelation function $C(\tau)$ is defined as relation (1):

$$C(\tau) = \frac{A - D}{p} \tag{1}$$

When τ=0 we have in-phase-autocorrelation and in this case, D=0 and A=p so that C(0)=1 . For $\tau \neq 0$ we have out-of-phase autocorrelation such that its amount for pseudo-random sequence is a negligible constant.

Furthermore, assume that each instance t_i(i.e., sequence of packets) in relation (2) indicates a vector of exchanged packets in a secure anonymous channel for loading a specific website.

$$t_i = p_{i_1}, p_{i_2}, \cdots p_{i_l} \tag{2}$$

In relation (2), l shows the last packet of instance t_i , the equation $p_{i_k} = \pm s_{i_k}$ means that p_{i_k} is k^{th} packet of instance t_i, s_{i_k} is the length of p_{i_k} packet and $\pm$ indicate the transmission direction where "+" shows data is transmitted from server to user (i.e. Incoming packets) and "−" shows data is transmitted from user to server (i.e. Outgoing packets). Each instance consists of several subsequences, where each subsequence as a sequential of burst packets is equivalent to a sinusoidal signal with distinct frequency. In addition, the number of subsequence samples in an instance is equal to the magnitude of FFT coefficient (Jahani and Jalili 2016). On the other hand, the existence of autocorrelation property in each instance is very useful in calculating the similarity between two instances and distinguishing them from

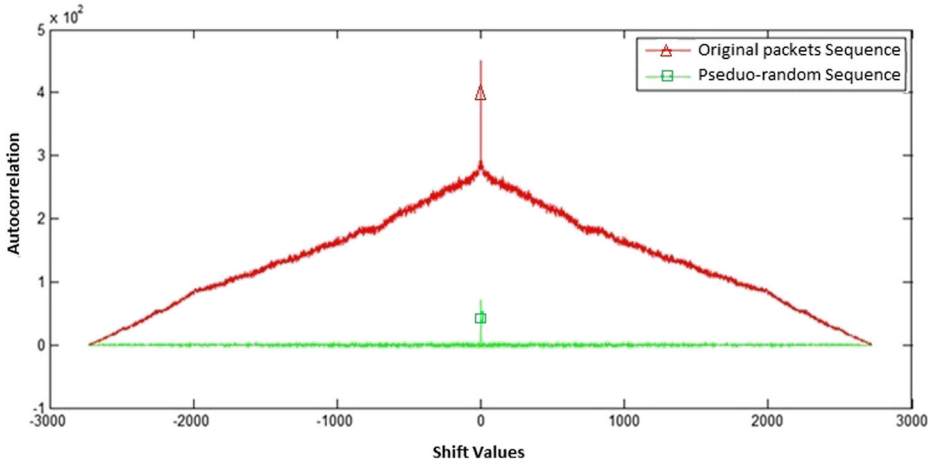

Fig. 2 The difference of autocorrelation property between a Tor instance (i.e., original packets sequence) and a pseudo-random sequence with same length

each other by relation (3), in which t_i^F indicates the FFT domain of instance t_i, l_i expresses the number of packets of instance t_i, L shows the length of output vector of FFT and finally α factor is used to normalize relation (3) such that sd values will be in the useful range.

$$sd(t_u^F, t_v^F) = \left(\frac{\frac{1}{L} \sum_{k=1}^{L} (t_{u_k}^F, t_{v_k}^F)^2}{\frac{(l_u + l_v)}{2}} \right)^{\alpha} \tag{3}$$

Figure 2 shows the existence of autocorrelation property for two different sequences, an original packets sequence(i.e., Tor instance) and the pseudo-random sequence. The corresponding plot of a real Tor instance (i.e., a sequence of 1384 packets between a Tor user and server to load a website) is shown on the top part of Fig. 2. This curve represents two facts. The first fact means: the correlation between packets is completely dependent on the number and length of subsequence in the whole instance. The Second fact means: during website loading, the correlation between the beginning packets and the final packets of the instance is decreased significantly (this is shown in both sides of the spectrum on top curve of Fig. 2).

The bottom curve of Fig. 2 represents the autocorrelation property of pseudo-random sequence. As shown in Fig. 2, this curve only has value in the in-pahse-autocorrelation (i.e., with the amount of shift Value=0) and the out-of-autocorrelationin of pseduo-random sequence (i.e., in the amount of shift Value $\neq$ 0) is neglibile, that this is in accordance with the second postulate proposed by Golomb. As a result, there is almost no correlation between the pseudo-random sequence and its shifted vectors. So, all of its subsequences and even every packet of this sequence are generated independently.

In WF-attack that is based on FFT (Jahani and Jalili 2016), we used the autocorrelation property in instance sequences and have shown that due to the existence of this property, detection of instances of different websites is possible. In this attack method, we generated the distance matrix[8] by calculating the similarities between instances of different websites,

[8]Distance Matrix represents a matrix that its rows correspond to instances of different websites and columns correspond to websites, while cell i,j indicate the similarity distance between instance i and website j.

i.e., classes. The produced distance matrix has an interesting property: all instances of each website follow a certain pattern that is easily distinguishable from the patterns of instances of other websites. Hence, by using the SVM learning technique we can learn the pattern of each website. Then we can distinguish those websites that we learned their corresponding patterns in the past by monitoring the Tor traffic.

For minimizing the autocorrelation property in an instance, each element of the sequence (i.e., packet of an instance) should be independent of its previous generated elements. In addition, the order of repetition of each element follows a uniform distribution. Note that, making full coverage of autocorrelation property in practical anonymity networks such as Tor is not possible, because we need mutual interaction between server and user and disproportion in the number of packets sent on both sides of the connection (i.e., upload and download directions). In the following, the proposed defense model is described, in which it is attempted to reduce distinguishability of the traffic of websites by minimizing and controlling the autocorrelation property in instances effectively.

4.2 Proposed defense model

We recall that: 1) the output of combining a given sequence with a pseudo-random sequence results in another pseudo-random sequence and 2) reducing the maximum leakage of information depends on minimizing and controlling the autocorrelation property. The proposed defense model is based on the aforementioned points as shown in Fig. 3. The model can operate on both sides of the communication independently. As Fig. 3 represents, the proposed defense model consists of: 1) Buffering, 2) Traffic distribution filter, and 3) Embedding Tor Cells in New packets steps.

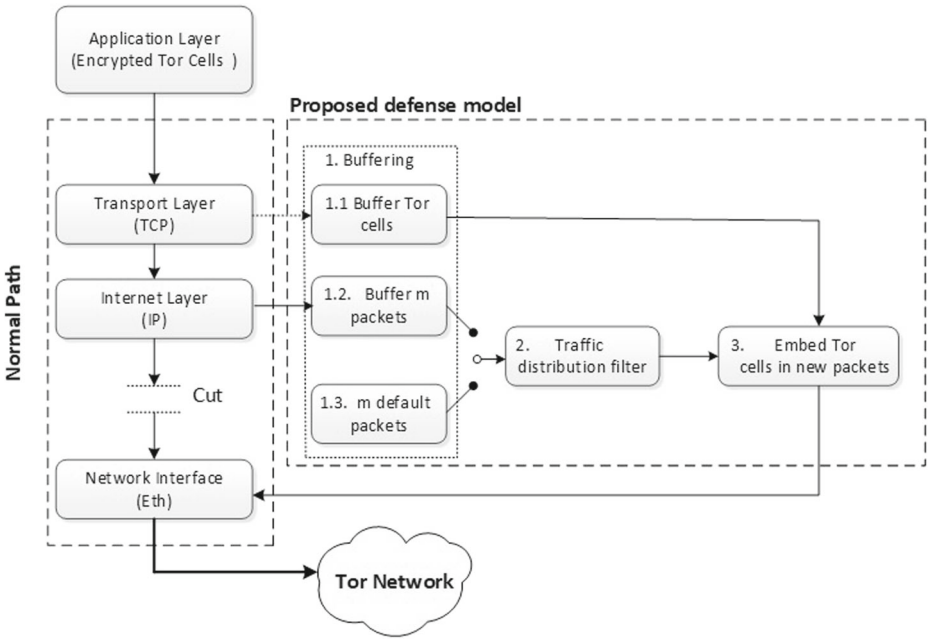

Fig. 3 The proposed defense model

We can apply the proposed defense model (Fig. 3) in the user (OP) and OR relays (Guard, Middle and Exit), but it is enough to run the defense model at a user and its corresponding Exit relay (i.e. Places where plain texts are presented).

4.2.1 Bufferingl

The Buffering step is responsible to save the Tor cells which include encrypted bytes of the original packet. As shown in Fig. 3, this step consists of three sub steps: "step 1.1. Buffer Tor Cells", "step 1.2. Buffer until m packets" and "step 1.3. m default packets". Step 1.1 buffers the Encrypted Tor cells in TCP Layer in which each cell includes 512 encrypted bytes. Then, in step 1.2, m sequential original packets of the IP layer are buffered by the second function (i.e., Buffer m packets). At the beginning of the connection to load the website or when the number of buffered packets is low, in order to avoid imposing an initial latency, the step 1.3 is introduced. this step includes m default packages which will only be used in two special cases (i.e. when the website starts loading or number of buffered packets is low).

4.2.2 Traffic distribution filter

According to Fig. 3 (i.e., the proposed defense model), most of the modifications are done at IP layer which is responsible for packet generation (based on two operations: Fragment, Defragment). In the IP layer, forwarding the generated packets to the network interface layer is prevented and they are buffered at step1 of the proposed defense model. Also the output of one of steps 1.2 or 1.3 (i.e., output of the Buffer m packets or output of the m default packets) enters into the traffic distribution filter step (i.e., step 2 of Fig. 3). This step is the main core of the proposed defense model that is designed to generate empty IP new packets, such that the generated new packets are random in terms of arrangement and the length of packets. In the following; a novel mechanism is introduced to achieve the randomness properties in the IP layer using a traffic distribution filter function.

In designing the traffic distribution filter function, the main goal is to minimize the auto-correlation property as much as possible. Moreover, setting parameters of traffic distribution filter function should obey the following two rules: 1) creating a sequence of skeleton packets which has random properties in terms of arrangement and length of packets, 2) the placement of the encrypted Tor cells in new packets should be feasible, so that the volume of new packets should be at least a few percent more than the current IP packets such that adding dummy bytes at the end of each packet becomes possible.

Unlike current mechanisms of IP layer in which the packet lengths are changed for MTU[9] or modified to a fixed value; the approach of this model is based on randomized packets length. In Step 2, placing the encrypted bytes (i.e., bytes of Encrypted Tor cells that are buffered at higher layers) in new packets are easily possible. This is done based on the two rules mentioned in the second paragraph at Section 4.2.2. Since in network interface layer and physical layer only constant headers of each layer are added to new packets, no violation occurs to change the order and length of packets. The autocorrelation property is not changed as well. The logic of the traffic distribution filter function is presented by Algorithm1 that shown in Fig. 4.

In Algorithm1, a sequence of new packets is created based on the sequence of original packets (i.e. packets related to instance loading of a given website) such that the

[9]Maximum transmission unit

Algorithm_1: sequence randomizing

Suppose that: 1) m packets of IP layer are buffered in Part_Sequence variable; and 2) n is the length of a random sequence.

GenPoly is a feedback primitive polynomial for example $P(z) = z^8 + z^6 + z^5 + z^4 + 1$

Initialstates is mandatory and for example is [1, 1, 0, 1, 0, 0, 0, 1] and NumBitsOut=n

$packet_{len}$ *represented a length of packet*

Commsrc.pn () and Generate (): functions that create the random sequences.

randperm (S): returns a random permutation of sequence S.

1. *Generate a random sequence with length n*
   H = commsrc.pn('GenPoly', 'Initialstates', 'NumBitsOut')
   $Rand\_V_{FFT}$ = 2* Generate(H) -1; % by using Generate(H), components of $Rand\_V_{FFT}$ are $\{\pm 1\}$
   $Rand\_V_{time} = real\left(iFFT(Rand\_V_{FFT})\right)$;
   Initialstates =H.CurrentStates;
2. *Convolution of* Part_Sequence *and* $Rand\_V_{time}$
   $New\_Part\_Sequence = conv(Part\_Sequence, Rand\_V_{time})$;
   $New\_Part\_Sequence = randperm(New\_Part\_Sequence)$;
3. *Correct the length of packets (less than 256 and over than 1500 in New_Part_Sequence).*
   for $i = packet_{len} < 256$
   $New\_packet_{len} = packet_{len} + 256 + randi(1000,1,1)$;
   End
   while $(j = packet_{len} > 1500)$ % Split packet to two packets with the following length
   $First\_New\_packet_{len} = 256 + randi(1000,1,1)$;
   $Second\_New\_packet_{len} = packet_{len} - (256 + randi(1000,1,1))$;
   End
4. *Check Total Number of bytes of New_Part_Sequence*
   while (# Bytes of New_Part_Sequence < # (1.05 *Bytes of Part_Sequence))
   add new packet with length $256 + randi(1244,1,1)$ to New_Part_Sequence ;
   End

Fig. 4 Algorithm1 : Sequence randomizing

autocorrelation of the new instance is decreased to a minimum possible value. So by applying Algorithm1 and generate sequence of new packets, the similarity between instances of different websites significantly increases and the likelihood of distinguishing them obviously decreases. In the following, the bytes of encrypted Tor cells that are buffered in TCP layer are put in new packets which are sent to the network interface layer by applying Embedded Tor Cells in New packets function.

Algorithm 1 takes as inputs two parameters m and n. m indicates the length of a subsequence that enters into the traffic distribution filter step (i.e., step 2 of Fig. 3) and n represents the length of a random sequence (i.e., output of a random generator). To select best values for m and n, we should consider two points: 1) do not impose any delays in loading the website and 2) the new sequence obtained by combining orgial packets and random sequence should have proper random properties. Performed with numerous evaluations to reduce the maximum autocorrelation properties, the best values for m and n are 100 and 66, respectively. Also concerning the size of shift registers and determining the primitive polynomial as its feedback, we select shift register with eight states which has two main

advantages: 1) its initialization is quick and easy, 2) this number of states, could generate a non-periodic random sequence of length 255 (i.e.,$2^8 - 1$) which is larger than length of m, n.

As aforementioned notes, the aim of Algorithm1 is to generate empty IP packets in accordance with the two rules mentioned at the second paragraph of Section 4.2.2. This algorithm is designed in four distinct parts as follows:

First part: this part, the shift register (based on primitive polynomial as a feedback tap and initial state of a given loading) is used to generate a random sequence with values of$\{\pm 1\}$ with length n. this sequence is then mapped to a time space (i.e., *Rand_Vtime*) by using *iFFT* function and the current state of a shift register which is stored as an initial state for the next iteration.

Second part: by applying *Rand_Vtime* sequence (i.e., the output of the first part) in the second part, the distribution of packets of each instance is randomized in terms of their order, arrangment and the length as much as possible. This is achieved by combining *Rand_Vtime* sequence with the *Part_Sequence* (i.e. subsequence that enters into the step 2 of Fig. 3) in the convolution function. The randomness property is further improved by applying randperm[10] function on *New_Part_Sequence*.

Third part: Next, in this part, the length of packets are controlled by checking and comparing them with the minimum and maximum permissible values (i.e., more than 256 and less than 1500).

Forth part: Finally, the total number of bytes of the*New_Part_Sequence* is checked with the original one, such that it becomes possible to fill encrypted Tor cells correctly. In addition an extra space is added to the end of the packets for adding dummy bytes. Hence, the number of bytes of the *New_Part_Sequence* should be at least 5% more than bytes of the original *Part_Sequence*.

4.2.3 Embed Tor Cells in New packets

In the "Embed Tor Cells in New packets" (i.e., step3 of Fig. 3), the generated skeleton packets in subsequence *New_Part_Sequence* at the fourth part of Algorithm1 are ready to be filled by the encrypted Tor cells. The symmetric cipher algorithm used in Tor is a block cipher (with a block size of 16 bytes) which operates in CRT mode (or any kind of mode that is used in Tor). Thus, in each packet, all blocks with 16 bytes are replaced by encrypted Tor cells. In addition, the last bytes of each packet (i.e., last uncompleted block) can be of length 1 to 15 bytes filled with dummy bytes. This operation continues until all bytes of buffered Tor cells are filled. After filling all encrypted bytes of Tor cells in new packets and forwarding them into the Tor network channel.

The whole processes (i.e., steps 1, 2 and 3) are repeated for the next subsequence (i.e., next m packets of the original sequence). Note that if the length of the last subsequence (i.e., *Final_Part_Sequence*) is less than m, the length of parameter n can be set based on this value. In order to satisfy the feature of packet rate in the channel, we can define parameter ρ like previous methods and manage it more freely due to the buffered new packets that are ready to send. It is certainly useful for decreasing attack accuracy, but this paper does not use parameter ρ.

[10]This function permutes the components of input vector in a random manner.

5 Evaluation

In this section the proposed defense model is evaluated on four main aspects: 1) autocorrelation property minimization, 2) covering features in a sequence of new packets, 3) the effectiveness of the proposed defense model against WF-attack, and 4) bandwidth and time overhead.

5.1 The effectiveness of the proposed defense model on autocorrelation minimization

In this study, the existence of autocorrelation property in instances of loading different websites is introduced as the most important factor the success of WF-attacks and the main security vulnerability of the Tor anonymity network. Thus, the main goal of the proposed defense model is to control and minimize this property such that the required features for successful WF-attack are diminished. Figure 5 shows the amount of autocorrelation in a given instance before and after applying the proposed defense model and also, the gap between autocorrelation values of both sequences and pseudo-random sequence.

As shown in Fig. 5, the autocorrelation property in the sequence of new packets compared with the original sequence is decreased dramatically (i.e., almost 5 times). Thus, it shows that the new sequence almost satisfies the properties of a pseudo-random sequence. The main reason that the new sequence cannot completely achieve the properties of a pseudo-random sequence and the mismatch of two curves in Fig. 5 is related to the conditions of packet exchange in anonymity networks which is inevitable. An example of these conditions is the disproportionality of the number of packets on both sides of the communication (i.e. download and upload routes). To compensate this disproportionality, it is required to add some dummy packets in upload route (from the user side to a server) and increase the bandwidth overhead. However increasing the bandwidth overhead is in contradiction with the main objective of the proposed defense model, i.e. controlling bandwidth overhead. Also, the impossibility of generating packets with any length (e.g., packets less than 64 bytes) is another cause of the mismatching.

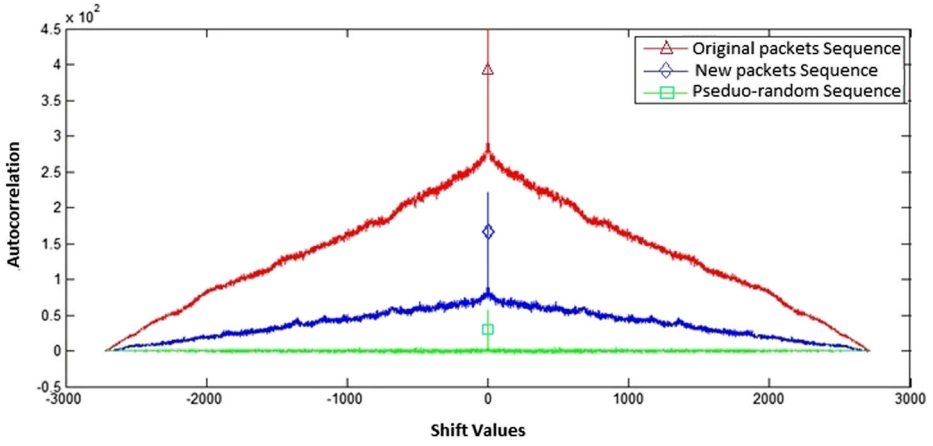

Fig. 5 The average autocorrelation of a website's packets sequence before and after applying the proposed defense model and their comparison with autocorrelation of a pseudo-random sequence

5.2 The impact of the proposed defense model on covering features

The accuracy rate of any WF-attacks completely depends on the amount of information leakage of each anonymity network and how to define features based on this information. The most important factors in information leakage are how to arrange and send packets, the number of each packet and time interval between packet transmissions. As mentioned in Section 3, in different WF-attack methods, various features are prepared for the optimal benefit of channel information leakage.

The most important idea of the proposed defense model is to reduce the information leakage of packet sequences by rearrangement and changing the length of packets and the number of each packet in a random fashion to minimize the usefulness of features used in WF-attack. Figures 6 and 7 show the effectiveness of the proposed defense model with the arrangement, length of packets and also the number of each packet in a given instance. Notice: In Fig. 6, positive and negative signs indicate incoming and outgoing of packets, respectively.

In Algorithm1, the sequence of new packets in an instance is obtained using convolution of the sequence of original packets with a random sequence. Thus, according to Fig. 6, the new sequence has random properties in terms of arrangement and the length of packets. In addition, this sequence is much more similar to a pseudo-random sequence compare with the original sequence. Also, due to the decrease of the average length of packets in the new sequence of packets (compared with the average packets length in the original one) the number of packets in the new sequence of packets is more than the original one.

The number of packets in both directions, download and upload (i.e., incoming and outgoing packets sides) is another feature that is very influential in the effectiveness of WF-attack. In Fig. 7, the number of each packet in the download direction (i.e., incoming packets side) of a given instance before and after applying the proposed defense model is shown. It should be noted that the upload direction behaves in a similar fashion as well.

As shown in Fig. 7a, the distribution of packets of an instance (the magnitude of their numbers in the original sequence) are completely non-uniform such that packets of length

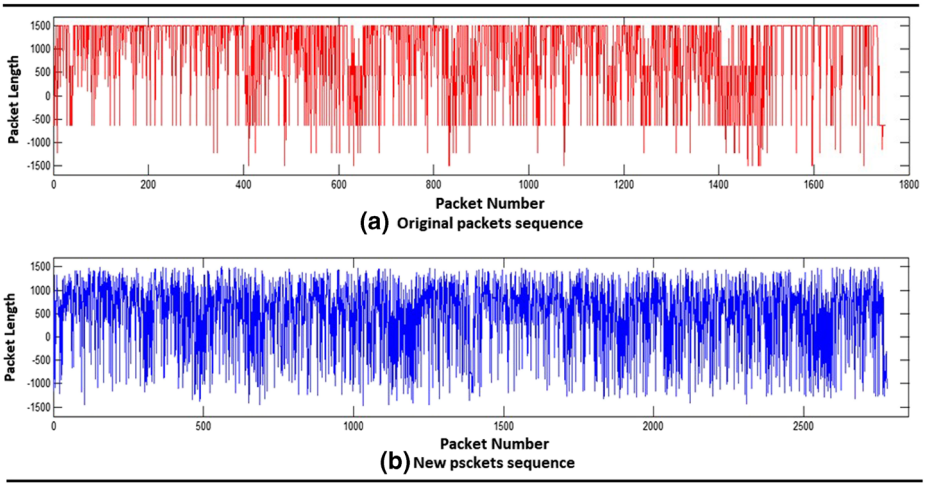

Fig. 6 The length and arrangement of packets of an instance (i.e., packets sequence of website) before and after applying the proposed defense model

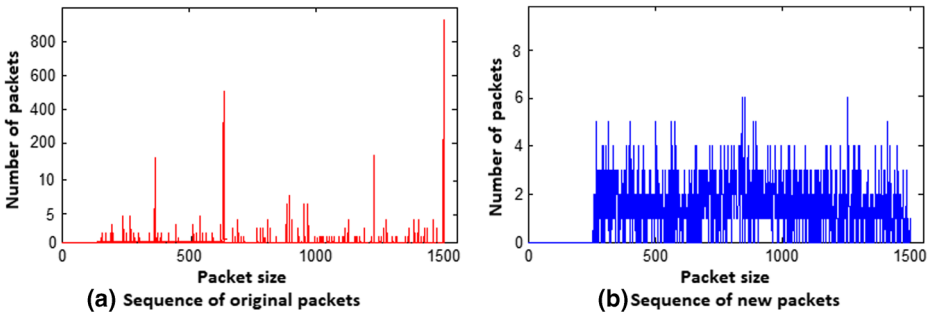

Fig. 7 The number of Packets of an instance before and after applying the proposed defense model. Note that the coordinates of the Y axis are not the samel

630 and 1500 are the most redundant ones which have frequencies of 542 and 917, respectively. On the other hand packets of different lengths rarely occur. According to Fig. 7b in the new sequence, the occurrence of packets with different length has a uniform distribution with their frequencies lying in the range of 1 to 6.

Summarizing the above discussions and recalling the principles provided in WF-attack based on FFT (Jahani and Jalili 2016), it can be concluded that:

– The existence of a distinctive pattern in the instances of different webfites (each website is equivalent with a class) is the most important factor for the success of WF-ettack. So, accordiig to relation (4), and the similarity distance matrix, the hamming distance between two instances of differenc websites (i.e.,$DH_{diff-class}$) is at least seven times more that the hamming distance between two instances of the same website (i.e.,$DH_{same-class}$) (Jahani and Jalili 2016).

$$\frac{DH_{diff-class}}{DH_{same-class}} \geq 7 \tag{4}$$

– By applying the proptsed defense model and establishtng a similarity distance matrix based on the new instances, relation (4) changes to relation (5). this means ihat the ratio decreases sharply to less than 1.86

$$\frac{DH_{diff-class}}{DH_{same-class}} \leq 1.86 \tag{5}$$

The advantages of the proposed defense model (as discussed in Sections 4.1 and 4.2) can be summarized as follows: 1) decreasing the autocorrelation property, 2) randomizing the arrangement and length of packets and 3) uniformization of the distribution of the number of packets. So it is concluded that the proposed defense model decreases information leakage in the sequence of packets. In other words, the similarity of instances of different websites increases.

5.3 The effectiveness of the proposed defense model against WF-attack

In this section, we examine the impact of the proposed defense model against WF-attacks. To evaluate the proposed defense model and measure its effectiveness practically, the data

sets of three papers (Cai et al. 2012; Wang and Goldberg 2013; Wang et al. 2014) are used as sequences of the original packets. These data sets are created using a limited number of websites (100 and 128).

5.3.1 Evaluation against the WF-attack based on FFT

To assess the ability of the proposed defense model, we compare the performance of the WF-attack based on FFT (Jahani and Jalili 2016) with the original instances and new instances (i.e. sequences of new packets have been protected by applying the proposed defense). Moreover, the 10-Fold cross validation is applied with the SVM classifier based on the Polynomial kernel with parameter C=10000. The evaluation results in term of accuracy are shown in Table 1 and Fig. 8. The detection accuracy (i.e., the ratio of recognizing the websites) is calculated by relation (6).

$$Accuracy = \frac{TP}{TP + FN} \tag{6}$$

where TP, is the number of instances that are correctly classified as true and FN is the number of instances that are incorrectly classified false.

The results presented in Table 1 and Fig. 8 show that when no defense method is used, due to the high autocorrelation property of instances, a distinctive pattern related to instances of different websites is easily recognized and classified.

As a result, by increasing the number of websites there is no significant impact on classifying instances in different classes and the accuracy remains almost constant (Jahani and Jalili 2016). But in the case of using the proposed defense model, the accuracy of WF-attack is severely decreased due to controlling and reducing the autocorrelation property of instances that belong to different websites. As it is seen in Fig. 8, by increasing the number of websites, the detection accuracy gap between the two cases (i.e., with/without the proposed defense model) grows sharply which represent the effectiveness of the proposed defense model against WF-attack based on FFT.

Table 1 The accuracy of FFT-based WF-attack[14] before/after applying the proposed defense model on different number of websites: k indicates the number of websites, t is the number of instances of each website

$\#Websites$	Without any defense method		With proposed defense model	
	#Correctly Classified Instances (TP)	Accuracy of WF-attack	#Correctly Classified Instances (TP)	Accuracy of WF-attack
k=10 ,t=40	389	97.25%	251	62.75%
k=32 ,t=40	1230	96.09%	741	57.89%
k=50 ,t=40	1944	97.20%	1060	53.00%
k=64 ,t=40	2440	95.31%	1237	48.23%
k=100 ,t=40	3848	96.21%	1734	43.35%
k=128 ,t=40	4983	97.32%	1999	39.05 %
k=200 ,t=40	7752	96.91%	2948	36.85%

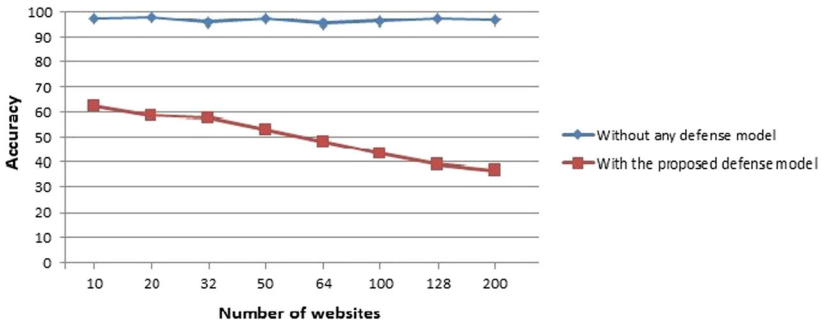

Fig. 8 The accuracy of FFT-based WF-attack with/without applying the proposed defense model

5.3.2 Evaluation of the effectiveness of the proposed defense model against the Known WF-attacks

In this section, by applying known WF-attacks[11] on the new instances derived from the proposed defense model, we evaluate the effectiveness of the proposed defense model on WF-attack in two scenarios, namely the closed and the open world scenarios. In the closed world scenario, the evaluation results in term of accuracy are shown in Table 2.

Table 2 shows the accuracy of the several known WF-attacks in which their least accuracy rate is 80% on the original data set. As shown in the last row of Table 2, by applying the proposed defense model, the accuracy rate of the known WF-attack drops sharply and reduces to the lowest possible value (i.e., between 9% and 23%).

In the following, we assessed the proposed defense model in open world scenario by KNN algorithm that is provided by Wang et al. (2014). In the open-world scenario we used 100 monitoring websites that includes 40 known instances for each website and also used 9000 non-monitoring website in which each of them contains only one unknown instance. We applied the 10-Fold cross validation on the KNN. The results of the evaluation of the proposed defense model in the open-world scenario is presented in Table 3. These results show a dramatic decline in the correct detection of the monitored websites (i.e. TPR). Note that data set that obtained from the proposed defense modal and the details of our evaluation result are provided at the URL:"https://github.com/h-jahani/Traffic-Analysis".

5.3.3 Comparison of the proposed defense model with effective current defense models

In this section, the advantages of the proposed defense model is compared with existing defense methods. As mentioned before, current defense models are based on three components: 1) traffic morphing, 2) control packet rate (i.e. control parameter ρ in previous defenses), and 3) Excessive use of packets padding. These defenses did not consider the most important vulnerability (i.e. the existence of autocorrelation property in instances) and just changed the length of packets to a fixed value, until the traffic of a given website is similar to the traffic of a specific website (i.e., target website) . Also, the packet rate depends on user's equipment and network characteristics (e.g. speed, bandwidth, etc), so although

[11]The source code of known WF-attacks are provided at URL:"https://www.cse.ust.hk/%7Etaow/wf/attacks/".

Table 2 The accuracy of known WF-attacks against the proposed defense model in the closed-world. Also the accuracy of the original accuracy of Known WF-attacks is presented for comparison (without any defense method). Number of websites is 100 websites and the number of instances of each website are 40

	Accuracy(%)				
	Pa-SVM	Dl-SVM	VNG++	KNN	Pa-CUMUL
No defense	82% (Panchenko et al. 2011)	87% (Cai et al. 2012)	80% (Dyer et al. 2012)	89% (Wang et al. 2014; Panchenko et al. 2016)	92% (Panchenko et al. 2016)
Proposed defense	11%	16%	13%	9%	23%

Table 3 The accuracy of KNN base WF-attack against the proposed defense model in the Open-world scenario

	KNN (Wang et al. 2014; Panchenko et al. 2016)	
	TPR (%)	FPR (%)
No defense	89.19%	6.4%
Proposed defense	1%	0%

controlling and resolving this feature, reduces the accuracy rate of WF-attack, but it will not improve security level of the Tor anonymity network directly. In the proposed defense model the arrangement and length of packets in different instances are changed such that their values follow a uniform distribution, Figs. 6b and 7b. These advantages let the similarity distance of instances of different websites be reduced such that the probability of distinguishing their distinctive patterns becomes hard.

As said before, most of the current defense model use data padding technique (i.e., adding dummy packets or dummy bytes to the end of packets) as the main way of making collisions between instances. But the proposed defense model merely uses dummy bytes at the end of packets for two main purposes: maintaining randomness property of the length of packets and the correct placement of encrypted bytes in packets. So, this approach significantly controls time and bandwidth overheads.

The proposed defense approach not only compensates security vulnerability of the Tor anonymity network with more suitable bandwidth and time overhead, but also it can be deployed in practice without reducing its efficiency. The results are presented in Table 4.

As shown in Table 4, compare with other defense models in the proposed model, the accuracy rate is reduced to the lowest value in all known WF-attacks and the bandwidth and time overhead are reduced as well.

5.4 Bandwidth and time overhead

Other metrics for the evaluation of the performance of defense approaches are: ”bandwidth overhead” and ” time overhead”. The bandwidth overhead of the proposed defense model is affected by three separate factors: 1) The number of dummy bytes inserted at the end of new packets, 2) The number of dummy packets transmitted within original packets, and 3) The header of new packets (according to Algorithm1 the number of new packets is almost 160% of the original packet).

According to Algorithm1, the overhead contribution of factors 1 and 2 is merely 5%. This amount of overhead is related to both dummy bytes and dummy packets. In other words, an average of eight dummy bytes are inserted at the end of all packets (given that block length of the cipher algorithm such as AES is a multiple of 16 and an average of 1 to 15 dummy bytes are inserted at the end of each packet); so statistically, 3% of overhead is related to dummy bytes and only about 2% is related to adding dummy packets.

Concerning the third factor, by fragmenting each original packet, 58-bytes of overhead are added as new packet header. The amount of bandwidth related to this header is about 2.7%. Hence, the total overhead of these three factors in the proposed defense model is less than 8%, which is very significant compared to the previous methods.

The time overhead of the proposed defense model is affected by the time gap between new packets. In this approach, the time gap depends on three parts: Preamble, Start of Frame and inter-packet gap which inserts 18 bytes of overhead for each new packet and causes an average time overhead of 1.03%. Consequently, the maximum time overhead in

Table 4 the results of comparison between the proposed defense model and current defense models

Defense models	Number of Websites	Pa-SVM (Panchenko et al. 2011)	DL-SVM (Cai et al. 2012)	Pa-CUMUL (Panchenko et al. 2016)	FFT-attack (Jahani and Jalili 2016)	Time Overhead	B.W Overhead
BUFLO (Dyer et al. 2012; Juarez et al. 2016)	100	14%	18%	N/A	N/A	145%	348%
CS-BUFLO (Cai et al. 2014)	120	31%	41%	N/A	N/A	173%	130%
Tamaraw (Juarez et al. 2016; Cai et al. 2014)	100	11%	19%	N/A	N/A	200%	38%
Walkie-talkie (Wang and Goldberg 2015)	100	51%	27%	N/A	N/A	15%	23%
WTF-PAD (Juarez et al. 2016)	100	15%	23%	N/A	N/A	zero	60%
D-ALPaCA (Cherubin et al. 2017)	100	N/A	22%	33%	N/A	42%	44%
Proposed defense	100	11%	16%	23%	43%	2%	8%
Proposed defense	128	11%	14%	23%	39%	2%	8%

the proposed defense model is less than 2%, As a result, the implementation and practical application of the proposed defense model in real anonymity networks is feasible.

6 Analysis and discussion

This study explored the autocorrelation property in web traffic instances as the principal factor of information leakage and success of WF-attack. Therefore, providing a mechanism to control this property in all instances of web traffic is inevitable in order to improve the robustness of the anonymity network against traffic analysis.

By examining a previous defense approach that is based on altering packet length to a fixed value (such as 1250 or 1500), control packet rate, and using padding component (i.e., inserting some dummy packets), the main bottlenecks of this approach are recognized as follow: 1) Increase of tens or hundreds percent of bandwidth/time overhead; hence, this approach is not practical against WF-attack, 2) facilitating DOS attack and the possibility of dropping all packets with the same length in the position of first relay (where the WF-attack occurs), 3) incompatibility with the TCP protocol (i.e., lack of interaction with TCP stack and not sensitive to congestion) , and 4) Not considering the autocorrelation property in instances.

In this study, a novel idea is proposed for traffic morphing to improve defense against traffic analysis attacks (i.e., WF-attack) that has a high generality (i.e., efficient for various classification attacks). The proposed defense model has a low overhead in bandwidth and time, therefore its practical use is possible. The idea is to randomize the arrangement and length of packets in all website instances. The information leakage from anonymity network is reduced as much as possible by controlling the autocorrelation property in all instances used for the loading of websites. Figure 8 and Tables 2, 3 and 4 show the highly effectiveness of applying the proposed defense model. By applying the proposed defense model, the features of WF-attack are vanished at a suitable level. Thus compared to others defense methods, the proposed defense method reduces the accuracy of WF-attack from 96% to the lowest value while using Full-duplex communication for the browser. By generation of m+n new packets simultaneously in the output of traffic distribution filter step, the management of packet transmission rate (i.e., the value of ρ) is truly possible. Consequently, the control parameter ρ can be added to decline the accuracy rate of the WF-attacks.

The approach of the proposed defense model (i.e., generating skeleton packets with random in terms of arrangement and the length) merely increases the number of packets with random lengths; hence the need to insert dummy packets falls sharply. Additionally, in previous approaches, the length of packets were changed to MTU or to some fixed values such as ”1250 or 1000” which requires the extensive use of dummy bytes to change the length of packets to the above value. However in the proposed defense model, the number of dummy bytes for each packet is at most 15 bytes.

7 Conclusion

In this study, a defense model is proposed to reduce peripheral information leakage in anonymity networks. This is achieved by controlling and reducing autocorrelation property in instances of loading websites. The proposed defense model has several unique features compare with previous methods: 1) its generality and feasibility to be used in various traffic analysis, 2) low bandwidth and time overhead, 3) compatibility with TCP protocol and

its dependence on traffic congestion, 4) resistance against DOS attacks (i.e., dropping all packets with same length), and 5) practical implementation of the proposed defense model on both sides of the communication independently.

The experiments showed that WF-attack accuracy and time and bandwidth overhead are reduced sharply from 42% and 44% to the lowest values 11% and to 8% respectively. Also, by equalizing buffered packets for all websites (i.e., 100 packets for all websites) and random changes in the initial status of the shift register, the ratio of hamming distance in relation (4) decline to about 1. For future works, WF-attack accuracy is conjectured to be reduced by adapting the defense model to benefit from some features of previous methods such as using Half-duplex communication.

References

Cai, X., Zhang, X.C., Joshi, B., Johnson, R. (2012). Touching from a distance: Website fingerprinting attacks and defenses. In: Proceedings of the 2012 ACM conference on Computer and communications security, pp 605–616.

Cai, X., Nithyanand, R., Johnson, R. (2014). Cs-buflo: A congestion sensitive website fingerprinting defense. In: Proceedings of the 13th workshop on privacy in the electronic society, pp. 121–130.

Cai, X., Nithyanand, R., Wang, T., Johnson, R., Goldberg, I. (2014). A systematic approach to developing and evaluating website fingerprinting defenses. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, pp. 227–238.

Cherubin, G., Hayes, J., Juarez, M. (2017). Website fingerprinting defenses at the application layer. *Proceedings on Privacy Enhancing Technologies*, *2017*, 186–203.

Dingledine, R., Mathewson, N., Syverson, P. (2004). Tor: The second-generation onion router, DTIC Document.

Dingledine, R., Mathewson, N., Syverson, P. (2008). Tor: anonymity online, ed,.

Dyer, K.P., Coull, S.E., Ristenpart, T., Shrimpton, T. (2012). Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In: 2012 IEEE symposium on security and privacy, pp. 332–346.

Golomb, S.W. (1967). Shift register sequences. Aegean Park Press.

Gu, X., Yang, M., Luo, J. (2015). A novel Website Fingerprinting attack against multi-tab browsing behavior. In: 2015 IEEE 19th international conference on computer supported cooperative work in design (CSCWD), pp. 234–239.

Hayes, J., & Danezis, G. (2016). k-fingerprinting: A robust scalable website fingerprinting technique. In: USENIX security symposium, pp 1187–1203.

He, G., Yang, M., Gu, X., Luo, J., Ma, Y. (2014). A novel active website fingerprinting attack against Tor anonymous system. In: Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD), pp. 112–117.

Herrmann, D., Wendolsky, R., Federrath, H. (2009). Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In: Proceedings of the 2009 ACM workshop on Cloud computing security, pp. 31–42.

Hintz, A. (2002). Fingerprinting websites using traffic analysis. In: International workshop on privacy enhancing technologies, pp. 171–178.

Jahani, H., & Jalili, S. (2016). A novel passive website fingerprinting attack on tor using fast fourier transform. *Computer Communications*, *96*, 43–51.

Juarez, M., Afroz, S., Acar, G., Diaz, C., Greenstadt, R. (2014). A critical evaluation of website fingerprinting attacks. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, pp 263–274.

Juarez, M., Imani, M., Diaz, C., Perry, M., Wright, M. (2016). Toward an efficient website fingerprinting defense for Tor. In: Lecture notes in computer science, pp. 27–46.

Luo, X., Zhou, P., Chan, E.W., Lee, W., Chang, R.K., Perdisci, R. (2011). HTTPOS: Sealing information leaks with browser-side obfuscation of encrypted flows. In: NDSS.

Murdoch, S.J., & Zieliski, P. (2007). Sampled traffic analysis by internet-exchange-level adversaries. In: International workshop on privacy enhancing technologies, 167–183.

Panchenko, A., Lanze, F., Pennekamp, J., Engel, T., Zinnen, A., Henze, M., et al. (2016). Website fingerprinting at internet scale. In: NDSS.

Panchenko, A., Niessen, L., Zinnen, A., Engel, T. (2011). Website fingerprinting in onion routing based anonymization networks. In: Proceedings of the 10th annual ACM workshop on Privacy in the electronic society, pp. 103–114.

Perry, M. (2011). Experimental defense for website traffic fingerprinting, Tor project Blog." https://blog.torproject.org/blog/experimental-defense-website-traffic-fingerprinting.

Shi, Y., & Matsuura, K. (2009). Fingerprinting attack on the tor anonymity system. In: Information and communications security, ed: Springer, pp. 425–438.

Sun, Q., Simon, D.R., Wang, Y.-M., Russell, W., Padmanabhan, V.N., Qiu, L. (2002). Statistical identification of encrypted web browsing traffic. In: Proceedings IEEE symposium, security and privacy, pp. 19-30.

Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I. (2014). Effective attacks and provable defenses for website fingerprinting. In: 23rd USENIX Security Symposium (USENIX Security 14), pp. 143–157.

Wang, T., & Goldberg, I. (2013). Improved website fingerprinting on tor. In: Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society, pp. 201–212.

Wang, T., & Goldberg, I. (2014). Comparing website fingerprinting attacks and defenses, Technical Report 2013-30, CACR, 2013. http://cacr.uwaterloo.ca/techreports/2013/cacr2013-30.pdf.

Wang, T., & Goldberg, I. (2015). Walkie-talkie: An effective and efficient defense against website fingerprinting.

Wright, C.V., Coull, S.E., Monrose, F. (2009). Traffic morphing: An efficient defense against statistical traffic analysis. In: NDSS.

Zhu, Y., Fu, X., Graham, B., Bettati, R., Zhao, W. (2005). On flow correlation attacks and countermeasures in mix networks. In: Privacy enhancing technologies, pp. 207–225.