An empirical study on the robustness of timing based traffic watermarks[⋆]

Yi Xie[1], Hui Feng[1], Louie Yan[1], Rui Xu[1], Xiapu Luo[2], and Guozhi Xue[1]

[1] Xiamen University, Xiamen, China
[2] The Hong Kong Polytechnic University, Hong Kong, China `lncs@springer.com`
`http://www.springer.com/gp/computer-science/lncs`

Abstract. Network Flow Watermarking (NFW) is an active approach of traffic analysis, which can effectively trace anonymous abusers and network attackers. But the existences of anonymous communication systems and various flow transformations/network interferences degrade/attenuate the performance of NFW. Then the robustness and non-detectability of NFW become indispensable/significant for real applications. However, the existing researches often study a specified NFW method against a few network interferences, but lack a uniform platform to objectively evaluate different NFW methods under different network interferences. Therefore, this paper analyzes seven representative NFW methods, and compares their performance in a controllable testbed where a self-established anonymous proxy server carries out different flow transformations, such as packet dropping, flow splitting, adding chaff, repacketization, and time perturbation. The experiment results show that the NFW methods based on Intervals generally outperform the ones based on Direct Sequence Spread Spectrum (DSSS) in terms of robustness. Moreover, we conduct a series of experiments to study the applicability/usability of NFW methods, which involve four commercial anonymizers: Tor, I2P, JonDo and Lantern. Under the interference introduced by these anonymizers, DSSS-based NFW methods are vulnerable, while interval-based NFW methods mostly survive.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

With the rapid development of Internet, there appears an endless stream of network security incidents which attack public web services, destroy client information and disclose personalized privacy, and then bring huge economic losses and serious social panic. In order to avoid detection and tracking, attackers do not directly attack targets, but often use the "stepping stone" host [8–10], anonymous communication systems [10–12], and botnet [13] to hide their real identities. Some criminals also use anonymous communication systems like Tor [14] and Shadowsocks [15] , to spread unhealthy or illegal information such as gambling,

[⋆] Supported by organization x.

Category	Method	Transmission efficiency	Difficulty of extraction	Robustness	Applicability
Interval-based	ICBW [1]	medium	high	high	high
	IBW [2]	low	high	high	medium
Inter-packet delay	IPD [3]	ex-high	ex-low	ex-low	ex-low
DSSS-based	DSSS [4]	high	low	low	medium
	Long-DSSS [5]	high	low	low	medium
	MMAR-SSW [6]	low	medium	medium	low
Hybrid	ICBSSW [7]	ex-low	high	high	high

Table 1: Characteristics of NFW Methods

pornography, and violence. All these means pose serious challenges to attacker localization, criminal forensics and network censorship. Therefore, how to effectively track network flows in complex networks has become an important issue of network security.

The traditional methods of traffic tracking like Intrusion Detection Systems (IDS) and Flow Correlation Technology (FCT) are mainly based on passive measurement. They collect network traffic through critical nodes, analyze the data characteristics (i.e. the packet header, number of packets [16], packet size [17], time sequence [18, 19]), and reveal the correlation and relationships among network flows. However, these methods using off-line analysis mode, results in high overhead on traffic collection and data analysis, poor real-time performance, lack of scalability, and vulnerability against network interferences due to anonymous communication systems. Therefore, an active and effective traffic tracking method is urgently needed.

Network Flow Watermarking (NFW) is an active approach of traffic analysis, where special marking information (i.e. watermarking) is embedded into network flows by adjusting data characteristics [20]. There are two main steps: (1) when transiting traffic, a NFW encoder/sender can change or modulate the traffic load, time intervals, and transmission rate to embed watermarking; (2) a NFW detector/receiver is responsible for detecting whether there is a watermarking embedded in the given traffic. Compared with traditional methods, NFW can effectively track network attackers and anonymous abusers with shorter tracking time, lower cost of computational and communication.

NFW still face the challenges of various flow transformations due to anonymous communication systems, such as packet dropping, flow splitting, adding chaff, repacketization, and time perturbation. Then researchers pay a lot attentions in the performance evaluation of NFW, which studies the robustness and non-detectability of NFW, where flow transformation is often regarded as network interference. However, the existing researches often study a specified NFW method against a few network interferences, but lack a uniform platform to objectively evaluate different NFW methods under different network interferences. It is difficult to objectively evaluate and compare the performance of different NFW methods under different interference factors.

To solve this problem, this paper implements and analyzes seven representative NFW methods that belong to three main categories of NFW techniques:

(1) Interval based, (2) Direct Sequence Spread Spectrum (DSSS) based, and (3) Hybrid, shown in Table 1. At the best of our knowledge, this is the first work establishing a unified testbed for a series of experiments where typical flow transformations [14, 15, 16, 17]are controlled by a customized anonymous proxy server, and studying the impacts of commercial anonymizers on these NFW methods. Firstly, we evaluate the performance of NFW methods by calculating accurate rates and detection true positive rates, when varied intensities of flow transformations are launched by the customized anonymous proxy. Secondly,

From the experimental results, we have acquired some important findings:

- DSSS-based NFW methods work well under most of flow transformations except flow splitting and timing perturbation.
- IPD-based NFW methods are sensitive to correspondence of pairs of packets and can only withstand a certain range of timing perturbation. The robustness of interval-based methods is better than that of DSSS-based methods. Especially, the advantages of interval-based methods become more and more obvious when the intensities of flow transformations increase.
- Some commercial anonymizer like Lantern and JonDo can be easily penetrated using NFW methods, while I2P is such that all the NFW methods can hardly survive.
- Hybrid method (ICBSSW) has the best overall performance on commercial anonymizers, while interval-based and DSSS-based methods show no significant difference.
- IPD has the poorest practicability because of their low veracity under the interferences introduced by commercial anonymizers.

The rest of the paper is structured as follows. Section II briefly introduces the basic ideas of seven representative NFW methods. Section III presents . Section IV deals with ., and Section V presents numerical results collected in a real-world testbed. Lastly, Section VII concludes the paper.

2 Related Work

In this section, we first give an overview of early network traffic tracing techniques, and then introduce the basic principles of the Network Flow Watermarking, or NFW, and finally the classification and details of seven NFW techniques involved in this paper.

Early techniques used to trace network traffics (e.g. a network attack flow) modify the basic fields in the IP packets including the timestamp, packet size and some other headers [1][2][3], to embed information; some techniques just simply add extra functions to the routers, enabling them to trace the route of packets. Apparently, such techniques rely heavily on the networks infrastructures and require analysis on a very large amount of network traffic data; therefore, it is of high cost but low efficiency. Plus, these techniques could hardly trace under the circumstance where encryption, mix network[5] or some other anonymization techniques are used; the information embedded in the original headers of IP packets can be easily eliminated.

Wang and Reeves [4] have published their work on NFW for the first time. NFW is a kind of technology that modulates the timing character, which would barely change in transmission, of one or multiple network flows (e.g. the speed of the flow, intervals of each packets, etc.) to embed some secret messages or special marks (i.e. watermarks). By using NFW, we can identify and trace the flows through general network and mix network. The general steps of NFW are the followings.

a) Injection. The payloads, intervals and speed of network flows are modulated by an injector on the router side to embed watermarks.

b) Detection. A detector is used to check the network flows of interest for any possibly embedded watermarks.

Compared with the traditional tracing technique, the new technique, NFW, is much more secretive and efficient for its low cost and fast locating. Although there are some countermeasures proposed by Xiapu Luo et al. [21, 22] to detect or eliminate NFWs, the secrecy of NFW methods is out of the scope of this paper.

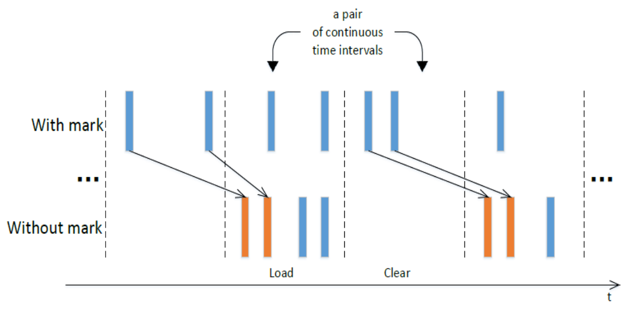
In this paper, seven NFW methods are involved in. They can be classified into the followings, DSSS-based (DSSS, short for Direct Sequence Spread Spectrum), interval-based and hybrid, where a) DSSS-based methods is the methods that modulate the speed of network flows utilizing Pseudo-Noise code; b) interval-based methods modulate the intervals of packets of network flows; c) hybrid methods combine both DSSS-based methods and interval-based methods.

2.1 Classification of NFW

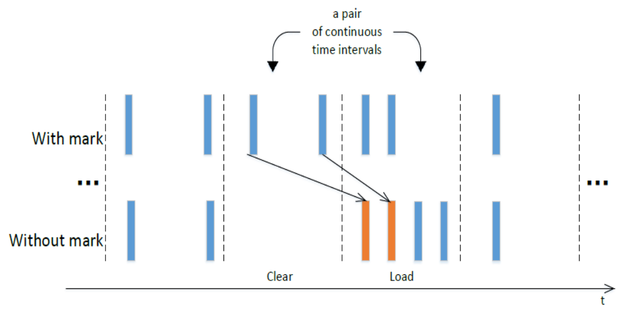
Interval-based In the so called interval-based method [2] (or interval based schemes), a flow is divided into intervals, and all packets within selected intervals are shifted to form a watermark pattern. The most representative interval-based method is Interval Centroid Based Watermarking (ICBW) proposed by Wang et al. [1]

Given a random timing offset, we can select 2*r regular intervals from the flow. Then randomly divide those intervals into two groups, A and B, each containing r intervals. By "squeezing" the packets within intervals of group A or B, centroids (quantitative) [1] can be shifted. We can make the difference between average centroids of group A and B symmetric around zero to embed watermark bit "1", or on the contrary to embed watermark bit "0". Being able to self-synchronize during the decoding process is one of the most distinct features of this method.

Interval-based Flow Watermarking [23] (IBW) embeds watermark bit in a similar approach, but with the following difference: in order to encode a single watermark bit, a pair of continuous time intervals should be selected randomly, then, instead of just "squeezing" one particular interval, make the difference between the numbers of packets within two intervals positive (or negative) by "loading" [23] the first interval and "clearing" [23] (analogous to "squeezing") the second interval (or “clearing” the first interval and “loading” the second interval) to embed watermark bit “1” (or “0”).



(a) To embed watermark bit “0”.



(b) To embed watermark bit “1”.

Fig. 1: interval fragments

Inter-packet Delay Compared with ICBW and IBW, Inter-packet Delay Watermarking (IPD) [3] divides flows by the number of packets while ICBW and IBW both divide flows into intervals by time. As shown in Figure 2, in each interval, $4s$ packets are divided into two halves containing $2s$ packets. The ith packet and $i + 2s$th packet are a pair of packets. Randomly divide $2s$ pairs into two groups containing s packets. Define IPD$_{diff}$ as the difference between timestamps of a pair of packets. To embed a single watermark bit “1” (or “0”), the difference between the averages of two groups’ IPD$_{diff}$ should be positive (or negative).

DSSS-based DSSS-based method, as its name implies, embeds watermark bits using a typical communication technique, Direct Sequence Spread Spectrum, or

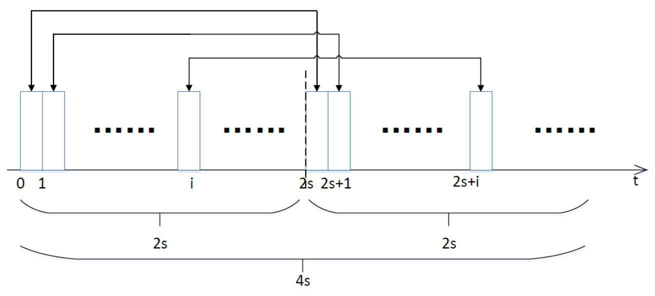


Fig. 2: IPD

DSSS for short, which utilizes a Pseudo-Noise (PN) code to “encode” the original watermark bits into so-called “signals”.

DSSS [4], utilizing an m-sequence [24], embeds the NFW by adjusting the flow rate of the data flow (such as constant bit rate stream, CBR stream for short). As shown in Figure 3, the original signal d$_t$ at the transmitter is a series of binary symbols (here we use bits encoded as +1 or -1 instead of 1 or 0). A PN code c$_t$ of +1 and -1 is generated at the transmitter and shared between the receiver. Then the original signal is “spreaded” [4] by PN code c and the transmitted signal t$_b$ is produced. Assume that there is no interference along the channel, the received baseband signal r$_b$ = t$_b$. Then the recovered signal (using c to “despread” [4] r$_b$) d$_r$ = d$_t$.

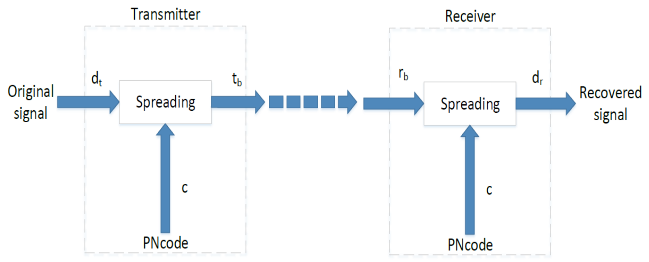


Fig. 3: Spreading and Despreading in DSSS

Long-DSSS [5] uses long PN code instead of m-sequence against MSAC [25] (mean-square autocorrelation). Therefore, Long-DSSS has no periodic pattern which could be detected by attackers.

MMAR-SSW [6] (MSAC and Multi-flow Attacks Resistant Spread Spectrum Watermarks) uses orthogonal PN code and embeds watermark signals to be transmitted by random interval positions. In this schema, MMAR-SSW can endure MSAC and multi-flow attacks [6].

Hybrid ICBSSW [7] combines the encoding of DSSS and the modulation of ICBW. That is, the original watermark bits are “spreaded” by PN code before being sent by the ICBW modulator. Therefore, this hybrid NFW method has high information redundancy inheriting from DSSS-based method.

2.2 Network Interferences

In order to anonymize users’ traffics, anonymous services will apply extra interferences on the flows running through, no matter how the original network condition is, by using “stepping stones” and modulating the packets in some unknown ways, which are attacks against both the traditional tracing techniques and the cutting-edge NFW methods.However, such intended interferences do serve the purpose of anonymization, but at the expense of network performance, whereas the interferences added by the anonymizers have to be acceptable to the users. In other words, the intended interferences cannot eliminate all the NFW methods without sacrifice of network performance; and therefore, some of the watermarks could survive the attack, and the resistances to all kinds of interferences are the performance indexes of all the NFW methods.

In this paper, five kinds of interferences are considered for controlled experiments: 1) Timing perturbation, 2) Packet dropping, 3) Adding chaff, 4) Flow splitting and 5) Repacketization.

When timing perturbation is applied to a packet, the transmission of this packet will be delayed for some random finite time in the timing interval, say *(0, T]*, where T is a finite positive number. It is one of the most common interferences in the world of network, and it usually comes with packets disorder.

Packet dropping could be considered as a special case of timing perturbation. Assume T could be infinite, i.e. the packets being delayed for infinite time are lost. Generally, unintended packet dropping comes with timing perturbation, as well as packets disorder. To simplify, the router (or the firewall) randomly drops packets of a flow with a probability of, say, $(\mu)_d$, where $(\mu)_d$ is a positive number that is less than 1, and the packets dropped are uniform distributed.

In the contrary, some packets could be added to a flow. For NFW methods, those packets are unexpected and could form an attack, whether they are chaff packets or not; for both sides of communication, those packets could be just chaff packets whose payloads are useless for communication (chaff addition), part of the original flow (flow merging), or even packets from another independent flow (flow mixing). To simplify the configuration of the experiments, flow merging and flow mixing will be considered as the same as chaff addition as all of them are adding non-NFW-related packets to the flow of interest. In the experiments, the number of chaff packets added to a flow is relative; it accounts for, say $_c$, where $_c$ is a proportion, of the number of packets in the flow.

Flow splitting is quite similar to packet dropping, but with different strategies. For the former, a flow could be split into multiple sub-flows, say s flows, where s is an integer that is no less than 2, then the s sub-flows are merged somewhere else to recover the original flow. The strategy of flow splitting can be software or hardware specific, or completely unknown. To a NFW method, flow splitting is equivalent to packet dropping as some packets involved in NFW embedding are transmitted on different routes, but it is still distinguishable comparing to packet dropping so that they have to be considered individually.A flow could be split into multiple sub-flows, say s flows, where s is an integer that is no less than 2.

Finally, anonymizers may adjust the MTUs of the "stepping stones" to apply an repacketization interference, forcing the packets to be reconstructed, and therefore changing both the number and the time of transmission of the packets. Similarly, in the experiments, the number of packets being repacketized is relative, too; it accounts for, say $(\mu)_r$, where $(\mu)_r$ is a proportion, of the number of packets in the flow.

3 Experiment platform

In this section, we build a platform for our controlled experiments to evaluate the robustness of seven watermark methods mentioned in section 2. The following parts consist of the structure of the platform, the design of the experiments including the detailed parameters, the robustness metrics, and then the result and analysis of the experiments. Finally, we evaluate the applicabilities of watermark methods on commercial anonymizers.

3.1 The Structure of the Platform

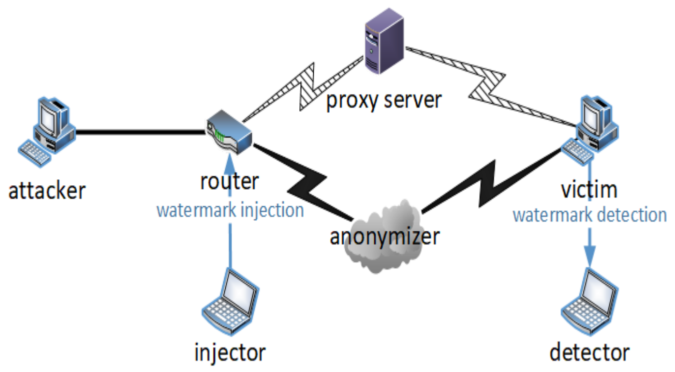


Fig. 4: Structure of the experimental platform

As shown in Figure 4, the platform consists of a simulated *attacker*, a *proxy server* in controlled experiments or a commercial *anonymizer* on real testbed and a simulated *victim*. In this scenario, the *attacker* attacks the *victim* through the *proxy server* or *anonymizer*. In order to trace the *attacker*, we embed watermarks into the attacking flow near the *attacker* and try to detect the watermarks near the *victim*. The configurations of the devices involved in the experiments are as follows: 1) The *attacker*, *victim* and the *proxy server* are cloud servers with exactly the same configuration: a single-core x86-64 CPU, 2 GB memory, 1 Mbps bandwidth and CentOS 7.4 as the operating system. 2) The *proxy server* forwards the traffic between the *attacker* and *victim*, and applies five flow transformations respectively. 3) The *anonymizer* is one of the most popular commercial anonymizers: Tor [26], Lantern [27], JonDo [28] and I2P [29], with HTTP or SOCKS tunnels.

3.2 Experiment Design

We use iperf3 [30] to generate target constant bit rate (CBR) flows; the size of each packet is 150 bytes. Then, we modulate the target flows to embed watermarks before they go through the *proxy server* or the *anonymizer*. In case of *proxy server* that under our control, five flow transformations with different intensities are applied respectively. Finally, we capture and try to recover the watermarks near the victim for further analysis.

As we want to evaluate and compare the robustness of watermark methods, we must ensure all the watermarks are in their "best" condition before going through the *proxy server* or the *anonymizer* so that flow transformations applied to target flows are limited within the *proxy server* and the *anonymizer*. The parameters of watermarks are shown in Table 2, where L denotes length of watermark, N_c denotes length of PN code, T_c denotes chip duration, T denotes interval, r denotes redundancy, t denotes threshold and o denotes offset. These parameters are recommended values or modified values based on the recommended values in the papers [1–7, 23–25] that propsed these NFW methods.

Method	L (bit)	N_c (bit)	T_c (s)	T (s)	r	t (bit)	o (ms)
ICBW IBW	32	/	/	600	18	6	2000
IPD	32	/	/	/	18	6	10
DSSSLong-DSSSMMAR-SSW	7	7	1	/	/	2	/
ICBSSW	7	7	/	600	18	2	2000

Table 2: Parameters of NFW Methods

3.3 Experiment Metrics

Our experiments include the following three performance metrics:

- 1) Average detection rate (δ): the proportion of successfully detected watermarks to all watermarks transmitted under the same configuration. The higher the detection rates, the higher the penetrability of NFW method.
- 2) Average bit correct rate (β): the average bit correct rate to all watermarks transmitted under the same configuration, which has a positive correlation with average detection rate, but comes with more details about the performance of NFW methods.
- 3) False positive rate (ψ): the proportion of false detections to non-watermarked flows. If a NFW method achieve a high detection rate and a low false positive rate, then it is considered to have high confidence level and applicability.

4 Experiment result and analysis

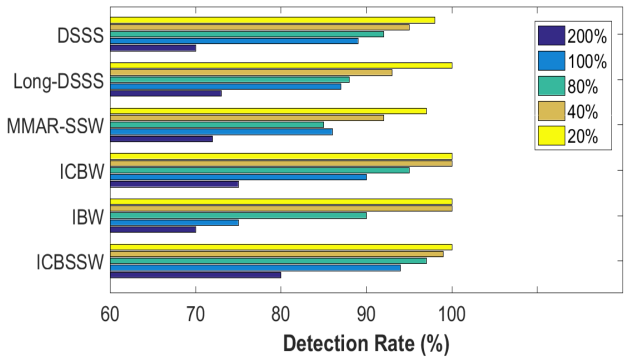


Fig. 5: Detection rate under chaff adding

4.1 Chaff added

Figure 5 and Figure 6 show that the detection and bit correction rates of the Interval-based NFW methods gradually decrease as the interference intensity increases. For example, when the proportion of chaff reaches 100% , the detection rate of IBW sharply drops to 75%. IBW uses the difference in the number of packets in a pair of intervals to embed watermark signals, which are sensitive to the number of packets. However, when faced with chaff added interference, it is best to use ICBSSW.. For example, when the proportion of chaff added reaches 100% ,the detection rate is still as high as 80%. ICBSSW, combining the advantages of the ICBW and DSSS methods, makes full use of the centroid difference and PN Code to improve the ability of the watermark to resist chaff added interference. IPD cannot resist to chaff added. Assume one of the original pairs of IPD packets is, say $< P_m, P_n >$; then ipd(m,n) = tm-tn. After chaff is added, the

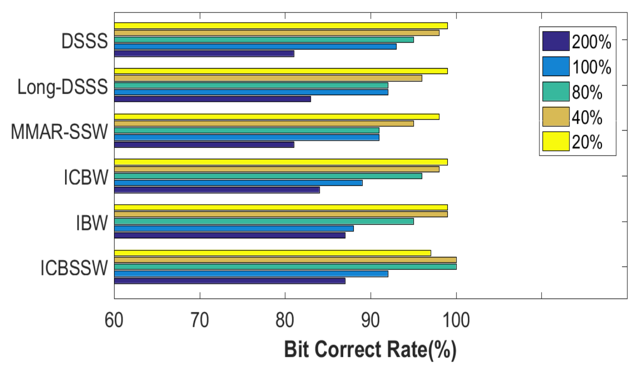


Fig. 6: Bit correct rate under chaff adding

pair of packets is changed to, say, $< P_m, P_k >$ (k != n). Then ipd(m,n) = tm-tk; Therefore, the result of ipd(m,n) is changed. Furthermore, the computation of IPD depends on the order of each packet pair. Therefore, because the order of the changed pairs is completely different from the original order, the watermark is unrecoverable. DSSS-based methods embedded watermark dependents on the rate of traffic, random chaff add cause a changed in the relative rate of traffic in a chip duration and the signal of +1 or -1 will be changed. The only different of DSSS-based NFW methods is the type of PN Code, therefor their ability to resist chaff add interference is similar. Experimental results prove that the higher the proportion of chaff, the lower the detection rate of all the methods. As long as the chaff packet remains within a certain range, the impact is limited.

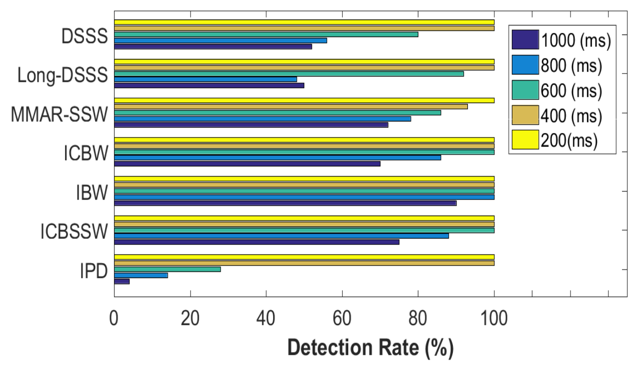


Fig. 7: Detection rate under timing perturbation

4.2 Timing perturbation

Figure 7 show the impact of Timing perturbation on detection and bit correct rates. All packets were randomly delayed from 0 ms to D ms (D is the maximum delay that an attacker can add to a packet). When the timing perturbation increases, the detection and bit correction rates decrease, resulting in a negative correlation between perturbation and the rates. all the NFW can resist to the interference below 200ms, when the Timing perturbed is greater than 400ms, their performance is significantly different. IBW and SWIRL perform better when faced with Timing perturbation. For example, when the delay time reaches 800ms, the detection rate is still as high as 100%. For IBW and SWIRL, as long as the random delay does not exceed a certain range, there are enough packets left in the interval, the watermark signals will not be changed. However, When the Timing perturbed is greater than 600ms, the detection rate of ICBW and ICBSSW dropped sharply. Random delay lead to the number of packets changed in the time intervals and the difference of a pair of the centroids of intervals will become small. Experimental results show that IPD cant resist the delay above 400ms. As to Long-DSSS and DSSS, Timing perturbed lead to the number of packets was changed randomly in a chip duration and the rate of flow in a chip duration was changed randomly. Because the watermark signal of LDSSS and DSSS are continuous, this effect will be superimposed on the subsequent watermark signals, the MMAR-SSW embedded watermark signal randomly has not the effect of superimposed. Experimental results prove that Interval-based NFW methods more power than DSSS-based NFW methods in resisting Timing perturbation.

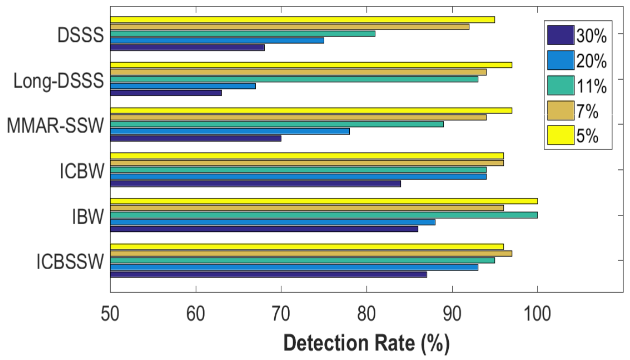


Fig. 8: Detection rate under packet dropping

4.3 Packet dropped

In terms of Interval-based NFW methods, the detection rate decreases to varying degrees as the proportion of packet loss increases, as shown in Figure 8. Packet dropped lead to the number of packets changed in the time intervals, so all the methods related to the number of packets are affected, such as IBW, ICBW, ICBSSW, IPD, SWIRL etc. When the packet loss rate is lower than 20%, the detection rate of each method is maintained at about 90%, and it has better resistance. Because there are enough packets left in the interval for each Interval-based NFW method, the watermark signals will not be changed. However, when the packet loss rate exceeds 20%, the detection rate of each method drops sharply. Packet dropped lead to packets out of order, so IPD cant resist this interference. When the packet loss rate exceeds 30%, ICBW has the lowest detection rate. Because a large number of packet loss results in insufficient number of remaining packets in the interval, so the difference of a pair of the centroids of intervals will become small. DSSS-based embed watermarking by changed the rate of flow, although packets loss following uniformly distribution in the watermarked flow, the flow rate difference between +1signal and -1 signal is not large, the relative rate will be changed when not enough packets are left. So, the detection rate sharply decreases with increasing packet loss rate, even to about 60%. Experimental results prove that Interval-based NFW methods are more power than DSSS-based NFW methods in resisting packet dropped, and all the NFW methods will fail when the packet loss rate continues to increase.

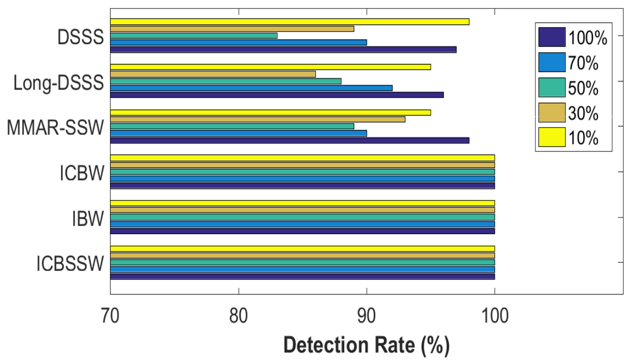


Fig. 9: Detection rate under repacketization

4.4 Repacketization

Figure 9 showed that repacketization has no effect on Interval-based NFW methods, such as IBW, ICBW, ICBSSW, SWIRL, their detection rate and bit correct rate are always maintained at around 100% within the interference range. However, In terms of IPD, repacketization also lead to packets out of order, so IPD cant resist repacketization. Repacketization resulted in a change in the number of packets, but each signal embedded by Interval-based NFW methods is redundant by many packets, therefore has a strong resistance to repacketization. The ability of DSSS-based NFW methods to resist repacketization is weaker than that of Interval-based NFW methods. When the interference range does not exceed 50%, the detection rate gradually decreases as the proportion of recombination increases. But when the interference range exceed 50%, the detection rate gradually rises. Because for relative rates, 80% repacketization and 20% repacketization have the same effect on the watermark are the same. Especially, when the probability of repacketization reach 100% that means all of packets will be reconstruction then the relative rate of watermarking and target flow will not be changed.

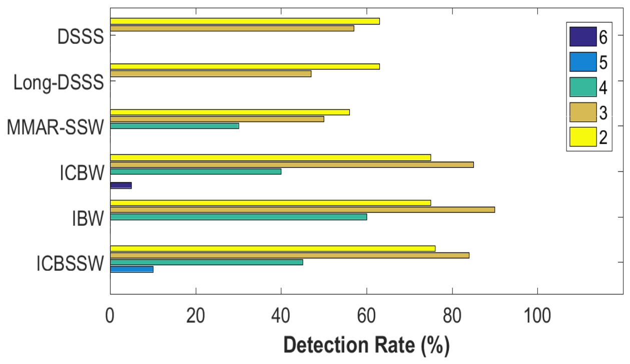


Fig. 10: Detection rate under flow splitting

4.5 Flow splitting

Flow splitting is a countermeasures based on packet dropped. Figure 10 showed that the effect of flow splitting on signal exceeds the above four types of interference. Both Interval-based NFW methods and DSSS-based NFW methods cannot resist more than three flow splitting interferences. For example, when the flow is divided into four, the detection rate of the Interval-based NFW method is reduced to below 60%, and DSSS-based NFW method is reduced to below 30%. Flow splitting changed the order of packets, the computation of IPD have a great deviation result in watermarking distortion. Flow splitting lead to a large number of packets in the intervals of signal was lost. For example, when the flow splitting is two, which means half of the packets in the watermarked flow will be dropped, this will result in insufficient packets in the interval. So both centroids and flow rate in the interval will change, and signal will distortion. Interval-based NFW methods is slightly better than DSSS-based NFW methods in resisting flow splitting, but when the flow splitting more than three all the NFW methods fail especially the DSSS-based NFW methods.

5 Analysis of real testbed

In this section, we use the platform introduced in Section 3 to conduct experiments on four commercial anonymizers: 1) Tor, 2) Lantern, 3) JonDo and 4) I2P respectively, to evaluate the performance of the above NFW methods under real applications. Also, false positive rates (ψ) are measured on a typical commercial anonymizer — Tor. The following parts of this section first show the experimental results and analysis, and then the evaluation on the anonymizers.

5.1 Experimental results

Category	Tor	Lantern	JonDo	I2P
DSSS	86.67%	100.00%	96.88%	30.00%
LongDSSS	68.33%	100.00%	100.00%	35.48%
MMAR-SSW	73.33%	86.21%	93.33%	12.90%
IBW	70.97%	100.00%	53.33%	3.23%
ICBW	70.00%	100.00%	100.00%	3.23%
ICBSSW	93.33%	100.00%	100.00%	48.28%

Table 3: Results on real testbed

The detection rates (δ) of NFW methods on commercial anonymizers is shown in Table 3. Note that the detection rates of IPD are zeros except that on Lantern. Overall, the detection rates of interval-based NFW methods approximate to those of DSSS-based methods while ICBSSW method, which is the only hybrid method involved, works better as expected. Notably, MMAR-SSW achieved lower detection rates than other DSSS-based NFW methods. This may be caused by the difficulty in synchronizing the signals.

On the other hand, the anonymizer that is most likely to be penetrated using NFW methods is Lantern, followed by JonDo, Tor and then I2P. As we can infer from the table, all NFW mehtods become impractical in I2P. However, I2P has a very limited bandwidth, meaning that it exchanges user experiences for anonymity. Notably, it seems harder for NFWs to penetrate Tor than JonDo, which is contrary to our trial tests. After checking the detailed results of experiment, we found that Tor is unstabler than JonDo. This may be caused by the random circuit for every several connections, resulting in impacts of connections on NFWs polarizing.

As discussed above, if target anonymizer is Lantern or JonDo, most NFW methods involved, whose detection rates can reach from 80% to 100%, are applicable. For other anonymizers, NFW mehtods we studied are too vulnerable to apply.

At the same time, false positive rates (ψ) of DSSS, Long-DSSS, MMAR-SSW and ICBSSW measured on Tor are around 20%, while those of IBW, ICBW and IPD are around 0%. As we can easily calculate, given the length of watermark L and the threshold t, we have the false positive rate:

$$\psi = \sum_{i=0}^{t} (\frac{1}{2})^L \binom{i}{L} \tag{1}$$

When L is 7 and t is 2, we have the false positive rate of about 22.66%; when L is 32 and t is 6, the false positive rate is about 0.03%. So this result is as expected. As a simple conclusion, higher threshold leads to higher false positive rate. To achieve better performance, tracers should adjust threshold and redundancy of NFW to fit specific network environment.

5.2 Analysis on anonymizers

According to the I/O graphs which reflects the distortion of watermarks of the above results, we found different anonymizers distort flows differently. As a hypothesis, some intended flow transformations, such as timing perturbation, repacketization, etc., are added by the anonymizers. In the following part, we will discuss the intensities of intended flow transformations.

Anonymizer	Maximum ARTT (ms)	Minimum ARTT (ms)	Average ARTT (ms)	Average Standard Deviation
Control Group	224	282	223.83	0.78
Tor	406	93464	3011.27	127.91
Lantern	262	580	254.29	5.96
JonDo	434	1133	599.90	68.09
I2P	1051	134225	28282.91	10590.81

Table 4: Average and deviation of ARTTs

Why do we design ARTT? To measure the intensity of timing perturbation in a specific network, we usually use Round-Trip Time (RTT). However, we can not measure the RTT of the connection between two ends directly on transportation layer in our experimental environment. The main reason is that the connection between the two ends is not established directly but through a tunnel (either HTTP or SOCKS) provided by a specific anonymizer, thus the

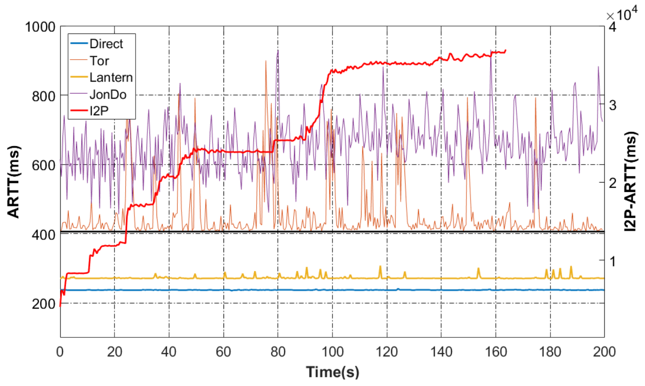


Fig. 11: ARTTs

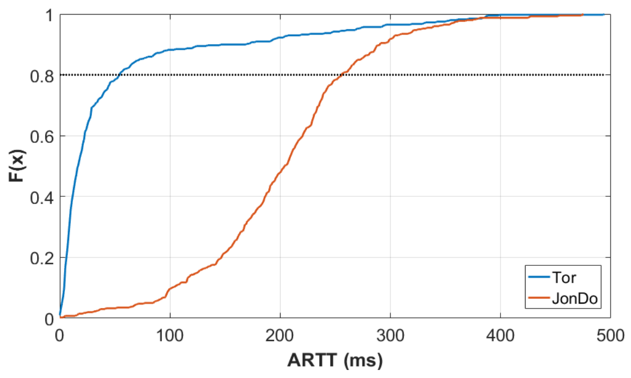


Fig. 12: CDF of ARTTs of Tor and JonDo

acknowledgements are not from the other end of connection but the entry or exit node. Plus, we have no access to the circuit, or stepping stones, of a specific anonymizer between the two ends.

What is ARTT? As shown in Figure 13, for example, the client sends a TCP packet ① to the entry of anonymizer so that the anonymizer can forward the packet to the desired server and the entry replies an acknowledgement packet ② to the client. The RTT is as shown in Figure 13. For we cannot measure the overall RTTs in the circuit on transportation layer, thus, we define and measure an Application-Layer Round-Trip Time, or ARTT, as: the time between the sending of an application-layer packet and the receiving of the corresponding application-layer acknowledgement. As shown in Figure13, the client sends a packet in application layer to the server proxied by the anonymizer (①, ③, and ④), then the server replies an acknowledgement packet in application-layer to the client proxied by the anonymizer (⑤, ⑥ and ⑦). As soon as the client re-

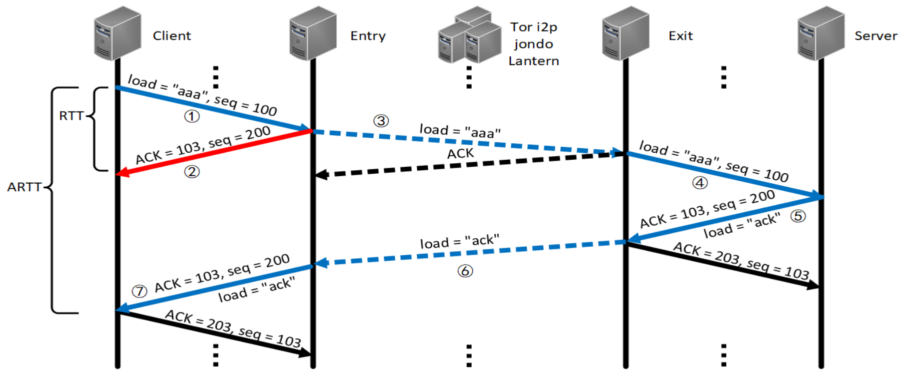


Fig. 13: RTTs and ARTTs

ceives the acknowledgement, it calculates ARTT. The more the ARTTs fluctuate in anonymizers, the worse the watermarks being distorted.

Result and analysis of ARTT The experiments are done on the same real testbed mentioned above, we measured the ARTTs of connections proxied by different anonymizers and direct connections as control group. Note that the observed packet losses, retransmissions and out-of-orders are few enough to be ignored. It's not the ARTT itself that impact on the traffic, but the fluctuation of ARTT. High yet stable ARTTs move the watermark as a whole while unstable ARTTs distort watermark. Therefore, the deviations of ARTTs are calculated, by which we can rank the anonymizers. The results are shown in table 4, which corroborate the previous conclusion in general. The ARTTs of control group and connections proxied by Lantern are much more stable compared to those of connections proxied by other anonymizers. Although the average standard deviation of ARRTs of connections proxied by Tor is a little higher than that of ARTTs of connections proxied by JonDo, the ARTTs of connections proxied by Tor have obvious baseline (the black line in Figure 11); those pulses might be a result of the long circuits of Tor instead of some anonymization strategies. The overlapped Cumulative Distribution Functions (CDF) of the ARTTs of Tor and JonDo are shown in Figure 13. Eighty percent of ARTTs of Tor are within the range [0, 54] while eighty percent of ARRTs of JonDo are within the range [0, 255], which is much more dispersed. As a result, despite of the average deviation of ARTTs of Tor is higher than that of JonDo. At the same time, as we can infer from the graphs in Figure 11 and Figure 13, JonDo seems to have intended timing perturbation added to flows running through it while Tor and Lantern seem not. The ARTT of a connection proxied by I2P increases with time, which might be a result of the very limited bandwidth in I2P.

6 Discussion

Acknowledgment

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks …". Instead, try "R. B. G. thanks…". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

1. X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Security and Privacy, 2007. SP '07. IEEE Symposium on*, 2007, pp. 116–130.
2. X. Gong, M. Rodrigues, and N. Kiyavash, "Invisible flow watermarks for channels with dependent substitution, deletion, and bursty insertion errors," *IEEE Transactions on Information Forensics & Security*, vol. 8, no. 11, pp. 1850–1859, 2013.
3. Z. Pan, H. Peng, X. Long, C. Zhang, and Y. Wu, "A watermarking-based host correlation detection scheme," in *International Conference on Management of E-Commerce and E-Government*, 2009, pp. 493–497.
4. W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *IEEE Symposium on Security and Privacy*, 2007, pp. 18–32.
5. J. Huang, X. Pan, X. Fu, and J. Wang, "Long pn code based dsss watermarking," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 2426–2434.
6. L. Zhang, Z. Wang, Q. Wang, and F. Miao, "Msac and multi-flow attacks resistant spread spectrum watermarks for network flows," in *IEEE International Conference on Information and Financial Engineering*, 2010, pp. 438–441.
7. J. Luo, X. Wang, and M. Yang, "An interval centroid based spread spectrum watermarking scheme for multi-flow traceback," *Journal of Network & Computer Applications*, vol. 35, no. 1, pp. 60–71, 2012.
8. Y. Zhang and V. Paxson, "Detecting stepping stones," in *Conference on Usenix Security Symposium*, 2000, pp. 13–13.
9. R. Shullich, J. Chu, P. Ji, and W. Chen, "A survey of research in stepping-stone detection," *International Journal of Electronic Commerce Studies*, vol. 2, no. 2, pp. 103–126, 2011.
10. A. Blum, D. Song, and S. Venkataraman, "Detection of interactive stepping stones: Algorithms and confidence bounds," *Recent Advances in Intrusion Detection*, vol. 3224, pp. 258–277, 2004.
11. C. Shields and B. N. Levine, "A protocol for anonymous communication over the internet," in *Proc of the Acm Conference on Computer & Communication Security*, 2000.
12. M. J. Freedman, E. Sit, J. Cates, and R. Morris, "Introducing tarzan, a peer-to-peer anonymizing network layer," *Lecture Notes in Computer Science*, vol. 2429, pp. 121–129, 2002.
13. T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and detecting fast-flux service networks," 2008.
14. R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," *Journal of the Franklin Institute*, vol. 239, no. 2, pp. 135–139, 2004.

15. "Shadowsocks: A fast tunnel proxy that helps you bypass firewalls." https://github.com/shadowsocks/.
16. T. He and L. Tong, "Detecting encrypted stepping-stone connections," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1612–1623, 2007.
17. K. Yoda and H. Etoh, "Finding a connection chain for tracing intruders," pp. 191–205, 2000.
18. D. L. Donoho, A. G. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford, "Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay," *Lecture Notes in Computer Science*, pp. 17–35, 2002.
19. X. Wang, D. S. Reeves, and S. F. Wu, "Inter-packet delay based correlation for tracing encrypted connections through stepping stones," in *European Symposium on Research in Computer Security*, 2002.
20. X. Wang and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *ACM Conference on Computer and Communications Security*, 2003, pp. 20–29.
21. X. Luo, P. Zhou, J. Zhang, R. Perdisci, W. Lee, and R. K. C. Chang, "Exposing invisible timing-based traffic watermarks with backlit," in *Twenty-seventh Computer Security Applications Conference*, 2011.
22. X. Luo, J. Zhang, R. Perdisci, and W. Lee, "On the secrecy of spread-spectrum flow watermarks," in *European Conference on Research in Computer Security*, 2010.
23. Y. J. Pyun, Y. Park, D. S. Reeves, X. Wang, and P. Ning, "Interval-based flow watermarking for tracing interactive traffic," *Computer Networks*, vol. 56, no. 5, pp. 1646–1665, 2012.
24. S. W. Golomb, *Shift Register Sequences - A Retrospective Account.* Springer Berlin Heidelberg, 2006.
25. W. Jia, F. P. Tso, Z. Ling, and X. Fu, "Blind detection of spread spectrum flow watermarks," in *INFOCOM*, 2009, pp. 2195–2203.
26. "Tor," https://www.torproject.org.
27. "Lantern," https://getlantern.org/.
28. "Jondo," https://anonymous-proxy-servers.net/en/jondo.html.
29. "I2p," https://iperf.fr/.
30. "iperf3," https://iperf.fr/.