Hindawi
Security and Communication Networks
Volume 2018, Article ID 5051270, 16 pages
https://doi.org/10.1155/2018/5051270

Research Article

Using Insider Swapping of Time Intervals to Perform Highly Invisible Network Flow Watermarking

Weiwei Liu[1], **Guangjie Liu**[1], **Yang Xia**[2], **Xiaopeng Ji**[1], **Jiangtao Zhai**[3] and **Yuewei Dai**[1,3]

[1] *School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *Wuhan Ship Communication Research Institute, Wuhan 430000, China*
[3] *School of Electrical and Computer Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China*

Correspondence should be addressed to Weiwei Liu; lwwnjust@njust.edu.cn

Received 19 August 2018; Revised 24 October 2018; Accepted 1 November 2018; Published 18 November 2018

Academic Editor: David Megias

Network flow watermarking (NFW) is an emerging flow correlation technique to deanonymize an anonymous communication system or detect stepping stones, in which a watermark is encoded into a network flow by manipulating some flow characteristics, predominantly by altering timing information. Although interval-based NFWs that employ time intervals as carrier have proven to be capable of resisting moderate network interference, they are vulnerable to some statistic-based attacks, which may expose the very existence of watermark and enable attackers to damage or remove watermark from the observed flow. In this study, using insider swapping of time intervals and an adaptive centroid quantization framework, we design a highly invisible NFW scheme, which is undetectable by multi-flow attacks (MFA), Kullback-Leibler divergence (KLD) test, Kolmogorov-Smirnov (K-S) test, and spread spectrum flow watermark (SSFW) detection. Experimental results using real traffic and public dataset show that the proposed NFW scheme can outperform three typical NFW schemes on invisibility while maintaining a strong interference-resistance capability of network jitter, packet loss, and dummy packet insertion.

1. Introduction

As a young branch of active traffic analysis, network flow watermarking (NFW) is a promising and urgent flow correlation technique, which has been employed for network monitoring or security enhancement of cloud computing [1], e.g., tracing network-based attacks, exposing the master in a botnet, and service dependency detection among network services in a complex platform like cloud platform. In recent years, the effectiveness for compromising specific anonymous communication systems has made NFW gain attention in cyberspace. To deanonymize an anonymous communication, the flow correlation on an anonymous node (e.g., onion router [2]) can be operated by actively manipulating the traffic characteristics of an incoming flow to create a particular pattern, which conceals the very existence of single or multiple watermark bits; the corresponding outgoing flow then can be recognized using the watermark decoding mechanism. Single-bit watermark can only be used to indicate the presence of the watermark, whereas multi-bit watermark can be employed to distinguish each watermarked flow when multiple paralleled flows and their sources need to be traced distinctly [3, 4].

According to the carrier type, NFW is usually classified into four categories: content [5], timing [6], size [7], and rate-based [8]. In these categories, however, timing-based watermarking [9, 10], in which the carrier signal is the arrival or departure time of the packets observed at a certain network node, has been mostly researched. In this study, we also focus on this issue since timing-based watermarking has been regarded as the most practical NFW. In fact, rate-based watermarking can also be viewed as a particular case of timing-based watermarking.

Similar to the information hiding technique in digital multimedia [11–13], robustness and invisibility are two main goals in the design of NFW. Here, robustness pertains to the ability of the NFW to operate despite the jitter inherent or maliciously added in the communication network,

especially in stepping stones or anonymous communication system where the flows may suffer from several types of channel interference, for example, network jitter, packet loss, dummy packet insertion, and packet padding. On the other hand, invisibility means that the adversary cannot detect the existence of the watermark by distinguishing between watermarked and legitimate traffic, using the statistical techniques as Kullback-Leibler divergence (KLD) test [14], Kolmogorov-Smirnov (K-S) test [15], multi-flow attacks (MFA) [16, 17], among others. Compared with network covert timing channels, another branch of timing channels that aims to transmit secret information in open overt communication channels, timing-based flow watermarking requires stronger robustness, as the embedded patterns always suffer from more serious channel interference in flow traceback. Robustness is the foundation for guaranteeing the success of timing-based flow watermarking. The challenge of designing a practical NFW scheme is to keep it invisible to hidden adversaries yet robust to network interference.

Enhancing robustness is one of the main reasons why researchers exploit the use of NFW as an alternative to passive traffic analysis [18]. The timing characteristics used in timing-based flow watermarking mainly contain inter-packet delay (IPD), interval packet counting, and interval centroid. In existing IPD-based flow watermarking schemes, the watermark is usually encoded by introducing small perturbations in IPDs between consecutive packets. Due to the vulnerability of IPDs to channel interference, especially to some desynchronizing attacks, mean balancing strategy or forward error correcting codes need to be employed for error reduction in IPD-based NFW. In so called interval packet counting based watermarking schemes [19], a flow is divided into intervals with fixed duration on the time axis. The number of packets in each interval is the carrier of the watermark; some specific patterns of interval packet counting are constructed to encode watermark bits by altering the statistical balance of the packet numbers in some selected intervals. The variation of interval packet counting under channel interference is significantly lower than that of IPD. Besides, interval centroid is exploited to be a robust timing carrier against packet loss and dummy packet insertion [20–22]; the spread spectrum (SS) technique can be used to spread the watermark signal or randomize the watermark locations. Interval centroid based watermarking manipulates the centroid of the packets within selected time intervals to encode a watermark. NFW based on interval packet counting and that based on interval centroid are collectively referred to as interval-based NFW.

Many state-of-the-art NFW schemes always trade off robustness at the expense of other performances such as capacity, which is not an important requirement for NFW in most application scenarios [1]. Thus, these schemes have been able to resist moderate network interference and active attacks that aim to make the watermark destroyed. Nevertheless, the invisibility is still regarded as the major obstacle to implement NFW [23]. Although some literatures have proposed watermarking algorithms that were claimed to be invisible, later works have shown that many of them can be easily identified by third parties [8, 24]. The key challenge to design an invisible NFW scheme is that the watermark encoding should introduce distortion as little as possible, yet meeting the requirements on robustness.

In this study, we propose a robust and highly invisible NFW scheme based on adaptive centroid quantization and insider swapping. Unlike other existing interval-based NFW schemes, the proposed NFW scheme has no requirement for pushing packets from one to the other in interval pairs, stretching or squeezing IPDs within some intervals. The cumulative distribution function (CDF) of the interval centroid is incorporated into a near-orthogonal sequence set to implement adaptive centroid quantization. The required centroid shift for each chosen interval is then realized using an insider swapping algorithm, which aims to retain statistical characteristics of IPDs while manipulating the interval centroid.

The following are the key contributions of this study:

(i) We propose an adaptive centroid quantization framework to design interval-based NFW schemes. The CDF of interval centroid is used to generate the quantization pattern, and a near-orthogonal sequence set is then used to determine the shift direction of the centroid. Compared with other existing interval-based NFW schemes, the proposed framework can reduce the required centroid shift effectively.

(ii) We propose an insider swapping algorithm which can manipulate the centroid of a time interval. Unlike existing strategies such as pushing packets from one to the other in interval pairs, stretching or squeezing IPDs within some intervals, the centroid shift is realized by swapping a limited number of IPD pairs within the interval. This algorithm can also be easily employed in other existing interval-based NFW schemes to further improve invisibility.

(iii) We present detailed comparative analysis of the proposed NFW schemes with other timing-based NFW schemes. The results have proven that the proposed scheme can evade detection of the MFA, KLD, and K-S tests, while maintaining inference-resistance capability. In particular, the proposed NFW scheme can preserve completely the same distribution of IPDs as the legitimate traffic.

The rest of the paper is organized as follows. In the next section, we present the background of NFW including its application on tracing in large-scale networks, adversary model, and related work on existing timing-based NFW. In Section 3, we give an overview of the proposed NFW scheme; the design of adaptive centroid quantization, insider swapping algorithm, and watermark decoding method are described, respectively. Experimental results and analysis are presented in Section 4. Finally, in Section 5, we provide a conclusion for this paper and discuss future work.

2. Background

In this section, we describe some background materials on NFW. First, beyond the applications introduced in prior work, we discuss the tracing in large-scale networks (e.g., state area network or metropolitan area network) based on NFW and big data processing technique. Next, we describe the adversary model of NFW and summarize related work on timing-based NFW.

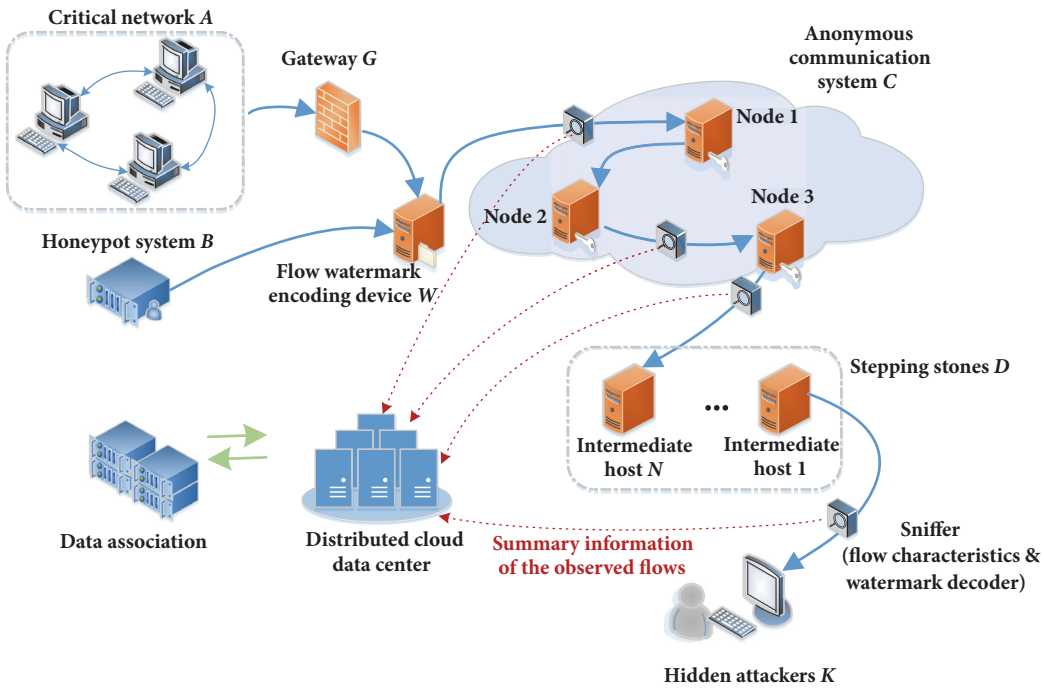


Figure 1: NFW-based tracing process in a large scale network.

2.1. Flow Tracing in Large-Scale Networks. Prior work on NFW mainly studies the design of the watermark encoding and decoding schemes, with a major focus on improving the robustness or invisibility. Nevertheless, there are less researches discussing the implementation of NFW in real world. In [25], BotMosaic, a novel countermeasure to botnets using an Internet Relay Chat (IRC) channel for command-and-control (C&C) channel, was proposed. It can perform network-based bot detection by inserting the watermark into the flows of all captured bot instances controlled by a watermarker and later detecting the specific pattern at a very low cost. In that paper, the deployment scenarios of BotMosaic were also discussed in detail; this scheme has proven to be promising for the detection of IRC-based botnets as it is much cheaper and easier to deploy than existing signature- and classification-based detectors. The most important contribution of BotMosaic is that it has actually extended NFW into a novel application scenario. Here, we discuss NFW-based flow tracing in real world, predominantly in a large-scale network. We consider the task of correlating malicious traffic or further finding out the hidden network attacker. NFW technique should be adopted along with big data processing technique to recover the attack path. The NFW-based tracing process in a large-scale network is depicted in Figure 1, which can also be simplified to apply in more basic scenarios.

Currently, under no circumstance would an experienced network attacker launch an offensive attack (e.g., denial-of-service attack, man-in-the-middle attack, and eavesdropping) to the victim using his or her own host directly. The victim is usually chosen from servers or computers that hold high value assets in a critical network. Most network-based attacks require an exchange of messages between the victim and the attacker. To the real identity or position being exposed, the hidden attacker usually creates a composite anonymous communication channel to the victim by designing a route that makes use of an available anonymous service on the Internet or stepping stones managed with botnet hosts. Watermarking has been found useful to trace the flow from the victim to the attacker.

Watermark encoder and watermark decoder are two main parts involved in NFW techniques. Usually, the watermark encoder is realized by a flow watermark encoding device W which is generally deployed at the gateway of a critical network A, or integrated into a Honeypot system B. All outgoing flows are injected with watermark bits to contain a particular pattern. It is evident that the interactive traffic between the victim and the attacker has no exception. The flows of the malicious communication usually pass through a public anonymous communication system C and a serial of stepping stones D that refer to the intermediate hosts. These intermediate hosts may threaten the existence of watermark due to several types of channel interference besides the ubiquitous network jitter.

The observation position in which there exists a passing-by flow containing the same pattern can be marked as the intermediate node or destination, which can help confirm the IP address of the final endpoint of the malicious communication. Eventually, the hidden attacker can be identified and part of or the entire attacking path can be detected. In this scenario, a number of sniffers need to be implemented at different levels of the large-scale network to collect traffic

characteristics including the watermark detection result. For each sniffer, the summary information of the observed flows will converge to a distributed cloud data center. The flow summary information needs five fundamental elements, including the arrival time of the first and last packets, protocol type, source IP, destination IP, and the watermark detection result. In the distributed data center, the summary information can be correlated using data association technique, which can help trace the source of the malicious attack in a large-scale network.

2.2. Adversary Model. In most application scenarios of NFW, the adversary is just the tracking object, who may attempt to apply costly countermeasures to remove the watermark from the incoming flows, or even interrupt communication when the tracking is perceived. For a passive adversary, the main goal is to distinguish between legitimate traffic and watermarked traffic while ensuring that the normal communication is not affected. We assume that the adversary has the full administrative authority over the legitimate and watermarked traffic, which implies that the statistical characteristics extracted from legitimate traffic are known to the adversary. On the other hand, an active adversary has higher privileges than the passive one. In order to remove the watermark, interference such as random packet delay and dummy packet insertion may be introduced into the network channel when the flows pass through the stepping stones when the adversary suspects the existence of a watermark. Nevertheless, these actions are usually restricted to a certain degree due to their affection on the normal communication.

In this study, we assume that the adversary has access to both legitimate and watermarked traffic, which corresponds to the Level I and Level II attack models described in [9]. In addition, we assume that some active attacking methods can be used by the adversary to endanger the existence of watermark at the stepping stones.

2.3. Related Work. The design goal of timing-based flow watermarking is to strengthen the robustness while introducing patterns as inconspicuous as possible. The earlier flow watermarking schemes are based on IPDs. In [26], the mean of randomly-chosen IPDs is modulated to embed a single watermark bit by using the quantization index modulation (QIM) framework; this scheme is robust against network jitter to some extent, whereas it cannot retain synchronization when some packets are dropped or split. In addition, the invisibility is not satisfactory as the watermark parameter can be inferred by intelligent attackers that observe IPDs of the flows passing through adjacent stepping stones, which may result in watermark removal. To address these problems, a nonblind IPD-based NFW scheme called RAINBOW was proposed in [10, 27], which can achieve better invisibility than the previous blind ones by inserting small delays. The repeat-accumulate codes are then utilized as forward error correction for further improvement on robustness [28]. Next, an IPD-based flow watermarking scheme that can resist dependent substitution, deletion, and bursty insertion errors was proposed in [9]. In this scheme, a Hidden-Markov Model (HMM) decoding scheme is incorporated into QIM to realize watermark encoding and decoding. An alternative scheme using linear error-correcting codes and Varshamov-Tenengol'ts (VT) codes was then proposed to further reduce the complexity of watermarking decoding [29].

To improve the robustness against desynchronization attacks such as packet count changing, dummy packet insertion, and repacketization, interval-based watermarking (IBW) scheme was proposed in [19], which modifies packet counts of randomly-chosen interval pairs, specifically, pushing all packets of one interval to its adjacent interval in the direction with respect to the watermark bit. However, IBW is vulnerable to multi-flow attack (MFA). Besides interval-packet counting, another important carrier in interval-based NFW is interval centroid, which has a stronger stability in the presence of flow mixing, flow splitting, and flow merging. An interval centroid based watermarking (ICBW) scheme was proposed in [22], which embeds a watermark bit by adjusting the timing offset of the packets in an interval pair. An interval centroid based spread spectrum watermarking scheme (ICBSSW) for tracing multiple flows was then proposed by combining the ICBW scheme with the spread spectrum (SS) technique [21], which can be viewed as an improvement of ICBW. However, the patterns of these interval-based NFW schemes can be found out by the MFA test. To resist MFA, SWIRL was proposed by selecting mark intervals and determining the packet transferring pattern based on the characteristics of the flow [30]. Although the invisibility of SWRIL has been better than its previous interval-based watermarking schemes, it still can be easily detected by a chosen flow attacker [31] or BACKLIT [24].

Thus, invisibility plays a significant role in the design of NFW under the circumstance that some state-of-the-art NFW schemes have a strong capability of resisting network interference, while they are vulnerable to some statistic-based attacks. In this study, we design a robust and highly-invisible NFW scheme based on adaptive centroid quantization and insider swapping, which can preserve the very similar statistical characteristic as the legitimate traffic.

3. The Proposed Scheme

Similar to the existing interval-based NFW schemes, we assume that the following parameters are shared between the watermark encoder and decoder in advance: the time offset o (In this study, we set $o = 0$, namely, the sending or arrival time of the first packet in flow f) and the interval length $T > 0$, a secret key K that is used to randomly choose intervals. The framework of the proposed NFW scheme can be presented in Figure 2.

We first employ the spread spectrum method in [32] to construct a near-orthogonal sequence set that consists of S temporally white sequences. There exists one-to-one mapping relationship between the temporally white sequences and watermarking information containing at most $\log_2 S$ bits. This near-orthogonal sequence set can be used for generating the centroid quantization pattern, and the watermark information is used to determine the adopted sequence for generating quantization pattern.

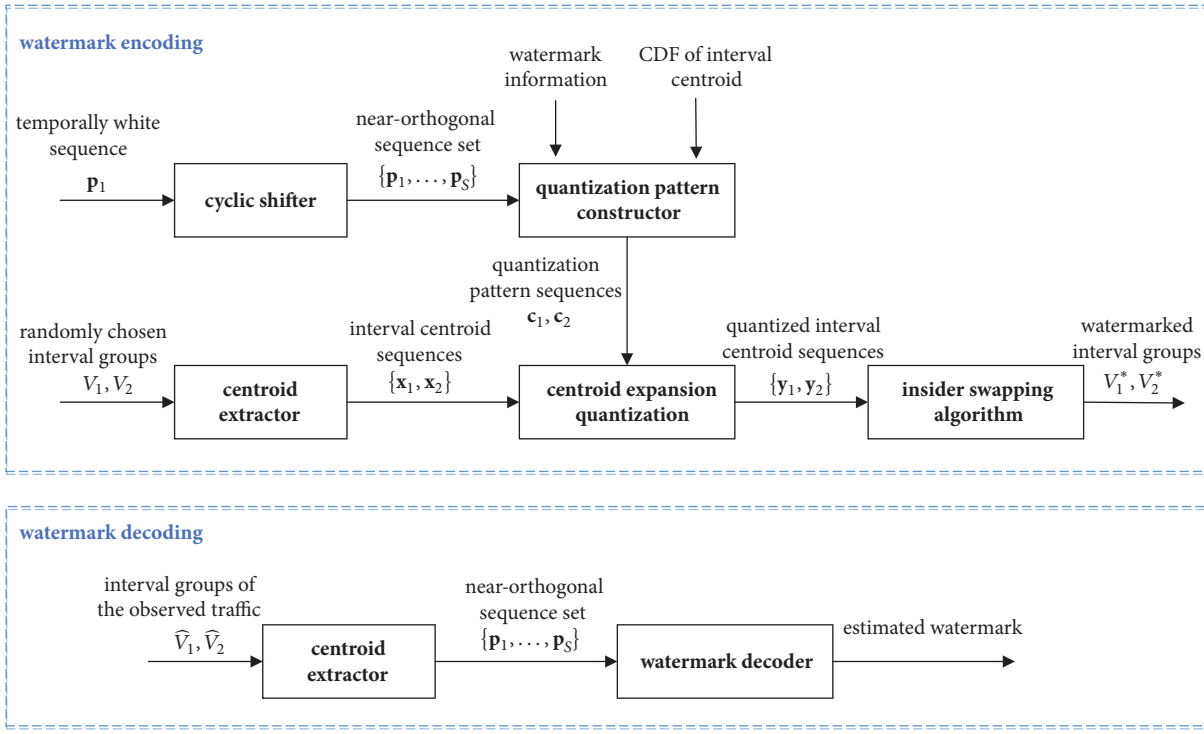


Figure 2: The framework of the proposed NFW scheme.

For a given flow f, its duration can be divided into n intervals $\{I_1, \ldots, I_n\}$, where I_1 starts from the time offset o. The cumulative distribution function (CDF) of the interval centroid is acquired from IPDs of the captured legitimate traffic. Then, the quantization pattern sequences $\mathbf{c}_1$ and $\mathbf{c}_2$ can be generated from the quantization pattern constructor with the selected temporally white sequence and the CDF of the interval centroid.

With the quantization pattern, a pair of interval groups are randomly chosen from the interval set with the shared secret key K. They are denoted as $V_1 = \{I_{a_1}, \ldots, I_{a_N}\}$ and $V_2 = \{I_{b_1}, \ldots, I_{b_N}\}$, respectively. Each of them contains N randomly chosen intervals, $\{a_1, \ldots, a_N\}, \{b_1, \ldots, b_N\} \subset \{1, \ldots, n\}$ and $\{a_1, \ldots, a_N\} \cap \{b_1, \ldots, b_N\} = \varnothing$. Their corresponding centroid sequences $\mathbf{x}_1$ and $\mathbf{x}_2$ are extracted to be the carrier for quantization. Then, the quantization pattern constructor is used to quantize the carrier sequences into the quantized interval centroid sequences $\mathbf{y}_1$ and $\mathbf{y}_2$. To minimize the statistical distortion, the insider swapping algorithm is used to manipulate a limited number of packets in each interval to make the resulting interval centroid approximate to the quantized one. At the watermark decoding side, the noisy versions of the watermarked interval groups, V_1^* and V_2^*, are exploited to verify the existence of the watermark. As shown in Figure 2, the proposed frame mainly consists of adaptive quantization (quantization pattern constructor and centroid expansion quantization), insider swapping algorithm, and the decoder. These are described in detail in the following subsections.

3.1. Adaptive Quantization of Interval Centroid. Let $\mathbf{p}_1 \in \{-1, +1\}^N$ be a temporally white sequence of length N (In this study, we generate it with M-sequence). Then, the near-orthogonal sequence set $\{\mathbf{p}_1, \ldots, \mathbf{p}_s\}$ can be generated with a cyclic shifter.

$$\mathbf{p}_i = \mathbf{p}_1 \gg (i-1) \tag{1}$$

where $i = 1, \ldots, S$, "$\gg$" denotes the cyclic right shift operation; it holds that $(\mathbf{p}_i \cdot \mathbf{p}_j^T)/(\mathbf{p}_i \cdot \mathbf{p}_i^T) \approx 0$ when $i \neq j$, $\mathbf{p}_i^T$ denotes the transposition of $\mathbf{p}_i$.

Let N_i be the packet number of the interval I_i; the arrival time of each packet is denoted as $t_{i,j}$, $i = 1, \ldots, n$ and $j = 1, \ldots, N_i$. The corresponding interval centroid can be obtained with Equation (2).

$$C(I_i) = \frac{\sum_{j=1}^{N_i} (t_{i,j} - o_i)}{N_i} \tag{2}$$

where o_i denotes the starting time of the interval.

Denote F_T as the empirical CDF of the interval centroid when the interval length is set to T, the maximum and minimum centroid values are $c_{\max}$ and $c_{\min}$, namely, $F_T(c_{\max}) = 1$ and $F_T(c_{\min}) = 0$. The watermark bits are converted into S-ary symbols so that they can be represented by the sequence index in the near-orthogonal sequence set. Without loss of generality, we consider the condition that the watermark is a single S-ary symbol, the scheme with multi-symbol watermark can be easily extended from it.

Assume that the temporally white sequence corresponding to the watermark symbol is $\mathbf{p}_w = \{p_1, \ldots, p_N\}$. The quantization pattern sequences $\mathbf{c}_1 = \{c_{a_1}, \ldots, c_{a_N}\}$ and $\mathbf{c}_2 = \{c_{b_1}, \ldots, c_{b_N}\}$ can be obtained with the quantization pattern constructor. The high-low-level values in the determined sequence $\mathbf{p}_w$ are mapped into the dynamic values in the CDF of the interval centroid. Here, "1" is mapped into the value in the first half part of the CDF curve, whereas "-1" is mapped into the value in the latter part of the CDF curve.

$$\begin{aligned} c_{a_j} &= F_T^{-1}\left[0.5 + p_j \cdot \left(\delta_{a_j} + \gamma\right)\right] \\ c_{b_j} &= F_T^{-1}\left[0.5 - p_j \cdot \left(\delta_{b_j} + \gamma\right)\right] \end{aligned} \tag{3}$$

where F_T^{-1} denotes the inverse of the function F_T, $\gamma \in (0, 0.5)$ denotes the constant guard parameter, and $\delta_i \geq 0$ denotes the randomly generated dither, where $i \in \{a_1, \ldots, a_N, b_1, \ldots, b_N\}$. The randomly generated dither can be used to relieve the quantization regularity. In this study, all dithers are set to zero as the compensative quantization strategy is employed in the following centroid expansion quantization, which can also relieve the quantization regularity.

With the quantization pattern sequences $\mathbf{c}_1$ and $\mathbf{c}_2$, centroid expansion quantization is employed, which encodes the watermark by manipulate the centroid difference between the two interval groups. Let $\mathbf{x}_1 = \{x_{a_1}, \ldots, x_{a_N}\}$ and $\mathbf{x}_2 = \{x_{b_1}, \ldots, x_{b_N}\}$ be the interval centroid sequences corresponding to the interval group pair V_1, V_2; the quantized centroid sequence pair $\mathbf{y}_1 = \{y_{a_1}, \ldots, y_{a_N}\}$ and $\mathbf{y}_2 = \{y_{b_1}, \ldots, y_{b_N}\}$ can be obtained using Equation (4).

$$\begin{aligned} \mathbf{y}_1 &= \alpha \mathbf{c}_1 + (1 - \alpha)\, \mathbf{x}_1 \\ \mathbf{y}_2 &= \alpha \mathbf{c}_2 + (1 - \alpha)\, \mathbf{x}_2 \end{aligned} \tag{4}$$

where $\alpha \in (0, 1]$ denotes the quantization ratio, and $1 - \alpha$ can be interpreted to be the compensative parameter.

3.2. Insider Swapping Algorithm. Most existing passive attacks for flow watermarking attempt to find out the existence of a watermark by comparing the timing characteristics of the observed traffic to that of the legitimate one. The timing characteristics are always extracted from statistical modeling on the inter-packet delays (IPDs), which are the most fine-grained representation for timing information of network traffic. Thus, to minimize the statistical distortion of IPDs, whereas making the centroid of the interval groups approximate to the quantized ones, the insider swapping algorithm is proposed to swap a very limited number of IPDs in each watermarked interval. As the centroid shift is operated by several IPD swaps in each interval, the first-order statistics represented by KL divergence and K-S test can be reserved without any distortion, and the multi-order statistical distortion is also negligible enough to ensure the watermarked flows indistinguishable from the legitimate flows.

With the quantization centroid sequences $\mathbf{y}_1$ and $\mathbf{y}_2$, we manipulate the arrival time of the packets to make the centroid of each manipulated interval approximate to the quantized one. Let $\{t_1, \ldots, t_{N_I}\}$ be the packet timing sequences of a given interval I. N_I denotes the number of packets in the interval. Define

$$d_j = \begin{cases} t_j - t_{j-1} & j > 1 \\ t_j - o_I & j = 1 \end{cases} \tag{5}$$

where o_I denotes the starting time of the interval, d_j denotes the j-th IPD in the interval, and specifically, d_1 denotes the first IPD that is split into broken one due to the interval partition. Thus, the interval centroid in Equation (2) can be rewritten as follows.

$$C(I) = \frac{\sum_{j=1}^{N_I} (N_I - j + 1)\, d_j}{N_I} \tag{6}$$

In the existing interval centroid based NFW schemes, the centroid shift will result in some extremely large or small IPDs, or significant regularity of centroid shift pattern. Here, we use an IPD swap strategy to address these problems, which can help maintain high invisibility. In addition, the required buffer length to implement the proposed scheme is very small, as the swaps only occur among the IPDs in the same interval. The proposed insider swapping algorithm is described in Figure 3.

Because of the independence of IPD swaps in each interval, we describe the insider swapping algorithm with a single watermarked interval I, which contains N_I packets. We denote the original and target interval centroid as C_o and C_w, respectively. The number of IPD swaps is limited to L_e; namely, we swap two IPDs in an interval at most L_e times. L_e is usually a small integer which is relevant to the packet number of the interval.

Without loss of generality, let i and j be two indices satisfying $1 < i < j \leq N_I$. If we swap the ith IPD with the j-th IPD, the resulting centroid shift will be

$$\Delta C_{i,j} = \frac{(j - i) \cdot \left(d_j - d_i\right)}{N_I} \tag{7}$$

Next, we determine a swap pair in each iteration. According to Equation (6), if there exist two swap pairs $\{i_1, j_1\}$ and $\{i_2, j_2\}$ that satisfy $\{i_1, j_1\} \cap \{i_2, j_2\} \neq \varnothing$, e.g., $i_1 = i_2$, the centroid shift result from the two swaps would be equivalent to that result from the swap $\{j_1, j_2\}$. Thus, to guarantee the efficiency of the swaps, we assume that each IPD in the interval can be swapped at most one time. In each iteration, a centroid shift matrix is first constructed to represent the resulting centroid shifts for all available IPD pairs; $\mathbf{\Delta C}_k$ denotes the centroid shift matrix in the k-th iteration. In the first iteration, the centroid shift matrix $\mathbf{\Delta C}_1$ is initialized with $\mathbf{\Delta C}_1(i, j) = \Delta C_{i,j}$, where $\mathbf{\Delta C}_1(i, j)$ denotes the j-th element in the ith row of the matrix. As shown in Figure 3, it is apparent that matrix $\mathbf{\Delta C}_1$ is a symmetric matrix with all-zero diagonal elements. Then, all elements in the upper triangular parts are reshaped and sorted to find the best element, which is the most similar one to the target centroid shift ΔC_k^{tar}. ΔC_1^{tar}

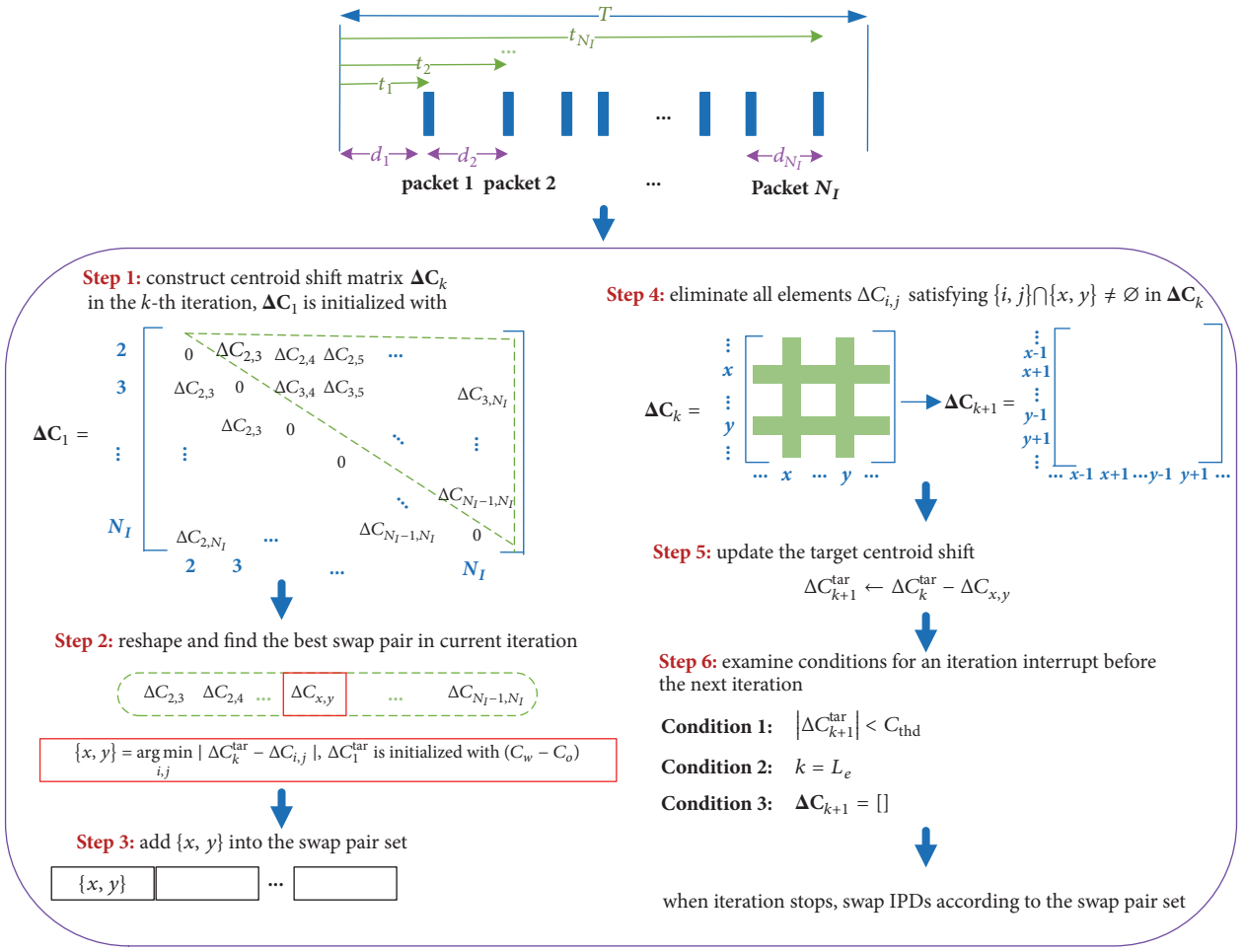


Figure 3: The process of the insider swapping algorithm.

is initialized with $C_w - C_o$. The best swap pair in the k-th iteration can be obtained as follows.

$$\{x, y\} = \arg\min_{i,j} \left|\Delta C_k^{\text{tar}} - \Delta C_{i,j}\right| \tag{8}$$

The selected pair $\{x, y\}$ is then added to the swap pair set Ω, which is initialized with $\varnothing$. To start the next iteration, we need to update the centroid shift matrix and the target centroid shift. As each IPD in the interval can be swapped at most one time, we eliminate all elements $\Delta C_{i,j}$ satisfying $\{i, j\} \cap \{x, y\} \neq \varnothing$ from $\Delta\mathbf{C}_k$ to obtain the centroid shift matrix in the next iteration, i.e.,

$$\begin{aligned} \Delta\mathbf{C}_{k+1} &= \Delta\mathbf{C}_k \vee \Upsilon(x, y) \\ \Upsilon(x, y) &= \left\{\Delta C_{i,j} \mid \{i, j\} \cap \{x, y\} \neq \varnothing\right\} \end{aligned} \tag{9}$$

where "$\mathbf{A} \vee U$" denotes elimination of all elements belonging to the set U from the matrix $\mathbf{A}$.

The target centroid shift for the next iteration can be updated using the difference between the current target centroid shift and the actual centroid shift.

$$\Delta C_{k+1}^{\text{tar}} = \Delta C_k^{\text{tar}} - \Delta C_{x,y} \tag{10}$$

There exist the following three conditions for an iteration interrupt before the next iteration.

(i) The absolute value of the target centroid shift for the next iteration is lower than the threshold value, namely, $|\Delta C_{k+1}^{\text{tar}}| \leq C_{\text{thd}}$. In this study, the threshold value C_{thd} is set to ηT, where $\eta \in (0, 0.1)$ denotes the scale parameter. Too large value of η will result in obvious performance degradation on robustness due to the gap between the actual centroid shift and the target one, whereas too small value of η will increase the number of swap pairs.

(ii) The current iteration number k has achieved the maximum number L_e. The maximum number of IPD swaps is relevant to the number of IPDs in each interval, as the numbers of IPDs in different intervals are diverse. In this study, L_e is set to $\min(10, N_I/2)$.

Table 1: Timing statistical characteristics of the used traffic.

Traffic Type	Maximum of IPD	Minimum of IPD	Mean of IPD	Standard Deviation of IPD
Traffic A: CAIDA 2016	249.7ms	0.001ms	15.2ms	35.7ms
Traffic B: Google Chrome Remote Desktop	57.9ms	0.009ms	7.3ms	13.8ms

(iii) The centroid shift matrix for the next iteration is a null matrix.

The iteration continues until one of the above three conditions is tenable. When iteration stops, we swap IPDs in the swap pair set to make the centroid approximate to the target one.

3.3. Watermark Decoding. To decode the watermark from an observed flow, we first calculate the centroid sequence pair using the shared secret key, the correct decoding offset, and the interval length. Similar to the other existing interval-based NFW schemes, the decoder has the capability of deriving the exact random interval groups used for encoding the watermark. Thus, we can use the self-synchronization strategy discussed in [22] to find the correct decoding offset. Let $\mathbf{y}_1^* = \{y_{a_1}^*, \ldots, y_{a_N}^*\}$ and $\mathbf{y}_2^* = \{y_{b_1}^*, \ldots, y_{b_N}^*\}$ be the derived centroid sequence pair, which are the noisy versions of $\mathbf{y}_1$ and $\mathbf{y}_2$, respectively.

Next, we define the difference between the derived centroid sequence pair as

$$\begin{aligned}\Delta\mathbf{y}^* &= \mathbf{y}_1^* - \mathbf{y}_2^* = \mathbf{y}_1 - \mathbf{y}_2 + \boldsymbol{\delta} \\ &= \alpha\left(\mathbf{c}_1 - \mathbf{c}_2\right) + (1-\alpha)\left(\mathbf{x}_1 - \mathbf{x}_2\right) + \boldsymbol{\delta}\end{aligned} \tag{11}$$

where $\boldsymbol{\delta} \in \mathbb{R}^N$ denotes the difference variation vector due to the channel interference, the power of which is determined by the signal-noise ratio (SNR).

With Equation (3), we have

$$\begin{aligned}\Delta\mathbf{y}^* &= \alpha\mathbf{p}_a \cdot \left[F_T^{-1}\left(0.5+\gamma\right) - F_T^{-1}\left(0.5-\gamma\right)\right] \\ &\quad + (1-\alpha)\left(\mathbf{x}_1 - \mathbf{x}_2\right) + \boldsymbol{\delta}\end{aligned} \tag{12}$$

As the inequality $N \cdot [F_T^{-1}(0.5+\gamma) - F_T^{-1}(0.5-\gamma)] > (\mathbf{x}_1 - \mathbf{x}_2) \cdot \mathbf{p}_a^T$ can hold with a very large probability that approximates to 1 when the guard parameter γ is of a suitably large value, we can obtain the following result when both sides in Equation (12) are multiplied by the transposition of the adopted temporally white sequence.

$$\begin{aligned}\Delta\mathbf{y}^* \cdot \mathbf{p}_w^T &= \alpha \cdot N \cdot \left[F_T^{-1}\left(0.5+\gamma\right) - F_T^{-1}\left(0.5-\gamma\right)\right] \\ &\quad + (1-\alpha)\left(\mathbf{x}_1 - \mathbf{x}_2\right)\mathbf{p}_w^T + \boldsymbol{\delta}\mathbf{p}_w^T \\ &> \left(\mathbf{x}_1 - \mathbf{x}_2\right)\mathbf{p}_w^T + \boldsymbol{\delta}\mathbf{p}_w^T\end{aligned} \tag{13}$$

With Equation (13), the watermarked flow can be distinguished from the legitimate flows using the shared near-orthogonal sequence set at the watermark decoding side. Furthermore, the watermark symbol can be extracted from the watermarked flow according to Equation (14), as the orthogonality holds when $\mathbf{p}_w \neq \mathbf{p}_k$, namely, $(\mathbf{p}_w \cdot \mathbf{p}_k^T)/(\mathbf{p}_w \cdot \mathbf{p}_w^T) \approx 0$.

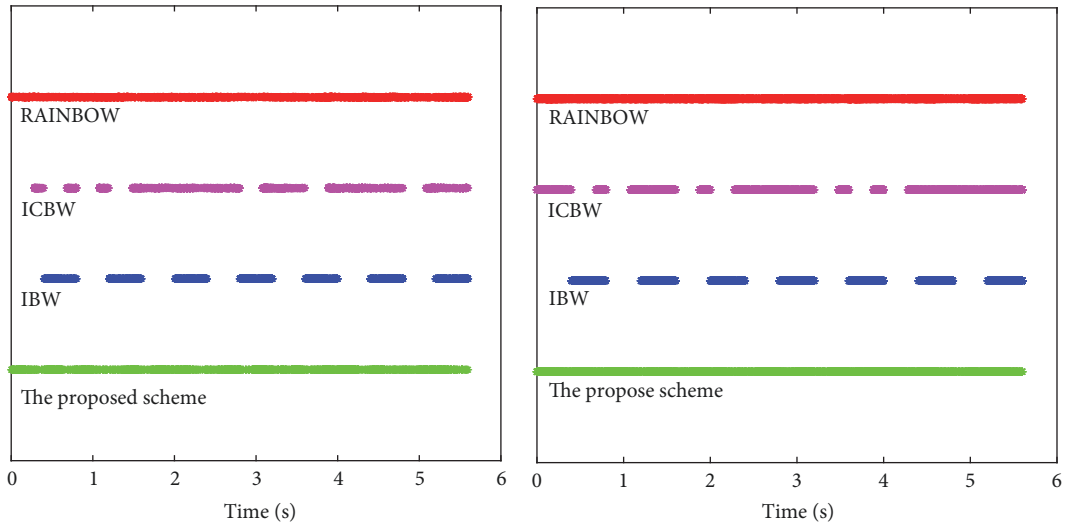
$$\Delta\mathbf{y}^* \cdot \mathbf{p}_a^T > \Delta\mathbf{y}^* \cdot \mathbf{p}_k^T \tag{14}$$

Thus, for $i = 1, \ldots, S$, if $\max(\Delta\mathbf{y} \cdot \mathbf{p}_i^T) > T_D$, we can determine that there exists a watermark in the observed flow. Otherwise, it will be judged to be a legitimate flow. Here, T_D denotes the threshold with respect to false alarm rate, which needs to be determined according to the CDF of the interval centroid, the quantization ratio, and the guard parameter. Furthermore, based on the one-to-one mapping relationship, the watermarking symbol can be extracted by finding the temporally white sequence $\mathbf{p}_i$ that can maximize $\Delta\mathbf{y}^* \cdot \mathbf{p}_i^T$.

4. Experimental Results and Analysis

In this section, we benchmark the proposed NFW scheme by examining the invisibility and robustness. We compare it with three typical schemes including ICBW [22], IBW [19], and RAINBOW [10]. Among interval-based watermarking schemes, ICBW and IBW are quite popular ones. In ICBW, randomly-chosen time intervals are classified into two different subsets, and the centroid of the chosen interval pairs is manipulated to encode a watermark. In IBW, the packet counts of randomly-chosen interval pairs are modified to encode the watermark by pushing all packets of one interval to the corresponding adjacent interval. RAINBOW is the first nonblind approach for IPD-based flow watermarking, which encodes the watermark bits by delaying some packets; the timing of incoming flows is recorded and compared with the timing of outgoing flows to decode the watermark.

4.1. Experimental Setup. To analyze the performance of the NFW schemes, we use two types of network traffic including the real traffic generated by typical application and that extracted from public dataset. The first type of traffic is the SSH flows extracted from the CAIDA network traces gathered in 2016 [33]. The length of each extracted SSH flow is larger than 2000. These SSH traces can represent typical human behavior in interactive network connection. This type of network traffic is referred to as *Traffic A*. The other one is the traffic generated by *Google Chrome Remote Desktop*, the samples of which are the same as those used in our prior work [34]. The destination was implemented in a host at the Computer Science Department, University of California, Davis (UCDavis), which was connected to the Internet using wired Ethernet. The source was a laptop that was connected to the Internet via public WiFi in the UCDavis campus. For this case, the end-to-end connection was over multiple hops. This type of network traffic was referred to as *Traffic B*. The statistical characteristics of the two types of network traffic are shown in Table 1.



(a) Traffic A: CAIDA (b) Traffic B: Google Chrome Remote Desktop

Figure 4: MFA test for different schemes. (a) *Traffic A*, (b) *Traffic B*.

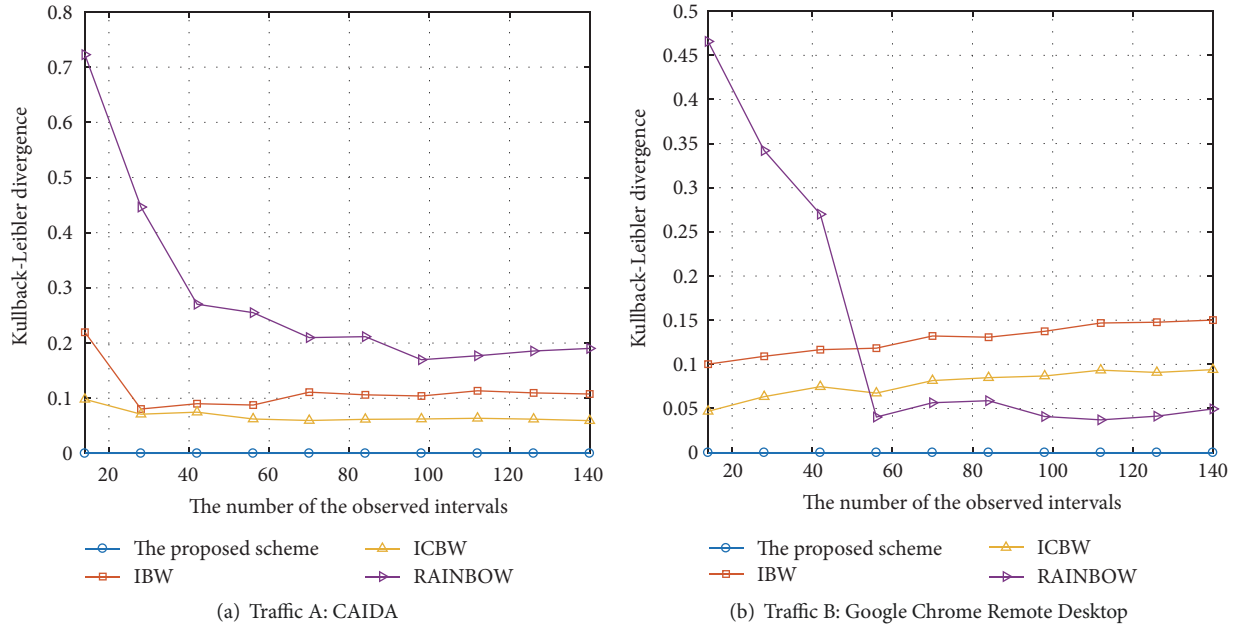
To evaluate the invisibility, we employ three statistical tests including MFA test, KLD test, and K-S test. MFA test is mainly used to attack the interval-based NFW, in which the watermark is always encoded within the same intervals in the flows. Therefore, an attacker that observes multiple watermarked flows can align them to render the watermarks visible. KLD test and K-S test are used to measure the difference between the distribution of IPDs of the watermarked flow and that of the legitimate flow, which are effective approaches to find the very existence of watermark as most NFW schemes result in a significant variation of the distribution of IPDs. To test the robustness, we first consider the watermark detection accuracy in the presence of network jitter. Then, packet loss and dummy packet insertion are also introduced into the performance evaluation. It is worth noting that RAINBOW is a nonblind NFW scheme that needs original flows for watermark decoding, which is actually unfair to compare it with blind NFW schemes on robustness. Thus, we only use it in invisibility evaluation.

4.2. Invisibility. To evaluate the invisibility of the four NFW schemes, we first employ MFA test on the watermarked flows generated by different schemes. As the MFA test with multi-message watermarks can be extended from the MFA test with single-message watermarks, we only present the results with single-message watermarks in Figure 4. MFA test relies on collecting a series of flows that are watermarked with the same message. These flows are combined into a single flow and examined for large gaps between packets. In Figure 4, each horizontal line shows the arrival time of the packets for 10 combined flows after the corresponding watermark has been applied. The horizontal axis denotes the duration. For ICBW, IBW, and the proposed scheme, the interval length is set to $T = 400ms$. The length of the temporally white sequence in the proposed scheme is set to $N = 7$. To compare the four schemes fairly, the numbers of the observed intervals in each flow for the four NFW schemes are all set to 14. The same parameter setting is employed in the following experiments unless otherwise specified.

From Figure 4 we can find that there exists no visible gap without packet arrivals for RAINBOW and the proposed scheme, which is consistent with the theoretical analysis for legitimate traffic. Nevertheless, the watermark patterns are clearly visible in the combined flows for ICBW and IBW, which can reveal the presence of a watermark. In addition, the watermark pattern for IBW exposed by MFA test is more regular than that for ICBW. Based on the combined flows, it is easy to recover the watermark parameter of ICBW and IBW, which means the following watermark removal attempts launched by watermark attackers. The results in Figure 4 prove that the proposed scheme is effective to resist MFA test unlike the other two interval-based NFW schemes.

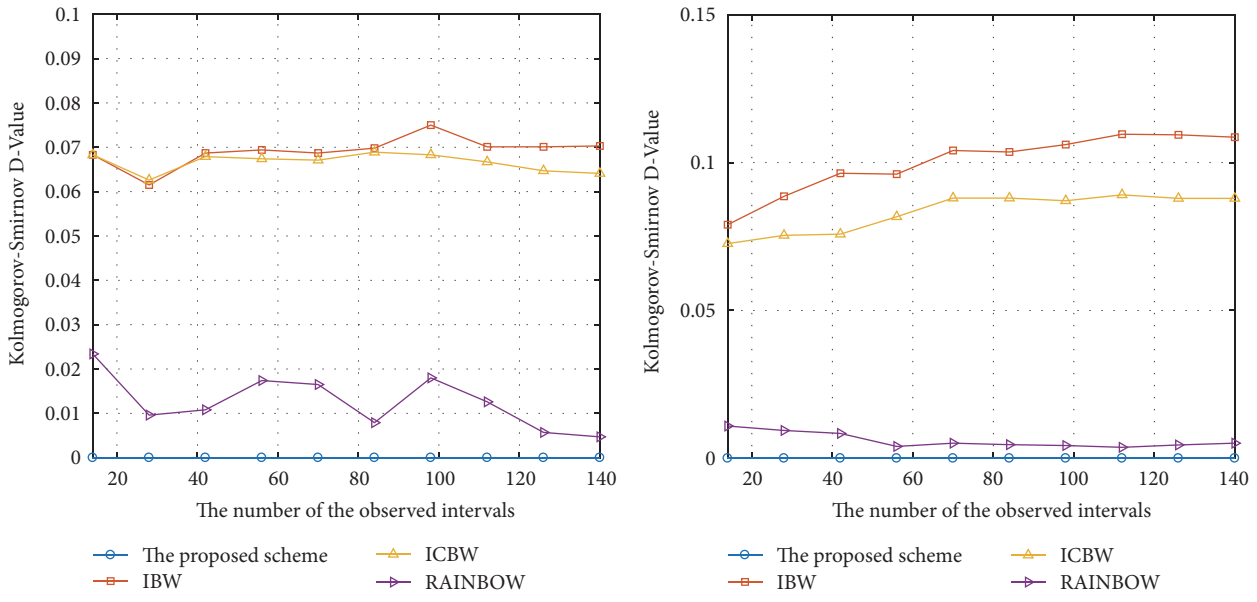
To measure the gap between the distribution of IPDs of the watermarked flow and that of the legitimate flow, we also employ KLD test and K-S test in Figures 5 and 6, respectively. Each point in the figures is obtained using an average of 10 samples. The horizontal axis denotes the number of the observed intervals and the vertical axis denotes the detection results.

From Figure 5 we see that the results of statistical tests tend to be stabilized gradually with the increasing number of intervals. ICBW and IBW have similar performance on KLD test; their KLD is both about 0.1. The KLD of RAINBOW is the largest among all schemes for *Traffic A*, whereas it is smaller than that of ICBW and IBW for *Traffic B*. For *Traffic A*, the results in Figure 5(a) show that the KLD of RAINBOW is about 0.2 when the number of the observed intervals is larger than 100. In Figure 5(b), the KLD of RAINBOW is only about 0.05 when the number of the observed intervals is larger than 60. For the proposed scheme, the KLD keeps



(a) Traffic A: CAIDA (b) Traffic B: Google Chrome Remote Desktop

Figure 5: KLD test for different schemes. (a) *Traffic A*, (b) *Traffic B*.

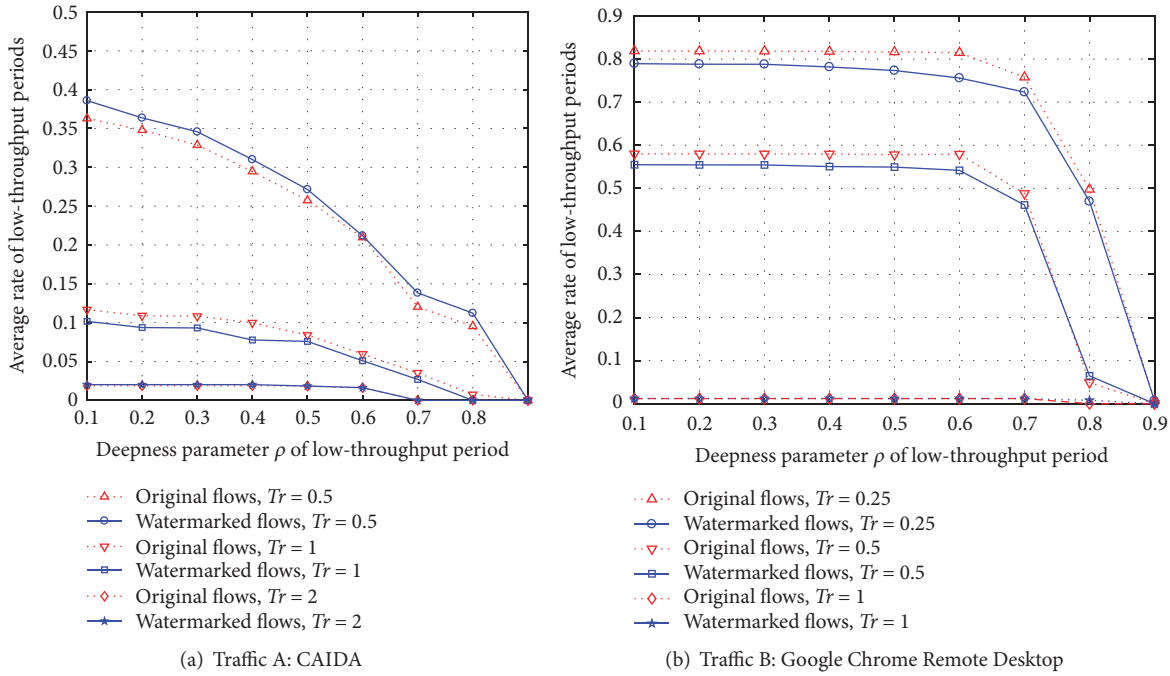


(a) Traffic A: CAIDA (b) Traffic B: Google Chrome Remote Desktop

Figure 6: K-S test for different schemes. (a) *Traffic A*, (b) *Traffic B*.

zero for different traffic type and interval number. It shows significant superiority on KLD test when compared with the other three schemes.

Figure 6 shows the K-S test results for the four schemes. The K-S D-values of RAINBOW and the proposed scheme are much smaller than those of ICBW and IBW. For *Traffic A*, the K-S D-values of ICBW and IBW are both about 0.07. And for *Traffic B*, their K-S D-values are both about 0.1. RAINBOW can achieve very small K-S D-value for both types of traffic. As shown in Figure 6, the K-S D-values of RAINBOW for *Traffic A* and *Traffic B* each are about 0.01. Similar to the performance on KLD test, the K-S D-value of the proposed scheme keeps zero for different traffic and interval numbers. The results in Figures 5 and 6 show that KLD test and K-S test are effective in detecting ICBW and IBW, whereas the two tests are completely ineffective in detecting the proposed scheme. As an actual fact, its capability against KLD test and K-S test is the result that the distribution of IPDs remains unchanged in the centroid shift stage.



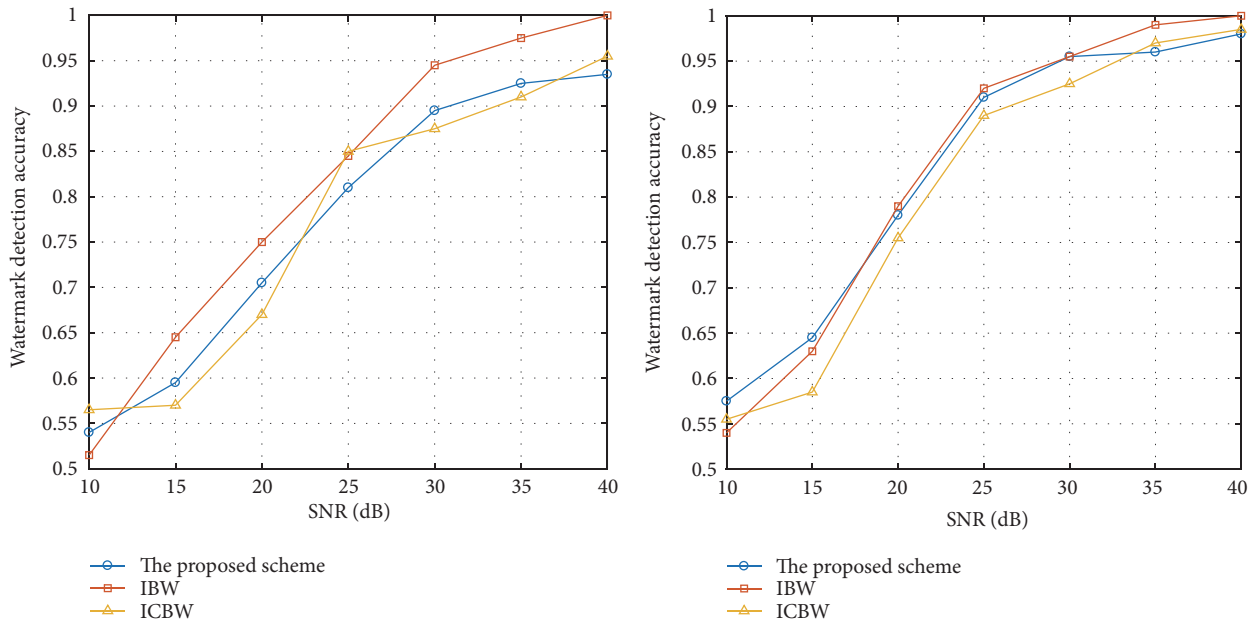
(a) Traffic A: CAIDA

(b) Traffic B: Google Chrome Remote Desktop

FIGURE 7: The average rate of low-throughput periods for the original and watermarked flows with different parameters. (a) *Traffic A*, (b) *Traffic B*.

We employ spread spectrum method to construct near-orthogonal sequences, which are then used for determining the shift direction of the centroid, to verify whether using these temporally white sequences would introduce additional risks of exposure to the very existence of the watermarks. We also use the detection scheme in [35] to evaluate the invisibility of the proposed scheme. In [35], a novel approach to detect spread-spectrum flow watermarks (SSFW) by leveraging their intrinsic features was proposed. The detection scheme works based on the fact that SSFW causes alternate low throughput and high throughput periods in a watermarked flow; using Pseudo-Noise (PN) codes increases the number of low-throughput periods. A significant advantage of the detection scheme is that it only needs to investigate individual flows to identify the embedded watermarks. For each type of traffic, we use 100 flows for watermark embedding, and the duration of each flow is longer than 40 seconds. To maximize the difference between the original and watermarked flows, we embed the same watermark bit repeatedly in a flow; namely, each flow is fully watermarked except the last few intervals. The interval length and the length of the temporally white sequence are still set to 400 ms and 7, respectively. In the SSFW detection scheme, a sequence of throughput samples are first constructed from each observed flow. For TCP flows, the basic time unit is suggested to be the round-trip time (RTT). For UDP flows, the basic time unit could be set to the average IPD. As *Traffic A* is the SSH flows while *Traffic B* is the UDP flows, we count the packet bytes within each RTT duration to form the throughput samples for *Traffic A* and count the packet bytes within each average IPD duration to form the throughput samples for *Traffic B*. As the main criterion of the SSFW detection scheme is the abnormal increment in the rate of low throughput periods, we compare the average rate of low throughput periods for the original and watermarked flows. In the SSFW detection scheme, there are two parameters that affect the number of low throughput periods, whose abnormal behavior is the basis of the detection system. The two parameters are the run parameter T_r and the deepness parameter ρ, T_r determines the minimal duration of a low throughput period, and ρ is related to the deepness of a throughput period. The appropriate parameters in the detection scheme for different types of traffic are varying. When T_r is too large, the rate of low throughput periods will remain close to zero for most traffic, whereas many short low throughput periods will be included in both the original and watermarked flows when using a too small T_r. Thus, after testing with different values in the suggested range, we let T_r to be 0.5, 1, 2 for *Traffic A*, and 0.25, 0.5, 1 for *Traffic B* in our experiments. The deepness parameter ρ is set to the value from 0.1 to 0.9 with step 0.1. The average rate of low throughput periods for the original and watermarked flows with different parameters is shown in Figure 7. The horizontal axis denotes the value of the deepness parameter and the vertical axis denotes the average rate of low throughput periods. Each point in Figure 7 is obtained using the average of 100 samples.

From Figure 7(a) we can find that the rates of low throughput periods for the watermarked flows are very close to those for original flows. The rate decreases when the run parameter T_r and the deepness parameter ρ increase. For *Traffic A*, if we let $T_r = 0.5$, the rate will decrease from about 0.4 to 0 when the deepness parameter raises from 0.1 to 0.9. However, when $T_r = 2$, the rate remains below 0.05 with $\rho \leq 0.6$ and decreases to 0 with $\rho \geq 0.7$. For *Traffic B*, the



(a) Traffic A: CAIDA

(b) Traffic B: Google Chrome Remote Desktop

Figure 8: Watermark detection performances for different schemes under varying network jitter. (a) *Traffic A*, (b) *Traffic B*.

results in Figure 7(b) show that the rates of low throughput periods for the original and watermarked flows are still very similar. When $T_r = 0.25$, the rates of low throughput periods for both the original and watermarked flows are about 0.8 with $\rho \leq 0.6$ and decrease rapidly with $\rho \geq 0.7$. When $T_r = 1$, the rates are only about 0.01, which means that there nearly exist no low throughput periods for both the original and watermarked flows.

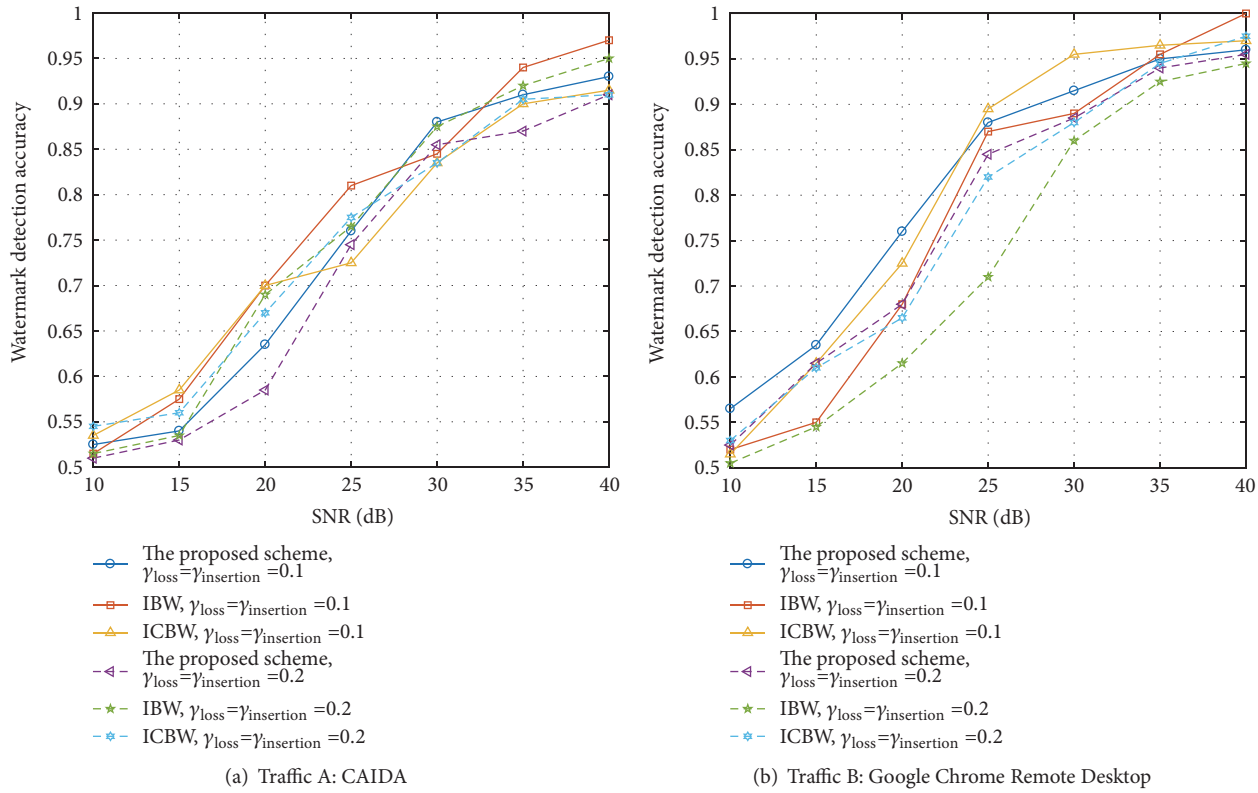
The comparative results in Figure 7 show that the proposed scheme can evade the SSFW detection. The underlying reason for achieving the undetectability is that the proposed insider swapping algorithm can make the throughput of the watermarked flow change within a very small range, as it only swaps very limited number of IPD pairs in each watermarked flow.

4.3. Robustness. To test the robustness, we first consider the watermark detection accuracy in the presence of network jitter. Let P_{TP} be the true positive rate, which denotes the probability of watermarked flows being detected successfully. And let P_{FP} be the false positive rate, which denotes the probability of legitimate flows being misclassified as watermarked ones. Then, the watermark detection accuracy P_R can be obtained by $P_R = (P_{TP} + 1 - P_{FP})/2$. Without loss of generality, we first consider an additive white Gaussian noise (AWGN), which is measured by SNR. Figure 7 shows the watermark detection performance for different schemes under varying network jitter. The horizontal axis denotes SNR and the vertical axis denotes the watermark detection accuracy. Each point in Figure 8 is obtained using the average of 1000 samples.

As shown in Figure 8(a), the robustness of IBW is a little better than the proposed scheme and ICBW for *Traffic A*. When SNR=40dB, the watermark detection accuracy of IBW can achieve 1, whereas those of the proposed scheme and ICBW are both about 0.95. It is apparent that the watermark detection accuracy will increase with a higher SNR. For *Traffic B*, we can find that the performances of the three NFW schemes are very similar. Their watermark detection accuracy is higher than 0.95 when SNR=35dB. Even when SNR=25dB, each of them can still achieve watermark detection accuracy higher than 0.9. From Figure 7 we see that the proposed scheme, ICBW, and IBW each can have a strong capability of anti-jitter.

There may exist other types of interference besides the inherent network jitter, for example, packet loss, dummy packet insertion. These interferences are caused by poor channel condition or maliciously attack that aims to destroy the possible watermark. Thus, we introduce packet loss and dummy packet insertion in the presence of network jitter. Let γ_{loss} and $\gamma_{\text{insertion}}$ be the packet loss rate and dummy packet insertion rate, respectively. The watermark detection performance for different schemes under varying packet loss rate, dummy packet insertion rate, and network jitter is shown in Figure 9. In this evaluation, packet loss is operated by randomly removing packets from the observed flow and dummy packets insertion follows a Pareto distribution. The horizontal axis denotes SNR and the vertical axis denotes the watermark detection accuracy. Each point in Figure 8 is obtained using the average of 1000 samples.

We can find that the proposed scheme can still guarantee a high detection accuracy even when as many as 20% of packets are dropped and injected simultaneously. It is worth noting that packet loss and dummy packet insertion have no significant influence on detection performance of the proposed scheme. When the packet loss rate and dummy



(a) Traffic A: CAIDA

(b) Traffic B: Google Chrome Remote Desktop

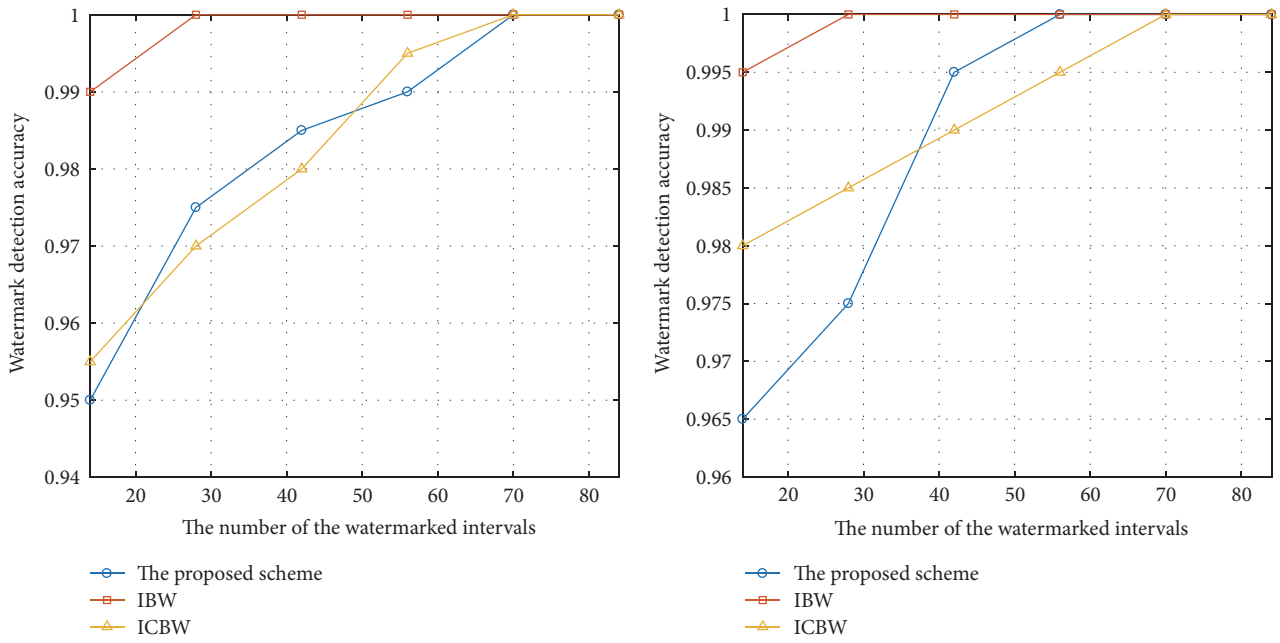
Figure 9: Watermark detection performances for different schemes under varying packet loss rate, dummy packet insertion rate, and network jitter. (a) *Traffic A*, (b) *Traffic B*.

Table 2: The statistical characteristics of the real channel noise.

Traffic Type	Packet Loss Rate	Maximum of Noise	Minimum of Noise	Mean Amplitude of Noise	Standard Deviation of Noise
Traffic A: CAIDA 2016	0%	432.6ms	-382.7ms	3.2ms	14.3ms
Traffic B: Google Chrome Remote Desktop	0.27%	412.4ms	-76.5ms	2.1ms	11.9ms

packet insertion rate each are set to 20%, the watermark detection accuracy of the proposed scheme is about 0.9 for *Traffic A* and 0.95 for *Traffic B*, respectively. The results in Figure 8 show that the proposed scheme can achieve similar accuracy as the other two interval-based NFW schemes in the network environment where network jitter, packet loss, and dummy packet insertion occur simultaneously.

To evaluate the robustness of the proposed scheme in the real network, we replayed the original and watermarked flows with Vultr Cloud Compute (VC2). The source was a cloud server located in Los Angeles, USA. The operation system was CentOS 7 x64 and the memory was 512MB. The destination was another cloud server located in New York, USA. The operation system was Ubuntu 16.04 x64 and the memory was 2048MB. For this case, the end-to-end connection was over multiple hops. We used 100 original flows for both *Traffic A* and *Traffic B*, with the duration of each flow being longer than 40 seconds. To replay the original and watermarked flows, we employed *tcpreplay* [36] and *udpreplay* [37]. For *Traffic A*, as the Ethernet header and payload of each packet in CAIDA dataset are removed, we first padded the Ethernet header and randomly generated payload into each packet of the pcap files. Then we used *tcpliveplay* (one tool in *tcpreplay* suite) to replay the original flows. To replay the watermarked flows, we first extracted the timing sequence from each pcap file and encoded it into watermarked one. The watermarked timing sequence was then used as the packet sending schedule; the inter-packet delay could be realized using the *usleep* function in *tcpliveplay*. For *Traffic B*, the replay process was similar; the main difference was that *udpreplay* was also necessary. We then captured the packets at the destination. The true positive rate and the false positive rate were computed by identifying the watermark from the observed watermarked flows and original flows, respectively. The statistical characteristics of the real channel noise between the source and the destination are shown in Table 2. The real channel noise is obtained by comparing the difference between the IPDs at two ends of connection, which is the average of the first 10 replays.



(a) Traffic A: CAIDA (b) Traffic B: Google Chrome Remote Desktop

FIGURE 10: Watermark detection performances for different schemes with the real channel interference. (a) *Traffic A*, (b) *Traffic B*.

Figure 10 shows the watermark detection accuracy for different schemes under the real channel interference. The horizontal axis denotes the number of the watermarked intervals and the vertical axis denotes the watermark detection accuracy. The watermarking parameters for the three NFW schemes are the same as those used in the above robustness evaluation experiments under AWGN. To evaluate the watermark detection performance with different number of the watermarked intervals, we repeatedly encode the same watermark bit into single flow before we replay the watermarked flow; the detection results are obtained using average decision value. Each point in Figure 10 is obtained using the average of all replayed flows.

As shown in Figure 10(a), the robustness of IBW is still better than that of the proposed scheme and ICBW for *Traffic A*. All the three NFW schemes perform a strong robustness under the real channel interference, and the watermark detection accuracy will rise when the number of the watermarked intervals increases. When we use minimal number of intervals, the watermark detection accuracy of the proposed scheme and ICBW is about 0.95, whereas that of IBW is 0.99. For *Traffic B*, the results in Figure 10(b) show that the watermark detection accuracy of the proposed scheme can achieve 0.98 when we use minimal number of intervals. From Figure 10 we can find that the proposed scheme is robust to the real channel interference.

5. Conclusion and Future Work

In this paper, we have proposed a robust and highly invisible NFW scheme that not only can resist MFA test but also has no statistical distortion under KLD test and K-S test. The watermark bits are encoded into the centroid of randomly chosen intervals using centroid expansion quantization. One of the key contributions of this study is that we propose an insider swapping algorithm which can make a centroid shift by swapping very limit number of IPD pairs in each watermarked interval. The underlying idea of the proposed scheme is to reduce the statistical distortion caused by watermark encoding through two strategies: the first is reducing the expected centroid shift for each interval using centroid expansion quantization, and the other is keeping the distribution of IPDs unchanged by swapping IPDs in the same interval. Using experiments with real traffic and public dataset we have demonstrated the effectiveness of the proposed NFW scheme with respect to invisibility and robustness.

The main drawback of the proposed NFW scheme is its inherent interference in the centroid quantization and the insider swapping. Besides the decoding error resulting from the compensative quantization strategy, the gap between the actual centroid shift and the target centroid shift is the inevitable obstacle for further improving the robustness. Although the insider swapping algorithm can make the watermarked flow maintain more timing characteristics of the original flow as it only swaps limited number of IPD pairs in each interval, it also introduces additional decoding error as it can only make the actual centroid shift approximate to the target one. This error floor can be eliminated using additional centroid shift strategy besides the insider swapping algorithm, or can be relieved using longer near-orthogonal sequences. In addition, we also need to consider the implementation of the proposed scheme in real-time traffic, which still remains a more challenging work, for example, the construction of the packet buffer. These problems should be addressed in future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants Nos. 61602247, 61702235, U1636117, and 61472188), Natural Science Foundation of Jiangsu Province (Grants Nos. BK20150472, BK20160840), CCF-VENUSTECH Foundation (Grant No. 2016011), and Fundamental Research Funds for the Central Universities (30920140121006, 30915012208).

References

[1] A. K. Biswas, D. Ghosal, and S. Nagaraja, "A survey of timing channels and countermeasures," *ACM Computing Surveys*, vol. 50, no. 1, 2017.

[2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The secondgeneration onion router," in *Proceedings of the 23rd USENIX Security Symposium*, USENIX Association, San Diego, CA, USA, 2014.

[3] J. A. Elices and F. Perez-Gonzalez, "The flow fingerprinting game," in *Proceedings of the 2013 5th IEEE International Workshop on Information Forensics and Security, WIFS 2013*, pp. 97–102, IEEE, Guangzhou, China, November 2013.

[4] A. Houmansadr and N. Borisov, "The Need for Flow Fingerprints to Link Correlated Network Flows," *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 205–224, 2013.

[5] X. Wang, D. S. Reeves, S. F. Wu, and J. Yuill, "Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework," in *Proceedings of the IFIP International Information Security Conference*, pp. 369–384, Springer, Paris, France, 2001.

[6] F. Rezaei and A. Houmansadr, "TagIt: Tagging Network Flows using Blind Fingerprints," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 290–307, 2017.

[7] R. Wang, G. Xu, B. Liu, Y. Cao, and X. Li, "Flow watermarking for antinoise and multistream tracing in anonymous networks," *IEEE MultiMedia*, vol. 24, no. 4, pp. 38–47, 2017.

[8] A. Iacovazzi and Y. Elovici, "Network Flow Watermarking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 512–530, 2017.

[9] 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 and Security*, vol. 8, no. 11, pp. 1850–1859, 2013.

[10] A. Houmansadr, N. Kiyavash, and N. Borisov, "Non-blind watermarking of network flows," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1232–1244, 2014.

[11] X. Y. Luo, X. F. Song, X. Y. Li et al., "Steganalysis of HUGO steganography based on parameter recognition of syndrome-trellis-codes," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13557–13583, 2016.

[12] Y. Zhang, C. Qin, W. M. Zhang, F. Liu, and X. Luo, "On the fault-tolerant performance for a class of robust image steganography," *Signal Processing*, vol. 146, pp. 99–111, 2018.

[13] Y. Ma, X. Luo, X. Li, Z. Bao, and Y. Zhang, "Selection of rich model steganalysis features based on decision rough set a-positive region reduction," *IEEE Transactions on Circuits & Systems for Video Technology*, Article ID 2799243, 2018.

[14] R. Archibald and D. Ghosal, "A comparative analysis of detection metrics for covert timing channels," *Computers & Security*, vol. 45, pp. 284–292, 2014.

[15] S. Cabuk, C. E. Brodley, and C. Shields, "IP covert timing channels: Design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004*, pp. 178–187, October 2004.

[16] N. Kiyavash, A. Houmansadr, and N. Borisov, "Multi-flow attacks against network flow watermarks: analysis and countermeasures," 2012.

[17] N. Kiyavash, A. Houmansadr, and N. Borisov, "Multi-flow attacks against network flow watermarking schemes," in *Proceedings of the in USENIX Security Symposium*, pp. 307–320, USENIX Association, San Jose, CA, USA, 2008.

[18] M. Conti, Q. Q. Li, A. Maragno, and R. Spolaor, "The dark side(-channel) of mobile devices: A survey on network traffic analysis," *IEEE Communications Surveys and Tutorials*, 2018.

[19] 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.

[20] M. Lin, G. Liu, W. Liu, and Y. Dai, "Network flow watermarking method based on centroid matching of interval group," in *Proceedings of the 3rd IEEE International Conference on Progress in Informatics and Computing, PIC 2015*, pp. 628–632, IEEE, Shanghai, China, December 2015.

[21] J. Luo, X. Wang, and M. Yang, "An interval centroid based spread spectrum watermarking scheme for multi-flow traceback," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 60–71, 2012.

[22] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proceedings of IEEE Symposium on Security and Privacy (SP '07)*, pp. 116–130, IEEE, Berkeley, CA, USA, May 2007.

[23] A. Iacovazzi, S. Sarda, D. Frassinelli, and Y. Elovici, "DropWat: An Invisible Network Flow Watermark for Data Exfiltration Traceback," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1139–1154, 2018.

[24] X. Luo, P. Zhou, J. Zhang, R. Perdisci, W. Lee, and R. K. Chang, "Exposing invisible timing-based traffic watermarks with BACKLIT," in *Proceedings of the the 27th Annual Computer Security Applications Conference*, pp. 197–206, ACM, Orlando, FL, USA, December 2011.

[25] A. Houmansadr and N. Borisov, "BotMosaic: Collaborative network watermark for the detection of IRC-based botnets," *The Journal of Systems and Software*, vol. 86, no. 3, pp. 707–715, 2013.

[26] X. Wang and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003*, pp. 20–29, ACM, Washington, DC, USA, October 2003.

[27] A. Houmansadr, N. Kiyavash, and N. Borisov, "RAINBOW: A robust and invisible non-blind watermark for network flows," *National Down Syndrome Society*, 2009.

[28] A. Houmansadr and N. Borisov, "Towards improving network flow watermarks using the repeat-accumulate codes," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1852–1855, IEEE, Prague, Czech Republic, May 2011.

[29] B. Assanovich, W. Puech, and I. Tkachenko, "Use of Linear Error-Correcting Subcodes in Flow Watermarking for Channels with Substitution and Deletion Errors," *IFIP International Conference on Communications and Multimedia Security*, pp. 105–112, 2013.

[30] A. Houmansadr and N. Borisov, "SWIRL: A scalable watermark to detect correlated network flows," *National Down Syndrome Society*, 2011.

[31] Z. Lin and N. Hopper, "New attacks on timing-based network flow watermarks," in *Proceedings of the in 21st USENIX Security Symposium*, pp. 20-20, USENIX, Bellevue, WA, USA, 2012.

[32] Y. Xiang, I. Natgunanathan, Y. Rong, and S. Guo, "Spread spectrum-based high embedding capacity watermarking method for audio signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 12, pp. 2228–2237, 2015.

[33] "CAIDA," 2018, http://www.caida.org/data/passive/passive2016dataset.xml.

[34] W. Liu, G. Liu, J. Zhai, Y. Dai, and D. Ghosal, "Designing analog fountain timing channels: Undetectability, robustness, and model-adaptation," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 677–690, 2016.

[35] X. Luo, J. Zhang, R. Perdisci, and W. Lee, "On the Secrecy of Spread-Spectrum Flow Watermarks," in *Computer Security – ESORICS 2010*, vol. 6345 of *Lecture Notes in Computer Science*, pp. 232–248, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[36] "Tcpreplay," 2018, http://tcpreplay.synfin.net/.

[37] "Udpreplay," 2018, https://github.com/rigtorp/udpreplay.