Detection and Parameter Estimation for Jitterbug Covert Channel Based on Coefficient of Variation

Hao Wang[1], Guangjie Liu[1], Jiangtao Zhai[2], Yuewei Dai[1,2]
[1]School of Automation, Nanjing University of Science and Technology
Nanjing, China.
[e-mail: whaanog@gmail.com, gjieliu@gmail.com]
[2]School of Electronics and Information, Jiangsu University of Science and Technology
Zhenjiang, China.
[e-mail: jiangtaozhai@gmail.com, dywjust@163.com]
*Corresponding author: Hao Wang

Received September 1, 2015; revised November 20, 2015; revised December 29, 2015; revised February 3, 2016; accepted February 27, 2016; published April 30, 2016

Abstract

Jitterbug is a passive network covert timing channel supplying reliable stealthy transmission. It is also the basic manner of some improved covert timing channels designed for higher undetectability. The existing entropy-based detection scheme based on training sample binning may suffer from model mismatching, which results in detection performance deterioration. In this paper, a new detection method based on the feature of Jitterbug covert channel traffic is proposed. A fixed binning strategy without training samples is used to obtain bins distribution feature. Coefficient of variation (CV) is calculated for several sets of selected bins and the weighted mean is used to calculate the final CV value to distinguish Jitterbug from normal traffic. Furthermore, the timing window parameter of Jitterbug is estimated based on the detected traffic. Experimental results show that the proposed detection method can achieve high detection performance even with interference of network jitter, and the parameter estimation method can provide accurate values after accumulating plenty of detected samples.

Keywords: covert channel detection, Jitterbug, coefficient of variation, parameter estimation

This work was supported by the NSF of China (Grant No. 61472188, 61170250), the NSF of Jiangsu province(Grant No. BK20150472) and the Fundamental Research Funds for the central Universities (Grant No. 30920140121006).

http://dx.doi.org/10.3837/tiis.2016.04.026 **ISSN : 1976-7277**

1. Introduction

In espionage activities against computer systems, the information leakage can commonly be prevented by some preset security equipments, such as network firewalls, IDS, or other traffic checking devices, which can identify the unauthorized or abnormal network traffic. If the information is hided into the normal traffic, these equipments will no longer work. Consequently, we can conceal the very existence of the data transmission, and this kind of transmission art is called network covert channel which is a stealthy communication technique utilizing redundancies of network protocols or packet-sequence characteristics to transfer secret message. It also can be named network steganography, which refers to the field of image steganography [1]. Similar to the concept of covert channel in multi-level security (MLS) systems, network covert channel can also be divided into storage and timing channels [2]. Network covert storage channel is constructed by modifying some unused or insensitive bits of protocol header in network packets. Network covert timing channel is constructed by modulating secret message into packet rates/inter-packet delays (IPDs). Besides the time sequence, other covert channels based on characteristics of packet-sequence are usually considered as the timing case. Network covert timing channel is also the current focus issue on network steganography research.

Padlipsky et al. [3] firstly described the principle of the on/off timing channel in which the sender either transmits or stays silent in each time interval to represent 0 or 1. Girling [4] also proposed a covert timing channel which can transmit secret message by particular delays between successive transmissions imposed by a sender. On this basis, Berk et al. [5] illustrated the delay-based channel by two delays (binary channel) and multiple delays (multi-symbol channel), and then investigated how to find the optimal symbol distribution to maximize the channel capacity, the possible detection schemes were also discussed for these channels. The delay-based timing channel does not require a synchronized clock, while an on/off timing channel needs a synchronization mechanism to ensure decoding accuracy. Thus, Cabuk et al. [6-7] implemented the on/off timing channel which introduces start of frame (SOF) and silent intervals to synchronize between sender and receiver. The modulation of inter-packet delays may change the overt communication pattern and make itself more exposed. To solve this problem, Gianvecchio et al. [8] proposed a model-based covert timing channel called MBCTC. In their scheme, the channel mimics the observed behavior of legitimate network traffic to evade detection. Liu et al. [9] proposed a simple binary covert timing channel based on Gianvecchio's framework, and this method is more practical in encoding/decoding and has a lower bit error rate. Although these covert channels can resist detection methods based on statistical properties, the algorithms are usually complicated with low data rate and are not easy to be deployed in a real network.

It is obvious that the detection against the covert timing channel is also an important problem. Researchers have made lots of endeavors. Cabuk et al. [6] proposed a detection method against on/off covert timing channel, in which two measures, regularity (i.e. patterns in the variance) and ε-similarity (i.e. similarity between adjacent inter-arrival times), were defined to judge whether the traffic was a covert one. Peng et al. [10] showed that the Kolmogorov-Smirnov (KS) test is able to distinguish watermarked inter-packet delays from normal inter-packet ones, and the KS test is utilized to detect abnormal shapes of IPDs created by covert timing channels [11]. Gianvecchio et al. [12] proposed an entropy-based detection approach which uses of entropy (EN) and corrected conditional entropy (CCE) to describe

abnormal shape or abnormal regularity separately. The approach is able to detect most of the existing covert timing channels but needs legitimate traffic samples to determine the bin ranges. Due to the various manners of network traffic, it is hard to choose proper reference samples to adapt to all of the cases, and the threshold setting is also a great challenge. Thus, the detection performance of entropy-based test may be reduced in practice.

This paper is focused on a typical delay-based covert timing channel named Jitterbug [13] which is originally designed as a keyboard device to leak typed messages. Jitterbug is a passive covert timing channel that utilizes the existing network traffic as over channels and transmits secret message by modifying their IPDs. The interactive communication applications such as Skype may generate continued traffic that can be applied as an overt channel for Jitterbug. Due to the widespread use of interactive communication applications, Jitterbug may become a very practical covert channel to leak information over Internet. Some improved covert timing channel methods, such as Liquid [14] and Mimic [15], also have the similar basic encoding/decoding scheme. Thus, the researches on designing detection scheme against this basic manner covert channel (i.e. Jitterbug) are imperative and worth. For implementing the detection of Jitterbug, we count the bins distribution of inter-packet delays in a fixed binning strategy and then calculate the coefficient of variation (CV) of partial successive bins that belong to a significant region. After that, these CV values are utilized to distinguish between normal traffic and Jitterbug traffic. We also make an effort to estimate the main decoding parameter of Jitterbug, which is the basis of extracting secret message from detected traffic. To the best of our knowledge, it is the first attempt to estimate the network covert timing channel parameters.

The remainder of this paper is organized as follows. Section 2 introduces the principle of Jitterbug and corresponding analysis. Section 3 gives our detection scheme and parameter estimation method. Section 4 presents the experimental results. Section 5 concludes the whole paper.

2. Background and related work

2.1 Revisiting Jitterbug

Shah *et al.* [13] designed Jitterbug covert channel as a hardware interception device installed between the computer and its keyboard. The covert timing channel performed by Jitterbug is a passive one utlizing the existing communication traffic, so no additional traffic needs to be generated for transmitting the secret message. The communication scenario of Jitterbug is illustrated by **Fig. 1**. In the scenario, the Jitterbug device embeds information into the keystroke timing in the form of small supplementary jitters. If each keystroke is sent within a single packet, the timing information will remain in the inter-packet delays. Therefore, the overt channel of Jitterbug must be an interactive network application (e.g. Telnet, SSH) in which each keystroke corresponds to a packet. The receiver monitors the packet flows of these applications and decodes the secret message from the manipulated IPDs. Although performance of keyboard Jitterbug may be affected by keyboard buffering, OS scheduling, and so on, its potential advantage is only to require a compromised input device rather than a compromised host.

Fig. 1. Communication scenario of Jitterbug

The mechanism of keyboard Jitterbug can also be extended to a useful covert channel method when a compromised host is obtained. In this scenario, a special driver for Network Interface Card (NIC) is considered to add timing information instead of hardware device. The secret message can be encoded by adding extra packet delays just in the software manner. In this case, the choice of overt channel can be extended to the applications containing continued traffic session (e.g. VoIP).

For the Jitterbug covert channel, the inter-packet delays are manipulated to satisfy certain properties depending on the secret message to be sent. In a binary encoding case, letting b_i denote a bit sequence and w (in milliseconds) denote the Jitterbug timing window which is used to manipulate the delays to represent encoding symbols, IPDs sequence denoted as δ_i should be manipulated to satisfy Eq. (1).

$$(\delta_i - s_i) \bmod w = \begin{cases} 0 & if\ b_i = 0 \\ \lfloor w/2 \rfloor & if\ b_i = 1 \end{cases} \tag{1}$$

Where $\delta_i = \delta_i' + \Delta t_i + s_i$ with δ_i' being original inter-packet delay between the packet p_{i+1} and p_i, and Δt_i being the added delay to satisfy the modulo operation. If the modulo operation is used only on δ_i, the manipulated IPDs will be changed to the value around multiples of $w/2$. To prevent the IPDs clustering around multiples of $w/2$, a pseudo-random sequence s_i with integer millisecond values from 0 to w-1 is additionally added, and this sequence is assumed only to be known by the sender and receiver.

After the transmission on the Internet link, the packet p_{i+1} and p_i arrive at the receiver end. The modulated IPD δ_i is changed to $\hat{\delta}_i$ due to the link delay jitters. The receiver decodes the message bit $\hat{b}_i$ with the shared s_i according to Eq. (2).

$$\hat{b}_i = \begin{cases} 0 & if\ \ (\hat{\delta}_i - s_i) \bmod w \in [0, w/4) \cup [3w/4, w) \\ 1 & if\ \ (\hat{\delta}_i - s_i) \bmod w \in [w/4, 3w/4) \end{cases} \tag{2}$$

2.2 Entropy-based detection method and its limitation

Except the EN test [12], most of the current detection algorithms mentioned in Section 1 fail to distinguish Jitterbug from normal traffic. The first-order entropy is estimated in EN test to

measure the shape of the investigated traffic. Due to finite number of samples, the empirical probability density function based on the method of histogram is employed to replace probability density function. The entropy-based detection uses equiprobable binning strategy to decide how the IPDs are partitioned. The binning strategy needs to work on normal sample traffic to determine the range of each bin. Thus, the bins may have different width while the sample IPDs distribute equiprobably in each bin. The total number of bins is another important parameter of binning strategy. There is a trade-off in choosing the number of bins, a larger number of bins retains more information about the distribution of the traffic while a small number of bins is able to measure the regularity of the traffic. Consequently, the EN test and CCE test adopt fine-grain binning and coarse-grain binning separately to obtain both advantages. Actually, the EN test is designed to measure how closely the tested IPDs fit the normal reference traffic and the number of its bins is as many as 65536 columns. Although Jitterbug only adds tiny delays to original IPDs, the changes of distribution in the bins are still perceived due to the fine-grain binning strategy. Commonly, EN test scores of normal traffic approach the upper-bound value of the first-order entropy and the lower EN test scores imply the possible existence of a Jitterbug covert channel.

However, there are some restrictions for deploying EN test in a real network environment. As mentioned above, the detection performance significantly depends on the binning strategy determined by training IPD samples, which is verified by our experiment. In the experiment, two SSH traffic sets are extracted from the traffic archive of WIDE Project [16]. These two SSH traffic sets belong to two absolutely different links, so the corresponding two IPDs sequences are unrelated to each other. The first IPDs set is divided into two subsets: Training set and Normal-1 set. The second one is named Normal-2 set. Two Jitterbug IPDs sequences, named Jitterbug-1 and Jitterbug-2 set, are generated according to the algorithm described in subsection 2.1 using Normal-1 and Normal-2 set, respectively. The lengths of IPDs sequences in these sets are as long as 100,000. The binning strategy is determined based on the Training set using the method given in Ref. [12]. The EN tests are performed on Normal-1, Jitterbug-1, Normal-2, and Jitterbug-2 with the recommended detection windows size of 2,000 and the corresponding EN scores are shown in **Fig. 2**.

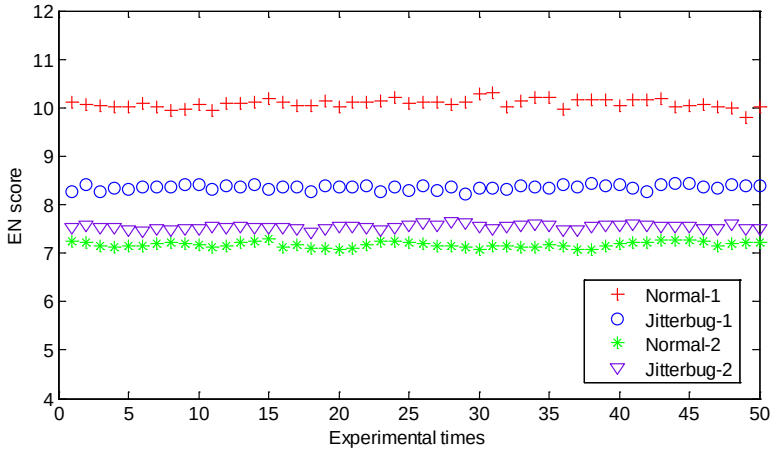


Fig. 2. Model mismatching case of EN test

Fig. 2 shows that Jitterbug-1 can be distinguished from Normal-1, while Normal-2 and Jitterbug-2 are both considered to be abnormal due to the low EN scores. It is because that Normal-1 set is extracted from the same source of Training set and the training samples have good indication to obtain proper bins and lead to ideal detection results. However, Normal-2

set extracted from another source has a notable deviation from training samples due to the different manners of SSH traffic. Hence, training samples become invalid for guiding binning and the EN scores of Normal-2 and Jitterbug-2 are hard to be separated by a constant threshold. We called this state a model (i.e. binning strategy based on training samples) mismatching case of EN test.

To solve the mismatching problem, a possible solution is to build enough models for adopting the diversity of the network application traffic. Technically, we can use an empirical estimation to get a normal distribution of IPDs from observed data, and the distribution regarded as a model of EN test can only be valid for the same type of application traffic under the same network condition. In other words, the inter-packet delays are related to the network application, the network connection environment, and the processing ability of the computer, thus the solution is infeasible in a real Internet detection scenario. The other solution is to study the detection methods based on the features of the covert channels themselves. With the limitation just valid to a specific covert channel method, these kinds of detection methods are more reliable. It is also the main motivation of this paper.

3. Proposed method

3.1 Binning strategy

According to Eq. (1), Jitterbug firstly adds tiny delays Δt_i to the original inter-packet delays δ_i', which cause that IPDs cluster around $mw/2$ (m=1, 2, 3,…). To smooth the obvious clustering abnormality, the pseudo-random sequence s_i is additionally added to scatter each clustered IPDs over all integer millisecond values in the range [$mw/2$, (m+2)$w/2$). To discover and investigate the fine change of the IPDs histogram, the following binning strategy is adopted in this paper.

$$\begin{aligned} &[L, L+B), [L+B, L+2B), ..., [L+(n-1)B, L+nB] \\ &L = \frac{\lfloor 1000 d_{\min} \rfloor}{1000} \\ &n = \lfloor (d_{\max} - d_{\min})/B \rfloor + 1 \end{aligned} \tag{3}$$

Where B is the width of each bin and is set to 0.001s, $d_{\max}$ and $d_{\min}$ denote the maximum and minimum values of observed successive IPDs, respectively.

To observe the effect of binning strategy on Jitterbug, an experiment was performed. Skype-VoIP stream between China Nanjing and Beijing was captured and a total of 10,000 IPDs are recorded. Based on the binning strategy defined by Eq. (3), the statistical result of normal IPDs amount for each bin is shown in **Fig. 3.** Processing these IPDs according to the Jitterbug method with the parameter w=20ms, the statistical result of Jitterbug's IPDs amount for each bin is shown in **Fig. 4.** The X-axis of the figures represents the label of each bin and the bin ranges are calculated by Eq. (3). It is clear that Jitterbug will cause approximately equal bin values of successive bins, which we call feature regions. However, this characteristic does not occur in the histogram of normal traffic. The characteristic is also unsurprising according to the principle of Jitterbug. It is expected that the measure of this characteristic will help detect and estimate the Jitterbug covert channel.

It also can be found that B=0.001s is a minimum value for choosing. If B is less than 0.001s, such as 0.0005s, we may not observe successive bins with approximately equal bin values (named abnormal feature). The bins whose range does not contain integer millisecond value will get smaller bin values (i.e. fewer IPDs fall into these bins). It means that the binning strategy on this bin width lost the ability to represent the abnormal feature which is caused by Jitterbug's parameter s_i. On the other hand, if B is greater than 0.001s, such as 0.002s, we can still catch the abnormal feature (other values of B in multiples of 0.001s can also make it), but the length of these successive bins will be cut in half. If the length is too small, the detection test will cause remarkable false alarm because the normal traffic may form the similar feature within small length of successive bins. Thus, the value of B>0.001s is also not feasible for the detection.

Compared with EN test, the application of fixed binning strategy removes the dependence of training samples and avoids the model mismatching problem. Moreover, the EN test also can-not work well under this binning strategy due to the lack of reference entropy value (the original reference value is calculated by training samples) to distinguish Jitterbug from normal traffic.

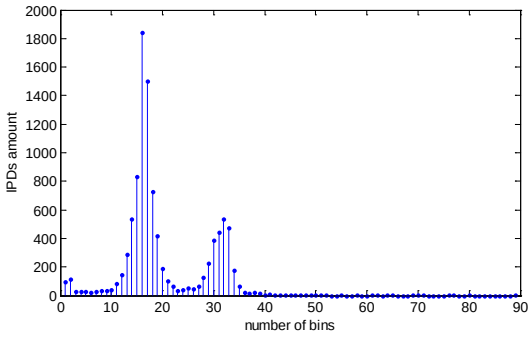


Fig. 3. Histogram of normal traffic

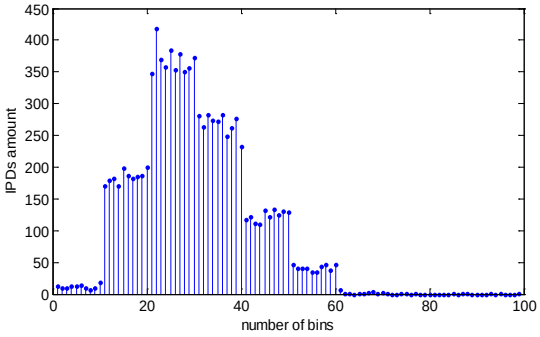


Fig. 4. Histogram of Jitterbug traffic

3.2 Detection based on coefficient of variation

Coefficient of variation is a standardized measure of dispersion of data points in a data sequence around the mean. It is defined as the ratio of the standard deviation to the mean [17]. CV is a dimensionless number made for comparing data sets with different units or widely different means. In this paper, CV is used to measure partial successive bins of the feature regions which we call significant regions. According to the characteristic mentioned above, the CV values of the significant regions in the histogram of Jitterbug traffic, which contains approximately equal bin values, are expected to be smaller than that of most regions in the histogram of normal traffic.

The selection of significant regions is the most critical step in our detection scheme. It is especially hard for this step because of the finite samples, it is worth nothing that 10,000 samples counted in **Fig. 3** and **Fig. 4** are only for distinct illustration, the actual detection window needs to be much smaller for a quick responding. The significant regions are selected by two principles, one is that the region must contain enough proportion of the samples, and the other is that the region must start at a proper position in order to obtain the remarkable feature of samples. The width of region, which is denoted as S (i.e. the total number of bins in the region) must less than $\lfloor w/2 \rfloor$ (in ms), is set to 7, because values of w less than 15ms may cause significant Bit Error Rate according to Table 1 of Ref. [13]. A sliding window with the

same width is used to select significant regions and the detection process can be described as follows.

First, in a detection window *W*, count the histogram of IPDs under the binning strategy mentioned above, and then obtain a sequence of IPDs amount in each bin.

$$t = (t_1, t_2, ..., t_n) \tag{4}$$

Second, slide the CV computation window (i.e. the width of significant region, *S*) on *t*, and obtain a new sequence $\hat{t}$ based on Eq. (5).

$$\hat{t} = \left(\hat{t}_1, \hat{t}_2, ..., \hat{t}_{n-S+1}\right), \quad \hat{t}_i = \sum_{j=i}^{i+S-1} t_j \tag{5}$$

The position set *l* of peak points of $\hat{t}$ is written as Eq. (6). The position set $\hat{l}$ of remarkable points of $\hat{t}$ is written as Eq. (7). Then the set l^* is obtained using Eq. (8) which is taken as the start positions of each significant region.

$$l = \{l_1, l_2, ..., l_x\},\ \hat{t}_{l_i} > \hat{t}_{l_i+1} \text{ and } \hat{t}_{l_i} > \hat{t}_{l_i-1} \tag{6}$$

$$\hat{l} = \left\{\hat{l}_1, \hat{l}_2, ..., \hat{l}_y\right\},\ \hat{t}_{\hat{l}_i} > 0.1W \tag{7}$$

$$l^* = l \cap \hat{l} = \left\{l_1^*, l_2^*, ..., l_z^*\right\} \tag{8}$$

Third, the coefficient of variation of each significant region is calculated by Eq. (9).

$$\begin{aligned} \overline{t_i} &= \frac{\hat{t}_{l_i^*}}{S}, \quad i = 1, 2, ..., z \\ \sigma_i &= \sqrt{\frac{1}{S}\sum_{j=l_i^*}^{l_i^*+S-1}\left(t_j - \overline{t_i}\right)^2} \\ cv_i &= \frac{\sigma_i}{\overline{t_i}} \end{aligned} \tag{9}$$

Fourth, the weighted mean of all computed CVs is obtained by Eq. (10) and the weighted coefficient related to IPDs amount of significant region is written as Eq. (11).

$$\overline{cv} = \sum_{i=1}^{z} \hat{w}_i cv_i \tag{10}$$

$$\hat{w}_i = \hat{t}_{l_i^*} \Big/ \sum_{j=1}^{z} \hat{t}_{l_j^*}, \quad i = 1, 2, ..., z \tag{11}$$

Finally, a threshold *T* is utilized to distinguish Jitterbug from normal traffic. If $\overline{cv} < T$, the traffic within *W* is considered to be a Jitterbug covert communication.

3.3 Parameter estimation

Detecting a covert channel is not the final step of the countermeasures, recognizing the type of covert channel, estimating decoding parameters and extracting secret message are also important issues for data leakage forensics. Since our detection scheme is based on the feature of Jitterbug itself, the detected traffic is surely recognized as Jitterbug one. So in this subsection, we pay our attention only on the parameter estimation.

For Jitterbug, there are two parameters w and s_i that need to be estimated. The pseudo-random sequence s_i is commonly generated by a shared key, so its estimation falls into the category of cryptography and is not considered in this paper. Based on the observation of **Fig. 4** and the principle of Jitterbug, w is related to the width of the successive bins with approximately equal bin values, which we call bin cluster. So if the edges of these clusters can be accurately located, the estimated value of w will be obtained. Additionally, since the parameter estimation works in an offline situation, we can use more detected IPD samples to carry on the estimation. The estimation process can be described as follows.

First, after accumulating several detected windows of Jitterbug traffic, the histogram of IPDs is counted using the above binning strategy, and the IPDs amount in each bin is denoted as Eq. (12) with a total of n bins.

$$t' = (t'_1, t'_2, ..., t'_n) \tag{12}$$

Second, to find the edges of each cluster, the forward difference sequence is calculated as Eq. (13).

$$t^* = (t^*_1, t^*_2, ..., t^*_{n-1}), \quad t^*_i = \left| t'_{i+1} - t'_i \right| \tag{13}$$

Most edges cause peak points of t^*, and a position sequence of these peaks is shown as Eq. (14). Some small peak points are excluded by satisfying $t^*_{P_i} < 0.3\max\{t^*_{P_i}\}$, and a new position sequence is shown as Eq. (15).

$$P = (P_1, P_2, ..., P_x), \quad t^*_{P_i} > t^*_{P_i-1} \text{ and } t^*_{P_i} > t^*_{P_i+1} \tag{14}$$

$$\hat{P} = (\hat{P}_1, \hat{P}_2, ..., \hat{P}_y), \quad t^*_{\hat{P}_j} \geq 0.3\max\{t^*_{P_i}\} \tag{15}$$

Third, according to the principle of Jitterbug, the distance of adjacent peak points is considered to be the approximation of $w/2$ (in ms), and a distance sequence is obtained in Eq. (16).

$$D = (D_1, D_2, ..., D_{y-1}), \quad D_i = \hat{P}_{i+1} - \hat{P}_i \tag{16}$$

When two adjacent clusters have a close amount of IPD samples, the edge between them might not be found and a larger distance value will occur. So the unexpected abnormal values should be excluded. For this, D is firstly sorted in order from the lowest to the highest and the sorted one is denoted as Eq. (17).

$$\hat{D} = (\hat{D}_1, \hat{D}_2, ..., \hat{D}_{y-1}) \tag{17}$$

Finally, the estimation value w^* (in ms) is calculated by Eq. (18) with a partial value of the sorted distance sequence, which is used to delete the larger or smaller unreliable values. Consequently, the estimated value is close to the real parameter with high probability.

$$w^* = \begin{cases} \dfrac{2}{y-1}\sum_{i=1}^{y-1}\hat{D}_i, & y \le 3 \\ \dfrac{2}{\lceil 2(y-1)/3 \rceil - \lfloor (y-1)/3 \rfloor}\sum_{i=\lfloor (y-1)/3 \rfloor+1}^{\lceil 2(y-1)/3 \rceil}\hat{D}_i, & y > 3 \end{cases} \tag{18}$$

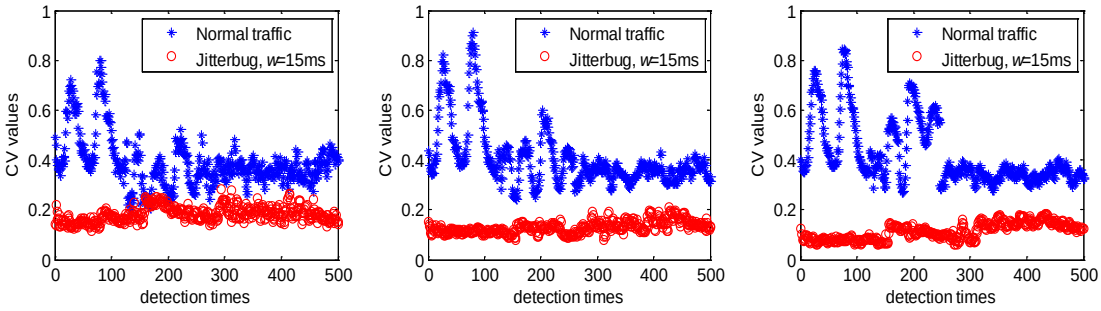
4. Experimental Results

4.1 Datasets and threshold determination

To evaluate the effectiveness of our detection scheme, several different types of normal traffic are prepared as overt channels. Since VoIP traffic is considered to be a suitable overt channel for network covert communication [18-19], Skype and QQ (the most widely used instant messaging software in China) are chosen to generate overt VoIP traffic. We started voice communications from our laboratory (Nanjing) to outside locations (Beijing, Shanghai, Chengdu) by two kinds of software separately and captured these traffics in our laboratory's gateway to make up two sets named Skype set and QQ set. The SSH set was obtained from the traffic archive of WIDE Project [16], which the traffic was captured from the samplepoint-F of WIDE backbone. Each of these sets contains over 600,000 packets.

Jitterbug traffic was generated based on the three normal traffic sets. We replayed the packets of each set and added delays according to the encoding scheme of Jitterbug. It is considered to be a software implementation of Jitterbug that manipulates IPDs directly. We also captured these traffics in the gateway and obtained three sets named Jitterbug-SSH, Jitterbug-Skype, and Jitterbug-QQ. Moreover, changing the encoding parameter w will generate more Jitterbug traffic for testing.

According to the discussion, the threshold T is a critical factor for the detection performance. To find a proper value of T, the CV values of normal traffic and Jitterbug traffic are calculated separately. The normal traffic is combined by Skype, QQ, and SSH sets. The packet number of the combined set is as many as 500,000, and the tested Jitterbug traffic is combined by the corresponding Jitterbug-Skype, Jitterbug-QQ, and Jitterbug-SSH. The packet number of Jitterbug traffic is the same as that of normal traffic. Before the threshold determination test, we performed an initial detection windows size effect test. In the test, the detection window size was set to 1,000, 2,000, and 3,000, respectively. **Fig. 5** shows that a larger detection window makes detecting the Jitterbug traffic easier.



(a) W=1,000 (b) W=2,000 (c) W=3,000

Fig. 5. CV values of normal traffic and Jitterbug traffic in a fixed w with different detection windows

In the threshold determination test, the detection window was set to 1,000, and the Jitterbug encoding windows size w was set to 15ms, 20ms, 25ms, and 30ms. According to the results of **Fig.5**, the chosen testing parameters are conserved enough to guarantee that the obtained threshold is reliable for practical conditions. The distribution of CV values is shown in **Fig. 6**. The CV values of two types of traffic have few overlaps and the CV values of Jitterbug traffic maintain in a stable range which are also less affected by the change of different w values. The threshold T to distinguish these two kinds of traffic was chosen as 0.25, which results in the average detection error rate to not be larger than 2%.

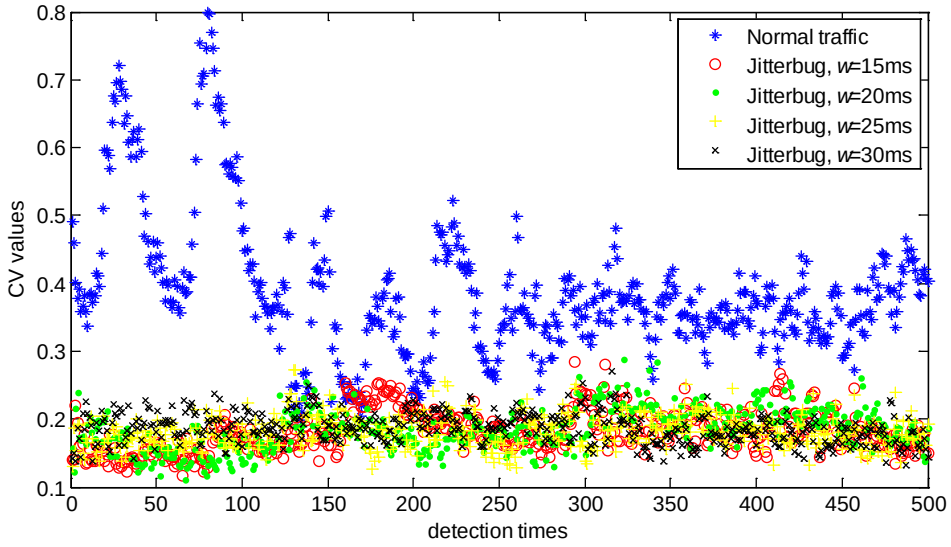


Fig. 6. CV values of normal traffic and Jitterbug traffic with different w in a fixed detection window

4.2 Results and analysis

4.2.1 Detection Performance

The detection performance under different detection windows, different encoding parameters and different overt channels were evaluated. The detection window W was chosen as 1,000, 1,500, 2,000, and 2,500 while timing window w of Jitterbug was set to 15ms, 20ms, 25ms, and 30ms, respectively. With the fixed threshold 0.25, the detection rate (denoted as TP) and false alarm rate (denoted as FP) are obtained by sliding the non-overlapped detection window on each test set. The detection results are summarized in **Table 1**.

Table 1. Detection results on different states

W	*w* (in ms)	Jitterbug-Skype	Skype	Jitterbug-QQ	QQ	Jitterbug-SSH	SSH
		TP	FP	TP	FP	TP	FP
1,000	15	98.33%	1.5%	98.83%	1.17%	98%	1.83%
	20	98.83%		98.17%		97.17%	
	25	99.17%		98%		97.83%	
	30	98.33%		98%		98.17%	
1,500	15	98.75%	1%	98.5%	1.25%	98.25%	1.75%
	20	99.25%		99%		98.25%	
	25	99%		100%		99%	
	30	99%		99%		98%	
2,000	15	99.33%	0.33%	99.67%	0.33%	98%	0.67%
	20	100%		99.33%		98.67%	
	25	98.67%		99.33%		100%	
	30	100%		100%		99.33%	
2,500	15	100%	0%	100%	0%	100%	0%
	20	100%		100%		100%	
	25	100%		100%		100%	
	30	100%		100%		100%	

As listed in **Table 1**, our scheme is reliable to detect Jitterbug traffic in all cases. It is obvious that the larger detection window size is helpful to make more accuracy detection. When the detection window is as high as 2,500, the detection error rate decreases to zero. However, the detection windows in the practical scenario are expected to be as small as possible to economize the memory and computation resources and to provide flexibility and rapidity. So with the comprehensive consideration of these factors, the recommended detection window is chosen as 1,500. Furthermore, the timing window w utilized in the table has little effect on detection results. Although a larger value of w needs more IPD samples for accurate detection (i.e. a larger detection window), it is not often used in Jitterbug covert channel because of the lower data rate. Thus, the chosen detection parameters are still reliable and flexible in a practical detection scenario.

4.2.2 Analysis of bin width setting

For further investigation, the influence on detection under the change of bin width is analyzed by carrying out several experiments. The normal traffic was combined from Skype, QQ, and SSH sets, while the Jitterbug traffic was combined from the corresponding Jitterbug sets with w=20ms. For fixed detection parameters W=1500, T=0.25, and S=7(the width of significant region is also the computing window of CV values), the detection performance of different bin widths under the same computing window is shown in **Fig. 7**. The average error rate (denoted as AP) is defined as AP=(1-TP+FP)/2, the lower value of AP implies the better detection performance.

Fig. 7 shows that the minimum value of AP can be obtained when B=0.001s. When B<0.001s, the detection rate drops to the very low value with the decrease of B. These bin widths may not catch the abnormal feature (successive bins with approximately equal bin values) of Jitterbug because some of bins whose range does not contain integer millisecond values will get smaller bin values (i.e. fewer IPDs fall into these bins). In a computing window, the larger appearance probability of this kind of bins leads to the greater CV value and the lower detection rate. When B>0.001s, taking the values in multiples of 0.001s, the detection

rate is also reduced with the increase of *B*. Although these bin widths can still catch the abnormal feature of Jitterbug, the width of the abnormal feature region has been reduced. If the width of the abnormal feature region is less than computing window *S*, the CV values of each computing window will raise and the detection rate will reduce. As a result, the fixed value of *S* in the detection test is not suitable for *B*>0.001s and *S* need to be adapted to the change of *B*. In fact, the value of *S* need to be satisfied with $S \cdot B \le w/2$. Thus, another group of experiments are performed with different values of *S*, and then the detection performance of different bin widths under the adaptive computing window is shown in **Fig. 8.**

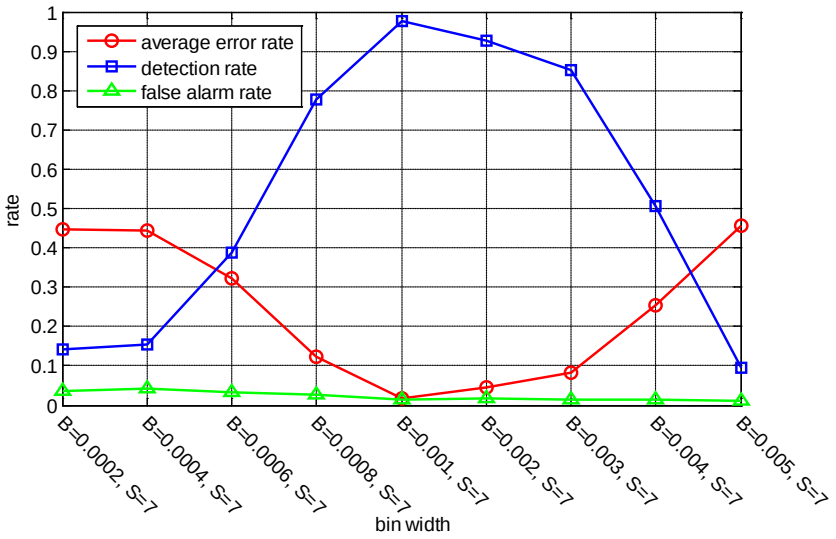


Fig. 7. Detection performance of different bin widths under the same computing window

Fig. 8 shows that the minimum value of AP is also obtained when *B*=0.001s. When *B*<0.001s, these bin widths still cannot catch the abnormal feature, the detection rate deteriorates faster due to the increased value of *S* comparing with **Fig. 7.** That means the appearance probability of the bins with smaller bin values in a computing window increases and leads to lower detection rate. When *B*>0.001s, taking the values in multiples of 0.001s, the detection rate maintains greater than 0.95 under each adaptive computing window. However, with the decrease of *S*, the false alarm rate raises rapidly. This is because the value of *S* is relatively small and the normal traffic may form the similar abnormal feature within small computing window. Thus, even if these bin widths get sound detection rate under adaptive computing window, the average error rate is still unacceptable for the detection. From the two figures, it can be found that 0.001s is a suitable value of bin width for the proposed detection scheme.

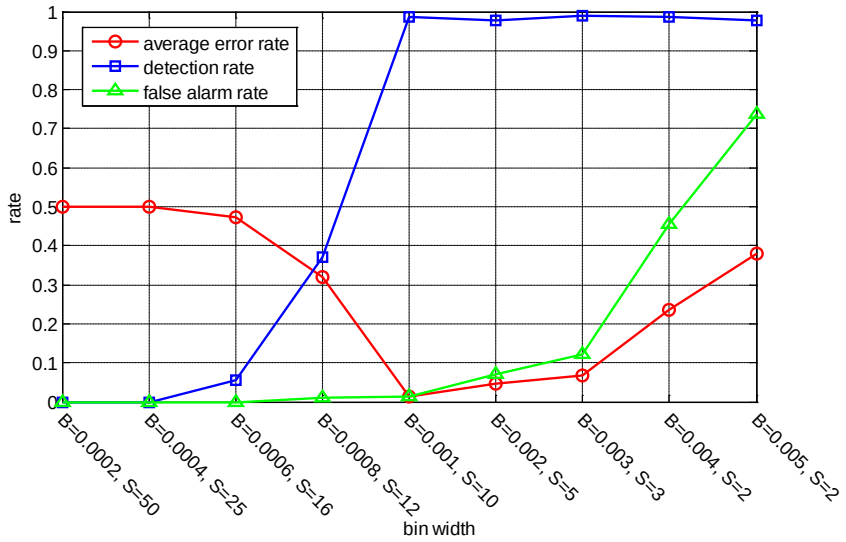


Fig. 8. Detection performance of different bin widths under the adaptive computing window

4.2.3 Robustness against network jitter

Commonly, the network covert channel detector is used to prevent the data leakage from inside to outside. Hence the detector is always deployed at the network boundary which is close to the sender. When the detector is close enough to the sender (within several routers), the captured IPDs will not suffer from the influence of network jitter. When the detector has to inspect the potential covert channel communication of users in a district, the detector has to be deployed in a backbone network. Therefore, the captured IPDs are unavoidably affected by network jitter caused by the forwarding equipments (e.g. routers). So, the detection performance resisting network jitter should be tested in the latter scenario. To simulate the network jitter, we add a network link emulator between Jitterbug sender and the detector. The network emulator was implemented on Linux host with double NICs. The Netem [20] was used to simulate packet losses, delays, delay jitters, and so on. For W=1500, w=15ms, 20ms, 25ms, and T=0.25, the detection rate and Jitterbug bit error rate (BER) are investigated under different intensity of jitters. Since there is no acknowledged model for network jitter, the normal distribution with zero mean and standard deviation σ is used to model the network jitter. And 3σ (in ms) is used to measure the intensity of jitters.

As shown in **Fig. 9**, with the increased the network jitter intensity, the BER of Jitterbug becomes larger and larger, which reflects that Jitterbug is not robust enough to resist network jitter. For fixed network jitter intensity, the smaller Jitterbug timing window is more sensitive than the larger one and results in a greater BER value. Fortunately, the detector's performance is always acceptable. **Fig. 9** shows that the detection rate maintains high performance under minor network jitter, and it is still over 70% despite the BER is as high as 50%. Although the network jitter may change the distribution of IPDs, the proposed feature based on partial successive bins can still represent Jitterbug's characteristic. Thus, the detection method is robust to resist network jitter.

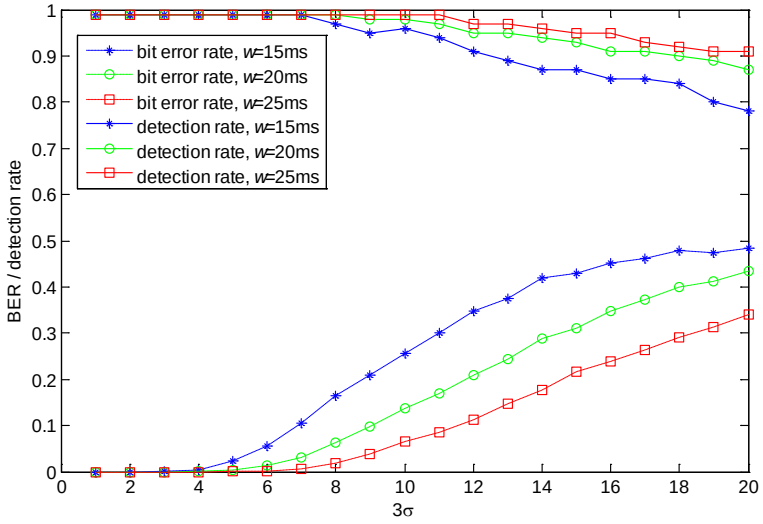


Fig. 9. Influence of network jitter

4.2.4 Parameter Estimation

In the parameter estimation experiments, the Jitterbug timing window w was set to 15ms, 20ms, 25ms, and 30ms. We chose 10,000, 15,000, and 20,000 as the estimation window, which was denoted as W_e. It means there are 12 estimation experimental conditions in total. We obtained estimated values of w by running estimation algorithm 50 times for each condition and the mean and standard deviation were used to evaluate the estimation performance. The results are listed in **Table 2**. From **Table 2**, it can be found that the accuracy is continually improved with the increase of estimation window. When W_e=20,000, the estimated value is very close to the real value. Since parameter estimation is an offline operation, we can start parameter estimation when gathering enough detected traffic to ensure accurate estimation.

Table 2. Estimation values under different conditions

w^*	w=15ms		w=20ms		w=25ms		w=30ms	
	mean	**stdev**	**mean**	**stdev**	**mean**	**stdev**	**mean**	**stdev**
W_e=10,000	13.24	1.99	18.79	3.12	23.21	3.33	28.54	3.03
W_e=15,000	14.19	1.68	19.59	1.5	24.56	2.07	29.46	1.45
W_e=20,000	15.31	0.32	19.93	0.46	25.01	1.34	29.75	1.17

5. Conclusion

Despite the entropy-based detection scheme is effective for most covert timing channels, the binning strategy based on training samples may cause a model mismatching case and cut down detection performance. In this paper, a new detection scheme for Jitterbug is proposed based on coefficient of variation feature, in which the fixed binning strategy works without reference samples. Furthermore, the parameter estimation method based on remarkable differential points is given. Experimental results show that our scheme maintains high detection performance for Jitterbug traffic generated by different overt traffic. Minor network jitter has little influence on detection rate while major network jitter makes Jitterbug invalid. The estimation scheme also works to a sound result by enough detected traffic. In the future, we will focus on estimating the pseudo-random sequence s_i to provide the ultimate decoding

message for data leakage forensics. Additionally, the detection and estimation problems of the Jitterbug variants, such as Liquid and Mimic, are also in the study plan.

References

[1] K. Muhammad, J. Ahmad, H. Farman, et al, "A Secure Method for Color Image Steganography using Gray-Level Modification and Multi-level Encryption," *KSII Transactions on Internet and Information Systems*, vol.9, no.5, pp.1938-1962, 2015. Article (CrossRef Link).

[2] National Computer Security Center, US DoD, "Trusted Computer System Evaluation Criteria," *Tech. Rep. DOD 5200.28-STD*, National Computer Security Center, Dec. 1985. Article (CrossRef Link).

[3] M.A. Padlipsky, D.W. Snow, and P.A. Karger, "Limitations of End-to-End encryption in Secure Computer Networks," *Tech. Rep. ESD-TR-78-158*, Mitre Corporation, Aug. 1978. Article (CrossRef Link).

[4] C.G. Girling, "Covert Channels in LAN's," *IEEE Transactions on Software Engineering*, vol.SE-13, no.2, pp.292–296, 1987. Article (CrossRef Link).

[5] V. Berk, A. Giani, and G. Cybenko, "Detection of Covert Channel Encoding in Network Packet Delays," *Technical Report TR2005-536*, Dartmouth College, Aug. 2005. Article (CrossRef Link).

[6] S. Cabuk, C. E. Brodley, and C. Shields, "IP Covert Timing Channels: Design and Detection," in *Proc. of the 11th ACM conference on Computer and communications security*, pp.178-187, 2004. Article (CrossRef Link).

[7] S. Cabuk, *Network Covert Channels: Design, Analysis, Detection, and Elimination*, Ph.D. Thesis, Purdue University, West Lafayette, USA, 2006. Article (CrossRef Link).

[8] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-based Covert timing Channels: Automated Modeling and Evasion," *RAID 2008, LNCS*, vol.5230, pp.211-230, 2008. Article (CrossRef Link).

[9] G. Liu, J. Zhai, and Y. Dai, "Network Covert timing Channel with Distribution Matching," *Telecommunication Systems: Modeling, Analysis, Design and Management*, vol.49, no.2, pp.199-205, 2012. Article (CrossRef Link).

[10] P. Peng, P. Ning, and D.S. Reeves, "On the Secrecy of Timing-Based Active Watermarking Trace-Back Techniques," *IEEE Symposium on Security and Privacy*, pp.334-349, 2006. Article (CrossRef Link).

[11] S. Gianvecchio and H. Wang, "Detecting Covert Timing Channels: An Entropy-Based Approach," *CCS'07*, Alexandria, Virginia, USA, 2007. Article (CrossRef Link).

[12] S. Gianvecchio and H. Wang, "An Entropy-Based Approach to Detecting Covert Timing Channels," *IEEE Transactions on Dependable and Secure Computing*, vol.8, no.6, pp.785-797, 2011. Article (CrossRef Link).

[13] G. Shah, A. Molina, and M. Blaze, "Keyboards and Covert Channels," in *Proc. of the 15th conference on USENIX Security Symposium*, pp.59-75, Aug. 2006. Article (CrossRef Link).

[14] R.J. Walls, K. Kothari, and M. Wright, "Liquid: A detection-resistant covert timing channel based on IPD shaping," *Computer Networks*, vol.55, no.6, pp.1217-1228, 2011. Article (CrossRef Link).

[15] K. Kothari and M. Wright, "Mimic: An active covert channel that evades regularity-based detection," *Computer Networks*, vol.57, no.3, pp.647-657, 2013. Article (CrossRef Link).

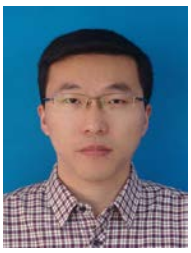
[16] "Packet traces from WIDE backbone," 2014. Article (CrossRef Link).

[17] S. Kadry, *Mathematical Formulas for Industrial and Mechanical Engineering*. Elsevier, 2014. Article (CrossRef Link).

[18] W. Mazurczyk, M. Karaś and K. Szczypiorski, "SkyDe: a Skype-based Steganographic Method," *International Journal of Computers, Communications & Control*, vol.8, no.3, pp.389-400, 2013. Article (CrossRef Link).

[19] E. Zielińska, W. Mazurczyk, and K. Szczypiorski, "Trends in steganography," *Communications of the ACM*, vol.57, no.3, pp.86-95, 2014. Article (CrossRef Link).

[20] "Linux Networking Workgroup", 2009. Article (CrossRef Link).



Hao Wang received the B.E. degree in automation from Nanjing University of Science and Technology, Nanjing, in 2008. Now, he is a Ph.D. candidate in control science and engineering at Nanjing University of Science and Technology. His research interests include network steganography, information hiding and network security.

Guangjie Liu received the B.E. degree in electrical and computer engineering, and the Ph.D. degree in control science and engineering, both from Nanjing University of Science and Technology, Nanjing, in 2002 and 2007, respectively. He is presently an Associate Professor in the School of Automation at Nanjing University of Science and Technology. His research interests are in wireless sensor network, information hiding and network security.

Jiangtao Zhai received the B.E., M.E. and Ph.D. Degrees from Nanjing University of Science and Technology in 2006, 2008 and 2013, respectively. Now, he is a lecturer in Jiangsu University of Science and Technology. He has contributed more than 15 refereed papers covering topics of network steganography.

Yuewei Dai received the B.E. and M.E. degrees in system engineering from East China Institute of Technology in 1984 and 1987, respectively, and the Ph.D. degree in control science and engineering from Nanjing University of Science and Technology in 2002. He is presently a Professor in the School of Automation at Nanjing University of Science and Technology. His research interests are in multimedia security, system engineering theory and network security.