Journal of Computers Vol. 28, No. 2, 2017, pp. 43-56
doi:10.3966/199115592017042802004

An Entropy and Hash Based Double Redundancy Watermark Model for Multi-flow Tracing

Xue-Yan Hou[1, 2], Yong-Hong Chen[1, 3,*], Hui Tian1[4], Tian Wang[1, 2], and Yi-Qiao Cai[1, 3]

[1] College of Computer Science and Technology, Huaqiao University,
Xiamen 361021, Fujian, China

[2] {hollyhxya, wsnman}@gmail.com

[3] {*djandcyh, yiqiao00}@163.com

[4] htian@hqu.edu.cn

Received 28 November 2015; Revised 10 March 2016; Accepted 07 April 2016

Abstract. Digital watermarking is a new method for network traffic tracing. Existing watermarking schemes have an obvious phenomenon that embedded watermarks have nothing to do with flows themselves. When we trace multiple flows, the watermark need to be encoded again which will reduce the capacity of watermarks greatly. Therefore, we propose an Entropy and Hash based Double Redundancy (EHDR) watermark model for multi-flow tracing. By using entropy and hash for feature extraction, data flows can be labeled. With the idea of double redundancy, EHDR model not only can reduce the overhead of time and space during the watermark detection, but it also can improve the detection efficiency effectively with the same redundancy. Moreover, this model has a good portability, to be used in many other watermarking schemes. In order to validate the efficiency of EHDR model, we introduce the real traffic from the Center for Applied Internet Data Analysis (CAIDA) dataset into the simulation environment. Experimental results show that this model is able to track the multiple flows, and improve effectively the robustness of the original watermarking scheme as well.

Keywords: double redundancy, feature extraction, flow watermark, multi-flow tracing

1 Introduction

In recent years, with the rapidly development of Internet, network security issues have become more and more serious, especially driven by economic interests, all kinds of network attacks lead to a huge economic losses to users. On the one hand, in order to avoid detecting and tracking, attacker often does not launch an attack on target host directly, but using SSH [1], IPsec Protocol [2] to login stepping stone [3-4, 19] or with the anonymous communication systems [5-6, 21], botnets [7-9] and other means to hide their real identity. It is a challenging task to trace attacks, not to mention locating the real attack source or network monitoring and management. On the other hand, due to the economic and political interests, criminals can use anonymous communication system spread gambling, pornography, violence, reactionary and other bad information. These illegal communication behaviors seriously pollute the network environment of legitimate users, and also making the behavior of forensics and censorship faces severe challenges.

Fortunately, traffic analysis techniques [27] which are practices of inferring sensitive information from communication patterns have been proposed to detect stepping-stone intrusion [10], therefore it has been a possibility to identify the sources of network attacks. As a kind of active flow analysis method, network flow watermarking technology has received wide attention in recent years. They embed the watermark information into the network data stream sent from the sender by active delaying selected packets or

[*] Corresponding Author

slight changing flow rate [22]. And then, the watermark information is decoded and recovered in the vicinity of the receiver. By comparing the similarity of the recovered watermark and original watermark to realize the correlation of data flow, and thus achieve the purpose of tracking the data stream. This technique can be widely used in anonymous user associations, tracking anonymous network telephone, locating the source of springboard attack and botmaster discovery and so on.

The watermark carrier which the existing watermark technology adopted mainly include packet payload [11], traffic rate [12-13] and packet timing [14-17]. Because of the environment of anonymous communication and encrypted traffic, it is difficult to embed watermark in the application layer of the data packet load [18]. Furthemore, traffic rate is unsuitable for tracing low data rate traffic and it is vulnerable to a Mean-Square Autocorrelation attack (MSAC) [10]. Therefore, most watermark embedding methods use packet time as the watermark carrier. For example, the inter-packet delay watermarking scheme [16] embedded watermark in the time of data packets, specifically, the time is described as packet pair. In order to achieve higher detection rate, big buffer is needed to store the large amount of data packets. This will significantly increase the packets'delay and also makes it difficult to track real-time flow [18]. What's more, there is no relationship between watermark and packet flow which is wanted to be labeled in existing watermarking algorithm. The embedded watermark does not represent the characteristics of the marked flow, this makes extra coding necessary if you want to track multiple streams. As a result, the capacity of watermark will be reduced.

Based on the above problems exist in the current watermarking schemes, in this paper, we propose a robust, secret and makable hybrid watermarking model EHDR for tracing multiple network flows. This model can make the embedded watermark and watermark carrier have some in correlation, and then the watermark itself can uniquely identify the data stream. Extra coding is not necessary any more when you track multiple streams, as a result, the watermark capacity is increased. Moreover, under the condition of same redundancy, the idea of double redundancy can greatly improve the robustness of the watermark. At the receiving end, with the idea of double redundancy, we can use a few packets to get the correct watermark, which could reduce the delay added by the watermark.

The rest of this paper is organized as follows. Section 2 summarizes previous work. Watermark model is presented in Section 3. In Section 4, we present and analyze a few key properties of the watermark model. The experimental results validating the analysis are presented in Section 5. The paper is concluded in Section 6 along with some future research directions.

2 Related Work

The passive traffic analysis (e.g., Detecting stepping stones [20]) confirm the correlation matching relationship between the various network flow by analyzing and comparing the characteristics of the traffic flow. However, this method need to capture and check all network traffic, which will significantly increase the time and space overhead of network equipment. Besides, the off-line analysis method has also led to the identification of hysteresis and poor real-time performance. It is difficult to apply in the large-scale, high bandwidth network environment, especially in the face of encrypted traffic and anonymous communication environment becomes more inadequate.

Comparing with passive traffic analysis, the active network flow watermark (ANFW) (e.g., Interval Centroid Based Spread Spectrum Watermarking scheme (ICBSSW) [10]) has more advantages on tracking and locating in encrypted traffic and anonymous communication environment. Consequently, now the research of active network flow watermark is the majority. According to the different watermark carrier, existing watermark schemes are based on three different characteristics: 1) packet payload (e.g., Sleepy Watermark Tracing (SWT) [11]; 2) traffic rate (e.g., Direct Sequence Spread Spectrum based Watermarking (DSSS-W) [13]); 3) packet timing (e.g., RAINBOW [17], Interval Centroid Based Watermarking scheme (ICBW) [14]).

Method based on packet payload (e.g., SWT [11]) require that payload can not be changed on the transmission. Furthermore, under the environment of anonymous communication and encrypted traffic, it is difficult to embed watermark in the application layer of the data packet load. As a result, there are only a few watermark scheme based on packet payload.

Approaches based on traffic rate is under the watermark W, through a certain method, slightly adjust the network flow rate in a period of time, to represent the watermark information w_i (1 or 0), so as to achieve the purpose of embedding watermark in the network flow. Yu et al. [13] brought Direct Sequence Spread Spectrum (DSSS) used in Code Division Multiple Access (CDMA) wireless communication system into the active network flow watermark and proposed DSSS-W method. However, this method has obvious drawbacks such as small watermark capacity. Besides, it is only suitable for the target flow with large velocity and long duration. And it is also vulnerable to a Mean-Square Autocorrelation attack (MSAC).

While packet-time based schemes embed watermark by changing the time of packets. Wang et al. [16] proposed Watermark Based on Inter-packet Delay (WBIPD) method and watermarks are hidden among the interval of packets. In order to achieve higher detection rate, big buffer is needed to store the large amount of data packets. This will significantly increase the packets'delay and also makes it difficult to track real-time flow. Different from the preceding method, some researchers use the idea of non-blind watermarking to operate the inter-packet delay (IPD) [23-25]. And the typical representative is RAINBOW [17] proposed by Houmansadr in 2009. But the watermark information detected from the traffic flow need to be compared with the existing IPD records in the database, which significantly increases the time overhead. It is difficult to deal with the real-time network flow, and at the same time, reduces the practicability of this method. To improve the robustness of ANFW technology to the communication network traffic noise, Wang et al. [14] proposed the Interval Centroid Based Watermarking scheme (ICBW). This method is vulnerable to a Multi-Flow Attack (MFA) because of the exposure of watermark information caused by comparing multiple streams processed by ICBW. To deal with MSAC and MFA attack at the same time, by combining the Interval Centroid Based Watermarking (ICBW) modulation approach with the Spread Spectrum (SS) watermarking coding technique, Luo et al. [10] proposed an Interval Centroid Based Spread Spectrum Watermarking scheme (ICBSSW) for efficiently tracing multiple network flows in parallel. Comparing with ICBW and DSSS-W, ICBSSW is more complex, and it needs more time and space overhead. Literature [26] is similar to ICBSSW.

However, the existing works have never correlated the data stream itself with the embedded watermark. When multiple streams are tracked, additional coding is required to distinguish different data flows. At the same time, watermark capacity is reduced. Furthermore, redundancy is used in many watermarking algorithms, how to improve the robustness under the same redundancy is a question worth studying. In the following sections we investigate these and other issues.

3 Entropy and Hash Based Double Redundancy Watermark Model

We herein propose a hybrid watermarking model EHDR aiming to let the watermark correlated with data flow as well as achieving high robustness by combining the extracting watermark approach with double redundancy technique.

3.1 EHDR Watermarking Model

Network watermarking is a technology that through a certain method to adjust some characteristics of the flow which can make it uniquely identify. Fig. 1 illustrates EHDR watermarking model. As we can see from the picture, the model is mainly composed of extracting embedded module and watermark recovery module.

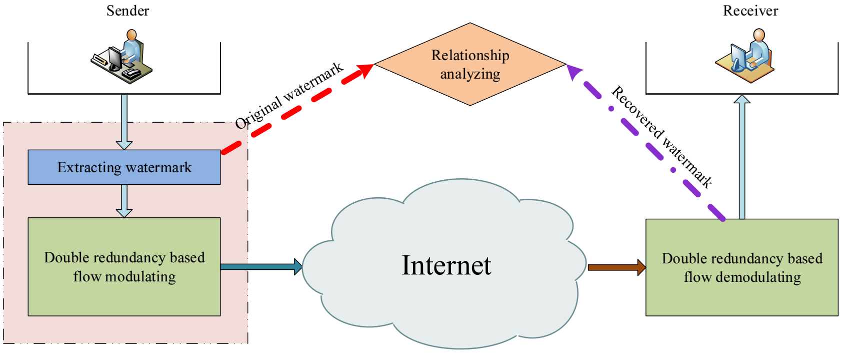

Fig. 1. EHDR watermarking mode

The watermark embedding module at the watermarker is responsible for modulating the target flow after extracting watermark from the original flow. In watermark extraction stage, the agent running on the border router can get the digital watermark which can only identify the traffic flow by analyzing the characteristics of the flow sent from the sender. In this case, the watermark will be associated with data flow. Different flows are labeled by different watermarks. When track multiple streams, extra coding is not necessary any more. And then, the watermark is embedded into the target stream by means of a certain method. In the end, the watermarked flow is sent into the network again. At the receiving end, the agent will check the existence of the watermark on receiver's inbound traffic and recover the embedded watermark further by using the parameter information shared with the watermarker. By comparing the recovery and original watermarks, the communication between sender and receiver will be easy to judge.

This method can be effectively used to track the attack source which is difficult to determine (such as SYN Flood attack) through tagging flow in the end and detecting at the other end. In order to improve the robustness of the watermark, in the watermark embedding stage, redundancy is used in most existing watermarking scheme. Double redundancy is an improvement of conventional methods in watermarking with the same redundancy rate. Experiments show that the proposed method can effectively improve the robustness of the watermark and to a certain extent improve the efficiency of watermark detection. The double redundancy can be used in many watermark schemes, such as DSSS-W, WBIPD et al. Since the operation of packet timing has the advantages in concealment and widely used, in this paper, we select watermark based inter-packet delay as the embedding method to elaborate the idea of double redundancy.

3.2 Extracting Watermark Technique

In this subsection, we will give a detailed introduction to the watermark extraction process.

Given a packet flow of duration $T_f > 0$, we want to extract l-bit watermark from the original flow. Starting from offset $o > 0$, we can choose a duration T_d and divide it into l intervals of length $\Delta t(\Delta t > 0) : I_0,...,I_i,...,I_{l-1}(i = 0,...,l-1)$. Assume there are $n_t > 0$ packets size in each interval. Let $N_{ij}(i\Delta t)(j = 1,...,n_t)$ represent the number of each size appearing in each interval. Then we have

$$N_i(i\Delta t) = \sum_{j=1}^{n_t} N_{ij}(i\Delta t) \tag{1}$$

N_i represent the total packet number of each interval. Now we are interested in the probability of each size within its interval, and we use $P_{ij}(i\Delta t)$ to represent the possibility.

$$P_{ij}(i\Delta t) = N_{ij}(i\Delta t) / N_i(i\Delta t) \tag{2}$$

At this stage, we get a distribution of probabilities $P = \{p(x_1), p(x_2),..., p(x_j),..., p(x_{n_t})\}$ with n_t elements, where $0 \le p(x_j) \le 1$ and $\sum_j p(x_j) = 1$, Therefor, the information entropy $H(X)'$ of $p(x_j)$ is

$$H(X)' = -\sum_{j=1}^{n_t} p(x_j)\log p(x_j) \tag{3}$$

The entropy obtained at this time reflects the random degree of different packet sizes in *a* time interval. The entropy need to be quantized before taking the next step. Given any $H(X)'$, we define the scalar of entropy with uniform scalar quantity $a > 0$ as the function

$$H(X) = H(X)' * a \tag{4}$$

Through the above operation, we have a entropy sequence:

$$H(X_0), H(X_1),..., H(X_i),..., H(X_{l-1}) \tag{5}$$

In order to get the final watermark, we uses the hash function $HASH()$, which is applied to the entropy sequence along with a set of secret key Key_i to compute the watermark. $HASH()$ can be any secure hash function such as $MD5$ or $SHA1$. Thus the watermark W is formed as follows:

$$W = HASH(Key_i, H(X)) \tag{6}$$

Fig. 2 shows the detailed process of the watermark extraction. From the figure we can see that the watermark extraction is mainly divided into three steps. At the first step, according to the variables o and Δt, we divide the data stream T_d into intervals: $I_0,...,I_i,...,I_{l-1}(i=0,...,l-1)$. And at the second step, calculate the probability of each size within its interval and then the information entropy is easy to be gotten. At the last step, after the quantization of the information entropy, the hash function will be used to calculate the watermark.

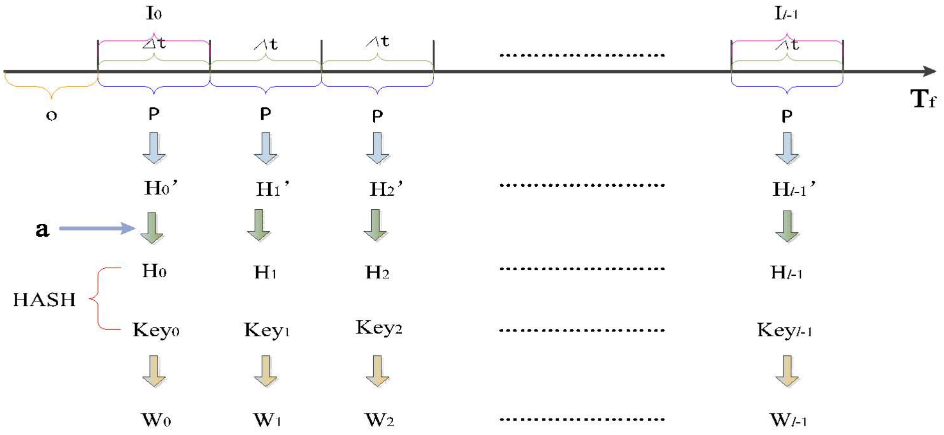

Fig. 2. Extracting watermarking technique

3.3 Double Redundancy Technique

The beginning of the double redundancy is similar to the extraction of the watermark. For example, starting from the same offset $o > 0$, and we also choose a duration T_d from the original flow T_f. But the following will be changed.

According to the number of packets, T_d is divided into $2n$ (where $n = l * r$) cells averagely: $d_0, d_1, d_2, ..., d_{2m}, ..., d_{2n-1}(m=0,1,...,n-1)$.There are $k/2$ packets in every cell. l is the total number of the embedded watermark. k is the first redundancy of watermark and r is the second. Two adjacent cells form an interval, so $2n$ cells have become n intervals and denote them as $p_0, p_1, ..., p_m, ..., p_{n-1}$. We use the following process to independently and randomly choose r intervals out of n intervals: we sequentially scan each of the n intervals and we independently and randomly choose the current interval with probability r/n. We can expect to have r intervals randomly chosen. We call the first r chosen intervals group I_0. This process cycles l times. Then we will have l groups and denote them as $I_0, I_1, ..., I_i, ..., I_{l-1}(i=0,1,...,l-1)$. Each I_i contains r intervals such as I_{ij} $(j=0,1,...,r-1)$. Fig. 3 shows the random grouping of the time intervals of a packet flow.

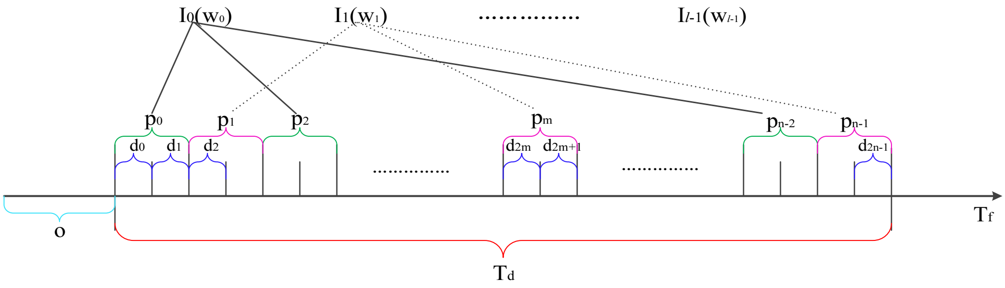

Fig. 3. Double redundancy technique

3.4 Watermarking Embedding and Detecting

It seems a little complex when there is both the extraction of the watermark and embedding. Actually, it is not. When we put them together, it will be very simple. Firstly, according to the idea of double redundancy, we divide the data stream into l groups: $I_0, I_1, ..., I_i, ..., I_{l-1} (i = 0,1,...,l-1)$. Every group I_i is equal to the interval T which we have seen in watermark extracting. And then, using the watermark extraction method calculates the corresponding watermark. At the end, embed watermarks into groups. It only need one division to realize the watermark extraction as well as completing the watermark embedding.

The following information about watermark embedding is shared between the watermark embedder and the detector. This information is assumed to be unknown to the attacker.

(1) The corresponding relation between watermark w_i and groups I_i.

(2) The first and second redundancy of watermark k, r.

(3) The number of watermark bits l.

(4) The offset o.

In the stage of watermark detection, according to the information shared by the sender, the division of the data stream will be completed. Calculate the watermark respectively by the detecting method of inter-packet delay in every interval p_m. Since group I_i contains r watermarks, the watermark appears in the most times is the finally watermark that group I_i takes along.

4 Theoretical Analysis

4.1 Multi-streaming Tracking

Yu et al. (2007) have confirmed that it is necessary to trace multiple flows simultaneously [13]. There must be a lot of other traffic flows with target stream in the actual network environment. When they share a link or router, the flows will interact with each other. As shown in Fig. 4, there are five flows (flow 1, flow 2, flow 3, flow 4, flow 5) entering a mix. The flow 1 and flow 3 are integrated in the output link 2 of the mix, at the same time, flow 2 and flow 4 are integrated in the output link 3 of the mix. How to reduce the influence and ensure the watermark detection efficiency becomes very important.

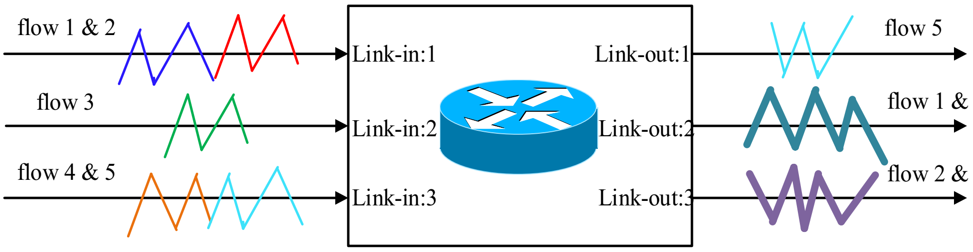

Fig. 4 Tracing multiple flows

In the EHDR model, we mark the data stream using the characteristics of itself and also use low correlate secret key to reduce the influence of interaction. Each flow has different distribution of packet size. The entropy sequences obtained by statistical calculation are also different. In order to reduce the impact of interaction, low correlated secret key is used in the extraction process of watermark. In addition, we embed the watermark into the traffic flow. So, repacketization has no effect on the original watermark in the process of transmission. Just the opposite, repacketization will make it impossible for attacker to obtain the watermark even they know the extraction method. And attacker also can not forge the sender to tag other flows. On output link 2, a sniffer can apply the detection techniques discussed previously to identify each flow.

4.2 The Robustness of EHDR

In realistic settings, the traffic is often attacked by the time delay. Increased packet interval will destroy the integrity of the watermark. It is important to guarantee the robustness of watermark against the time delay added by attacker.

We exploit the assumptions that

(a) The attacker does not know where the watermark bits will be embedded.

(b) The random delays added by the attacker are independent and identically distributed (iid).

In a data flow, we assume that there are n packets for embedding watermarks and each watermark is carried by m packets ($m = n/l$, l is the number of watermarks). New, we have two method for embedding: one is that m packets are continuous, which is the method used by most of the watermark; The other is double redundancy method we have introduced previously. m packets continues to be divided into r intervals (r is an odd number), and there are k packets in each interval. In other word, $m = r * k$. A watermark is embedded into the r intervals respectively. For single watermark, we assume the watermark will be destroyed when attack happens.

Suppose that the probability of each packet is attacked by p , and then $1-p$ is the probability of packet to be safe. When the packets are continuous, if the watermark is to be detected correctly, all packets must be safe. In this case, the probability P_c of watermark detected is

$$P_c = (1-p)^m \quad \textbf{(7)}$$

In the EHDR model, we assume the probability of each interval is detected by q . Since there are k packets in each interval, the probability q can be describe as

$$q = (1-p)^k \quad \textbf{(8)}$$

Since a watermark is independently carried by r intervals, as long as more than half of the interval have been detected correctly, the watermark will be extracted successfully. Therefore, the probability P_d of watermark extracted must contain all the conditions that the watermark can be detected correctly. So, the P_d is

$$P_d = P_{(\left\lfloor \frac{r}{2} \right\rfloor+1)} + P_{(\left\lfloor \frac{r}{2} \right\rfloor+2)} + \ldots + P_{(i)} + \ldots + P_{(r)} \quad \textbf{(9)}$$

According to the permutation and combination, when i intervals are detected, the probability of P_i can be described as

$$P_i = C_r^i q^i (1-q)^{r-i} \qquad i = \left[(\left\lfloor \frac{r}{2} \right\rfloor + 1), (\left\lfloor \frac{r}{2} \right\rfloor + 2), \ldots, r \right] \quad \textbf{(10)}$$

When $i = r$, which means all the packets should be safe. So the probability of $P_{(r)}$ can be described as

$$P_{(r)} = q^r = ((1-p)^k)^r = (1-p)^m \quad \textbf{(11)}$$

When all the probabilities $P_i(i = (\left\lfloor \frac{r}{2} \right\rfloor + 1), (\left\lfloor \frac{r}{2} \right\rfloor + 2), \ldots, r-1)$ are zero, $P_d = P_{(r)}$. And only then, $P_d = P_c$. Therefore, under normal circumstances, we have

$$P_d \geq P_c \quad \textbf{(12)}$$

The first method is more vulnerable to attack.

4.3 Time and Space Overhead of EHDR

The embedding method of EHDR is in the case of the same redundancy, divided into to a number of small intervals and randomly distributed in the whole data stream. Each of the small space is embedded the same watermark, so each watermark has r copies. In the detecting phase, the maximum number of occurrence is the final value of the watermark. This method can effectively reduce the time and space overhead.

From the section 4.2, we know that there are n packets for embedding watermarks and each watermark is carried by m packets. At the method of EHDR model, m is subdivided. During each interval, the probability of correctly detecting the watermark is p_k, then, the probability of loss is $1-p_k$. According to this model, when we extracted the watermark successfully with the least packets, there is only one kind of situation that the former $(\left\lfloor \frac{r}{2} \right\rfloor+1)$ intervals are correctly detected. Therefore, the probability $P_{(\left\lfloor \frac{r}{2} \right\rfloor+1)}$ of this situation is

$$P_{(\left\lfloor \frac{r}{2} \right\rfloor+1)} = (p_k)^{(\left\lfloor \frac{r}{2} \right\rfloor+1)} \qquad \textbf{(13)}$$

r is watermark number of copy,and k is the first redundancy of watermark. We use the symbol N to denote the number of packets used. In this case, the packets used $N_{(\left\lfloor \frac{r}{2} \right\rfloor+1)}$ is

$$N_{(\left\lfloor \frac{r}{2} \right\rfloor+1)} = k*(\left\lfloor \frac{r}{2} \right\rfloor+1) \qquad \textbf{(14)}$$

At worst, we need checking all the packets to extract the watermark. In this condition, the last interval must be detected correctly and there are half of intervals have been extracted inaccurately before. Using the idea of permutation and combination, the probability $P_{(r)}$ of this situation is easy to get.

$$P_{(r)} = p_k * C_{r-1}^{\left\lfloor \frac{r}{2} \right\rfloor}(p_k)^{\left\lfloor \frac{r}{2} \right\rfloor}(1-p_k)^{\left\lfloor \frac{r}{2} \right\rfloor} \qquad \textbf{(15)}$$

At the same time, the number of packet used $N_{(r)}$ is

$$N_{(r)} = k*r = m \qquad \textbf{(16)}$$

Now, we know the number of packet used in detecting and the probability, according to the formula of expected value, the expectation of packets $E(N)$ which need to be used in detecting is

$$\begin{aligned} E(N) &= P_{(\left\lfloor \frac{r}{2} \right\rfloor+1)} *k*(\left\lfloor \frac{r}{2} \right\rfloor+1) + P_{(\left\lfloor \frac{r}{2} \right\rfloor+2)} *k*(\left\lfloor \frac{r}{2} \right\rfloor+2) + \ldots + P_{(i)}*k*i + \ldots + P_{(r)}*k*r \\ &= \sum_{i=(\left\lfloor \frac{r}{2} \right\rfloor+1)}^{r} P_{(i)}*k*i \end{aligned} \qquad \textbf{(17)}$$

To detect a watermark correctly, $k*(\left\lfloor \frac{r}{2} \right\rfloor+1)$ packets are needed at best and m packets are needed at worst. So

$$k*(\left\lfloor \frac{r}{2} \right\rfloor+1) \le \sum_{i=(\left\lfloor \frac{r}{2} \right\rfloor+1)}^{r} P_{(i)}*k*i \le m \qquad \textbf{(18)}$$

Since l is the total number of the embedded watermarks, and m is the number of packet used in single redundancy (m packets are also needed at the worst situation in EHDR model). When all watermarks are detected, the packets saved N_s in EHDR model is easy to calculate.

$$N_s = l*m - l*\sum_{i=(\left\lfloor \frac{r}{2} \right\rfloor+1)}^{r} P_{(i)}*k*i \qquad \textbf{(19)}$$

According to the formula (18), we are easy to get the conclusion

$$N_s \geq 0 \qquad \textbf{(20)}$$

5 Experiment

We have systematically developed the EHDR model in previous sections. In this section, we use ns-2 simulations, CAIDA dataset and watermark based inter-packet delay as the embedding method to investigate the effectiveness of this model which can improve the robustness of the watermark.

5.1 Simulation Setup

Based on EHDR model, we implemented a simulate environment consisting of several hosts as shown in Fig. 5 to evaluate the effectiveness of EHDR. Fig. 6 gives a abstract model of experiment environment. CAIDA dataset contains most of the data flows on March 20, 2014 in Chicago from 1 pm to 2 pm, and general characteristics of network traffic can be very good response. To simulate a realistic environment, we extract some data streams from the CAIDA dataset as the target flows into the experimental environment. In the watermark modulating phase, firstly, we use the extraction method discussed in Section 3.2 to extract *96* bits watermark from target flow. Then, the EHDR model discussed in Section 3.3 will be combined with inter-packet delay watermark embedding method to embed those watermarks into the target flow. The watermarked flow is perturbed with random delays at the Perturber to simulate the timing perturbation. Since there are many other interference flows (some flows are imported from the CAIDA dataset, and some flows are produced by ns-2 simulator) in experimental environment, they are sharing the links with the target flow at the Mixer. Upon receiving the watermarked flow, the Detector extracted the watermark from the received flow, which will be compared against the original watermark for a match later. In order to achieve better experimental effect, sometimes more than thousands of packets are used to embed watermarks. Except where it is explicitly stated, we focus on the single target flow case in our discussion.

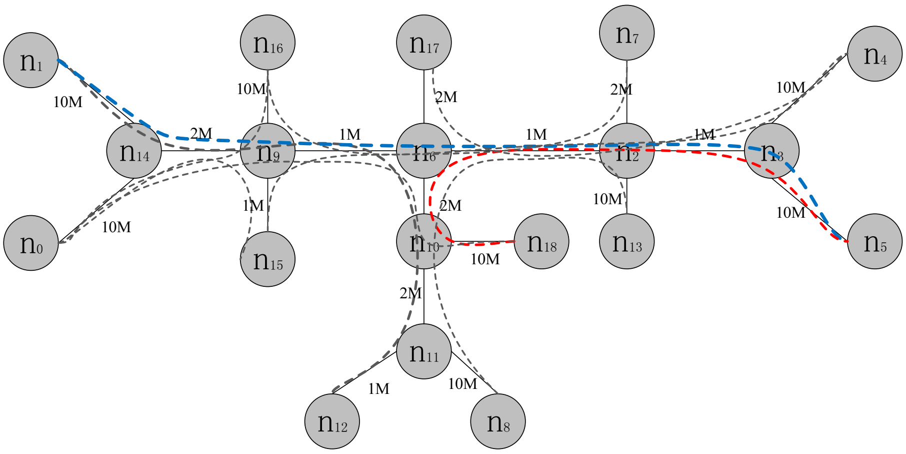

Fig. 5. Topology in ns-2 simulations

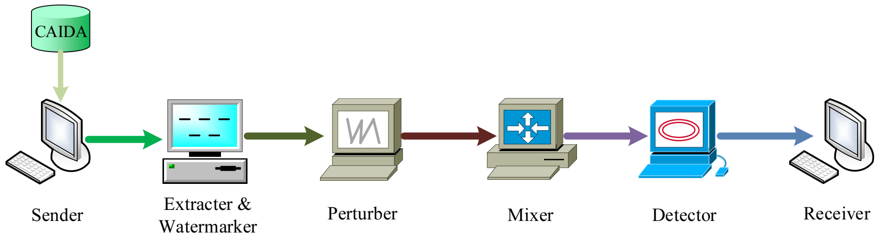

Fig. 6. Abstract model of experiment environment

5.2 Robustness of EHDR Against Interference

As we have already declared in the previous, in this experiment, we use the inter-packet delay as the method of embedding watermarks. As the second redundancy of watermark, when $r=1$, it is the single redundancy we often use in many watermark scheme and also the comparing object we need in this experiment. After the completion of the segmentation and extraction of data stream, we will prove that the double redundancy theory can effectively improve the robustness of the embedded watermark.

In order to prove the robustness of EHDR against interference, in this experiment, we use ns-2 to generate random time perturbations, and using packet flow produced by other hosts interfere the target stream. Because of the large number of watermarks and the limitation of the target flow, in order to guarantee the watermarks can be embedded into the stream entirely, let $r=3, r=5, r=7$ to verify the validity of the EHDR model in improving the robustness of the embedded watermark, and as the comparative object when $r=1$. The result in Fig. 7 shows that the detection rate of EHDR increases as the average number of packets increases. In comparison, the EHDR model achieves a higher detection rate than $r=1$ for a given number of packets used. In this model, the bigger the r value, the higher the true positive rate. From the diagram we are easy to see that we achieve the highest detection rate when $r=7$. Compared with $r=1$, EHDR requires no more than 8000 packets for a 100% detection rate when $r=7$, while the single redundancy requires more than 12000 packets. In Fig. 8, this argument is more persuasive. When the required detection rate is more accurate, more data packets are required for single redundancy. From Fig. 8 we can see, when accuracy is more than 0.76, with the increase of the detection rate, the number of packets between single redundancy and EHDR is showing a sharp upward trend.

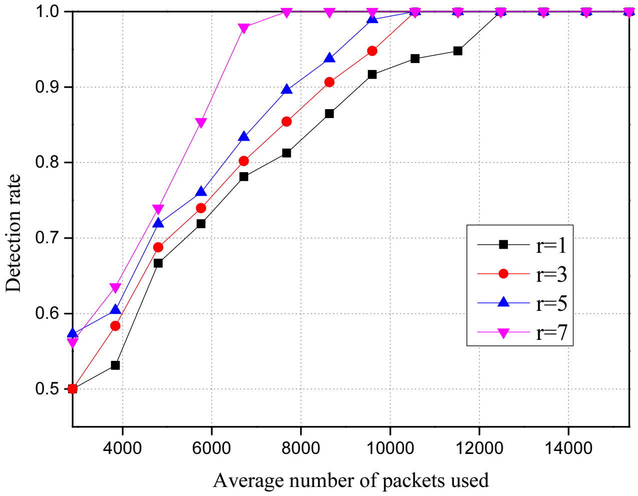

Fig. 7. Detection rate comparison of the different second redundancy of watermark

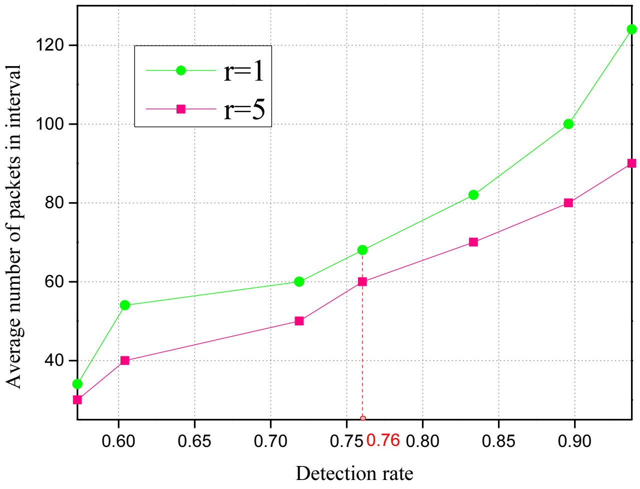

Fig. 8. Comparison of packets between r=1 and EHDR

5.3 Robustness of EHDR Against Different Quantization Step

Different quantization steps have an effect on the detection rate for the watermark scheme based inter-packet delay. To demonstrate the robustness of EHDR against different quantization, which was modeled using uniformly random quantization steps with a maximum valus of S ranging from 0 to 12000 us, we measured the detection rates of EHDR and single redundancy, respectively, for each different quantization step S as shown in Fig. 9. When the quantization step size is less than 2000 us, the true positive rate of the three was on the rise, and the difference was not big. The detection rate of $r = 5$ is only a little higher than that of single redundancy when the quantization step size is 2000 us. By contrast, the detection rate of 7 is much higher. The growth of accuracy when the second redundancy is 7 was relatively slow in the quantization steps from 2000 us to 4000 us. As long as the step size is over 6000 us, the detection rate of EHDR is much higher than the single redundancy's. And the accuracy of 7 is the highest, which highly matches the analytical results in Section 4.2. If the step size is too big, it will also reduce the detection rate due to the disruption of the original packets' order.

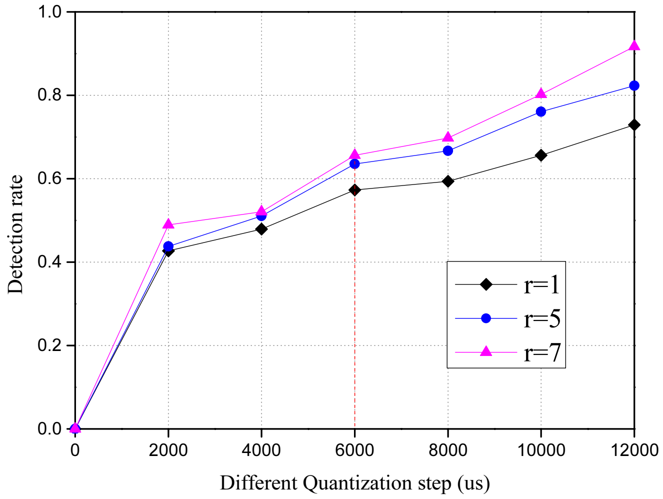

Fig. 9. Detection rate comparison of different quantization steps

5.4 Effectiveness of Tracing Multi-flow

To demonstrate the effectiveness of EHDR model marking technique at tracing multiple flows, we add another target flow of interest in Fig. 5. The two target flows are: one from n_1 to n_5 and the other from n_{18} to n_5. Both flows traverse mixes n_6, n_2 and n_3. To interfere with these two flows, we use ns-2 to generate random time perturbation and other hosts to produce packet flows.

In this set of simulations, let the second redundancy r is 5. Due to the fact that the small step size makes the watermarks weak to resist the interference, the big step size may also cause the failure of the watermarks embedding. But in order to ensure successful watermarks embedding and good anti-interference ability of the target flow, according to the theoretical calculation and observation experience, we make the s value $10000\,\text{us}$. Fig. 10 shows the detection rate for both flows with time perturbation and interferential flows. From this figure, we observe that the EHDR-based flow marking technique can effectively correlate senders and receivers of both flows. The detection rate of flow from n_{18} to n_5 is a little higher than the other flow. The reason is that flow from n_1 to n_5 received more interference. But both flow can achieve 100% detection rate when the redundancy reaches a certain degree. Therefore, we conclude that EHDR based framework can efficiently trace multiple flows simultaneously.

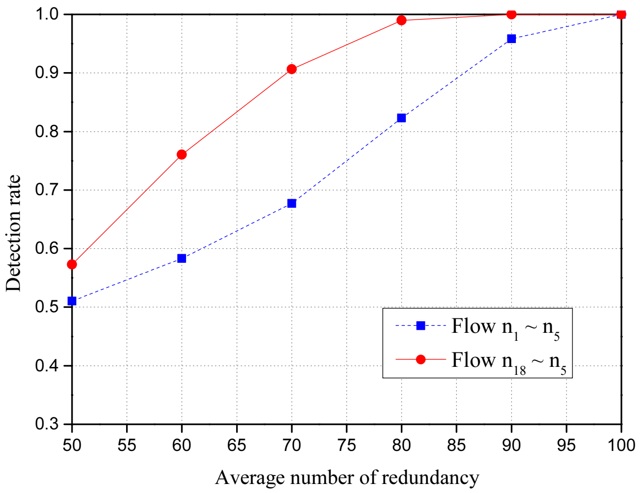

Fig. 10. Detection rate v.s. redundancy for tracing multiple flows

6 Conclusion and Future Work

Watermarking is a promising approach to tracing flows. The existing watermarking method has an obvious phenomenon that the embedded watermark has nothing to do with the flow itself. This leads to two-pass encoding when we trace multiple flows, which will also reduce the capacity of watermarks. The paper presents a theoretical EHDR model which combines the embedded watermarks with the characteristics of data stream. In this model, with the idea of double redundancy, we can effectively improve the robustness of the embedded watermark against time perturbation and traffic interference. Both our analytical and empirical results show that the model can not only reduce the overhead of time and space during the watermark detection, but also can effectively improve the detection efficiency in the case of same redundancy. Experiments also prove that EHDR can achieve high efficiency when tracing multiple flows simultaneously. Furthermore, this model approach can be applied to other watermarking schemes for effective and efficient multi-flow traceback. Thus our model is of practical value in optimizing the overall effectiveness of watermark in real world situations.

We import the actual traffic flows into the simulation environment in this experiment. Our future work will first verify the validity of this model in realistic settings, and then try to bring it into intrusion detection system.

Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC) under Grant Nos. 61370007, U1405254, 61572206, Fujian Provincial Natural Science Foundation of China under Grant No. 2013J01241, and Program for New Century Excellent Talents of Fujian Provincial under Grant No. 2014FJ-NCET-ZR06.

References

[1] Y. Ylonrn, The secure shell (SSH) protocol architecture [EB/OL]. <http://www.ietf.org/rfc/rfc4251.txt>, 2006 (accessed 15.04.28).

[2] IPSEC working Group, IP security protocol (IPSec) [EB/OL]. <http://datatracker.ietf.org/wg/ipsec/>, 1995 (accessed 15.01.07).

[3] S. Robert, C. Jie, J. Ping, C. Weifeng, A survey of research in stepping-stone detection, International Journal of Electronic Commerce Studies 2(2)(2011) 103-126.

[4] T. He, L. Tong, Detecting encrypted stepping stone connections, IEEE Transactions on Signal Processing 55(4)(2007) 1612-1623.

[5] R. Dinglediner, N. Mathewson, P. Syverson, Tor: the second-generation onion router, in: Proc. the 13th USENIX Security Symposium, 2004.

[6] M.J. Freedman, R. Morris, Tarzan: a peer-to-peer anonymizing network layer, in: Proc. the 9th ACM Conference on Computer and Communications Security, 2002.

[7] J. Jiang, J.-W. Zhuge, H.X. Duan, J.-P. Wu, Research on botnet mechanisms and defenses, Journal of Software 23(1)(2012) 82-96.

[8] E. Passerini, R. Paleari, L. Martignoni, D. Bruschi, FluXOR: detecting and monitoring fast-flux service networks, in: Proc. the 5th Detection of Intrusions and Malware, and Vulnerability Assessment, 2008.

[9] Z. Holz, C. Gorecki, K. Rieck, F.C. Freiling, Measuring and detecting fast-flux service networks, in: Proc. the 15th Network and Distributed System Security Symposium (NDSS'08), 2008.

[10] J.Z. Luo, X.G. Wang, M. Yang, An interval centroid based spread spectrum watermarking scheme for multi-flow traceback, Journal of Network and Computer Applications 35(1)(2012) 60-71.

[11] X.Y. Wang, D.S. Reeves, S.F. Wu, J. Yuill, Sleepy watermark tracing: An active network-based intrusion response framework, in: Proc. the 16th Int'l Conf. on Information Security (IFIP/Sec), 2001.

[12] X.W. Fu, Y. Zhu, B. Graham, R. Bettati, W. Zhao, On flow marking attacks in wireless anonymous communication networks, in: Proc. the 25th IEEE Int'l Conf. on Distributed Computing Systems (ICDCS), 2005.

[13] W. Yu, X.W. Fu, S. Graham, D. Xuan, W. Zhao, DSSS-Based flow marking technique for invisible traceback, in: Proc. the 2007 IEEE Symp. on Security and Privacy (SP), 2007.

[14] X.Y. Wang, S.P. Chen, S. Jajodia, Network flow watermarking attack on low-latency anonymous communication systems, in: Proc. the 2007 IEEE Symp. on Security and Privacy (SP), 2007.

[15]Y.J. Pyun, Y.H. Park, X.Y. Wang, D.S. Reeves, P. Ning, Tracing traffic through intermediate hosts that repacketize flows, in: Proc. the 26th IEEE Int'l Conf. on Computer Communications (Infocom), 2007.

[16] X.Y. Wang, D.S. Reeves, Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays, in: Proc. the 10th ACM Conf. on Computer and Communications Security (CCS), 2003.

[17] A. Houmansadr, N. Kiyavash, N. Borisov, RAINBOW: A robust and invisible non-blind watermark for network flows, in: Proc. the 16th Annual Network \& Distributed System Security Symp (NDSS), 2009.

[18] X.J. Guo, G. Cheng, C.G. Zhu, D.T. Truong, A.P. Zhou, Progress in research on active network flow watermark, Journal of Communication 35(7)(2014) 178-192.

[19] A. Blum, D. Song, S. Venkataraman, Detection of interactive stepping stones: algorithms and confidence bounds, in: Proc. the 7th International Symposium on Recent Advances in Intrusion Detection, 2004.

[20] Y. Zhang, V. Paxson, Detecting stepping stones, in: Proc. the 9th USENIX Security Symposium, 2000.

[21] M.K. Reter, A.D. Rubin, Anonymous web transactions with crowds, Communications of the ACM 42(2)(1999) 32-38.

[22] L.C. Zhang, Z.X. Wang, J. Xu, A watermark technology based on packet sequence rearrangement, Journal of Software 22(2)(2011) 17-26.

[23] A. Houmansadr, N. Kiyavashy, N. Borisov, Rainbow: a robust and invisible non-blind watermark for network flows, in: Proc. the 16th Network and Distributed System Security Symposium (NDSS’09), 2009.

[24] L.C. Zhang, Z.X. Wang, J. Xu, A novel invisible and private flow watermarking scheme, Journal of Applied Sciences 30(5)(2012) 524-530.

[25] L. Zhang, J.Z. Luo, M. Yang, Orthogonal flow characteristics based multi-dimensional flow watermarking technique, in: Proc. the 2010 China Communication Security Symposium, 2010.

[26] L. Zhang, J.Z. Luo, M. Yang, An improved DSSS-based flow marking technique for anonymous communication traceback, in: Proc. the 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (UIC-ATC'09), 2009.

[27] R. Dingledine, N. Mathewson, Technical report. <http://www.freehaven.net/tor/cvs/doc/tor-spec.txt>, 2004 (accessed 14.12.26).