Available online at www.ijpe-online.com

**International Journal of Performability Engineering**

vol. 16, no. 2, February 2020, pp. 265-283
DOI: 10.23940/ijpe.20.02.p10.265283

# Repeatedly Coding Inter-Packet Delay for Tracking Down Network Attacks

Lian Yu[a,*], Lei Zhang[a], Cong Tan[a], Bei Zhao[b], Chen Zhang[b], and Lijun Liu[b]

[a]*School of Software and Microelectronics, Peking University, Beijing, 102600, China*
[b]*Design Institute, China Mobile Group, Beijing, 100080, China*

**Abstract**

Attacks against Internet service provider (ISP) networks will inevitably lead to huge social and economic losses. As an active traffic analysis method, network flow watermarking can effectively track attackers with high accuracy and a low false rate. Among them, inter-packet delay (IPD) embeds and extracts watermarks relatively easily and effectively, and it has attracted much attention. However, the performance of IPD is badly affected when networks have perturbations with high packet loss rate or packet splitting. This paper provides an approach to improve the robustness of IPD by repeatedly coding the inter-packet delay (RCIPD), which can smoothly handle situations with packet splitting and merging. This paper proposes applying the Viterbi algorithm to obtain the convolutional code of a watermark such that the impact of network perturbation on the watermark can be worked off; applying the harmony schema, which controls the rhythm and embeds RCIPD bits into network flow, to improve the invisibility of watermarking; and applying K-means to identify dynamically bits of the watermark that may change the intervals due to the latency of networks. A cyclic-similarity algorithm (CSA) is designed to separate the repeated coding and eventually obtain the watermark. Experiments are carried out to compare RCIPD with other three schemas. The results show that the proposed approach is more robust, especially in the case of packet splitting.

*Keywords*: network attack tracking; network flow watermarking; repeatedly coding inter-packet Delay; Viterbi coding; K-means; harmony schema

(Submitted on November 15, 2019; Revised on January 10, 2020; Accepted on February 10, 2020)

(Special Section of the 6th International Conference on Dependable Systems and Their Applications)



## 1. Introduction

The infrastructure of an Internet service provider (ISP) sustains the continuous development of Internet technology, and its security and stability play a key role for the entire network. Once ISP infrastructures are attacked, the stability and security of the entire network will suffer serious threats, i.e., if the infrastructure becomes unsafe, there will no longer be a secure Internet environment.

Traditional methods of defending network attacks are passive and inefficient, such as firewalls, input debugging, and logging. Passive tracking uses various techniques to trace the attacker after detecting the attack. The main methods of accountability after an attack involve analyzing the recorded logs. This kind of approach consumes a large amount of resources. If an attacker uses a stepping-stone, it will make the tracking of the source more difficult. Active defenses proactively analyze the characteristics of network traffic, detect attacks during the attack, and take defensive actions or launch tracking to trace back to attackers. This method is more efficient and has gradually evolved in recent years, and it has become the focus of researchers.

The most popular method of active defending is network flow watermarking, which embeds the specific marking into the data flow from the attacked host to the attackers and inspects whether the network flow contains marking at a detector. The

* Corresponding author.
*E-mail address*: lianyu@ss.pku.edu.cn

tracking method includes two types. One is to write specific data or tags directly into the data flow from the attacked host to the attackers, detect whether the forwarded packets contain specific tags at the forwarding intermediate nodes, and then recursively rebuild the attack path down to the source of the attack. This method is accurate and efficient, but if attackers discover the existence of these special tags, they may take corresponding actions (such as modifying and disturbing these special tags) to prevent being tracked.

The other way to encode and embed specific tags with pre-defined schema into the data flow from the attacked host to the attacker is to modulate some traffic characteristics of the data flow, such as interval time, packet size, and transmission frequency. In the forwarding intermediated node of an ISP network, the features in the forwarded data flow are analyzed by extracting and decoding the embedded data, matching the characteristics of the data to determine whether it is a node on the attack path, and finally obtaining the entire attack path.

Figure 1 shows the main steps to perform watermarking for tracking down network attacks. With all the detecting results from these distributed intermediate nodes, a control center can rebuild the entire attack path.

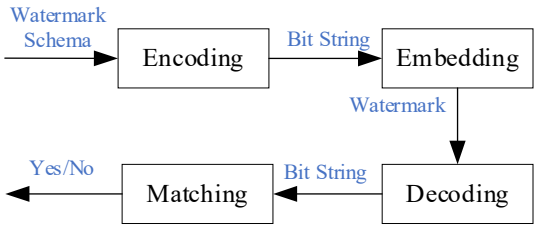


Figure 1. Main steps in watermarking

The second method is relatively secretive and makes it difficult for an attacker to detect the existence of a tracking program. However, the latter still has some issues that need to be addressed, as follows:

- Perturbations in the network: Some perturbations in the network will affect the characteristics of the data flow, and the tags embedded by modulating those characteristics will also be affected.
- Merging and splitting of packets: Intermediate forwarding nodes in the ISP network may have problems, such as packet merging and packet splitting. When a packet is too large, the forwarded router may split it into smaller packets and then forward them. When some packets are too small, they may be merged into one larger packet and then forwarded. Merging and splitting of packets will most likely impact embedded watermarks.
- Cascading of error sequences: When a previous tag has an error, the subsequent tags will also be affected. This leads to the failure of demodulating of watermarking eventually at the forwarding intermediated node in an ISP network.
- Lack of invisibility: A typical IPD embeds watermarks into the data flow by modulating the time interval of packets in the network data flow. Therefore, the time characteristics of the modulated data flow will be significantly different from the time characteristics of the non-modulated data flow. When attackers detect the existence of differences or detect these differences through third-party tools, they will take effective actions to disrupt watermarks and prevent tracking.

This paper proposes a time-based network flow watermarking for attack-tracking systems: repeatedly coding inter-packet delay (RCIPD), which is an improved version of IPD. Several improvements are provided to deal with some of the issues in the time-based network flow watermarking to enhance the tracking efficiency and accuracy of the whole system:

- In the encoding, the original watermark is been repeated n times, and the convolutional encoding algorithm [1] is applied. The repeated watermark plays an import role in resisting packet splitting and merging. The encoding algorithm can improve the accuracy with high fault-tolerance during the network transmission.
- When embedding the watermark into a packets flow, the strategy, called the harmony schema, is proposed. It ensures that the mathematical expectation of the time-delay of watermarking sequences is the same as the other random sequences (i.e., they are statistically indistinguishable) [1-3], to improve the invisibility of watermarks.
- During decoding, the K-means algorithm is used to re-determine the classification of bit 0 or bit 1 in a bit sequence that is transmitted through the network. The cyclic-similarity algorithm (CSA) is proposed to delimit a bit subsequence from a network data flow most likely to the original bit sequence of watermarking. The Viterbi algorithm [2] is applied to decode the bit sequence to obtain the extracted watermark.
- The Hamming distance algorithm [2] is used to calculate the similarity between the original watermark and the extracted watermark.
- The rest of the paper is organized as follows. Section II presents related work. Section III introduces the overall architecture and the process of the proposed approach. Section IV describes the schema design and theoretical principles. Section V demonstrates the experimental results and analysis. Section VI discusses the proposed approach in this paper and the problems to be improved. Section VII concludes the paper.

## 2. Related Works

The tracking of network attacks attracts many researchers and security vendors to explore various techniques and applications [4-6]. This section presents the main methods for network flow watermarking tracking, including log-based attack tracking, modifying network transmission data based on attack tracking, data flow matching-based attack tracking, and network flow watermarking-based attack tracking.

*Log-based Attack Tracking*

Log-based tracking [7,8] is a relatively mature and widely used technique in tracking network attacks. Its main participants include the attacker host, the victim host, and the routers on the attack path. Each participant records the data flow going through itself and restores the attack path by querying the recorded data at the time of tracking to find the attackers.

Log-based tracking techniques consist of two parts: logging and log query. In the log recording, each participant needs to record the key data related to the tracking to its own database, including the source address, source port, destination address, and destination port. Because each participant has a limited ability to store data, the data is generally sent to the central database of the control for unified processing and storage. When an attack occurs, the tracker starts from the victim and reversely queries the records of each participant. If the participant's records contain the queried data, the whole attack path containing the participant and the attacking host can be recovered. The technique is simple in principle and has been widely used by combining techniques, such as big data and cloud computing. However, it requires router support and a large amount of storage space, and it is relatively slow.

*Attack Tracking based on Modifying Network Transmission Data*

This method achieves the purpose of attack source tracing by making certain modifications to the data transmitted in the network. For example, the router randomly changes original value in some fields of IP packet header into the current router tags, such as the IP address or part of the IP address, according to a certain probability, and the attack path is recovered based on these modified fields. The recovery of an attack path is usually carried out at the victim side, based on whether the network packets contain the modified fields of the routers. At present, the tracking schemes generated based on this idea include the packet tracking method [9,10], probability packet tracking method [11-13], and route label tracking method [14].

*Tracking based on Data Flow Matching*

Different from the tracking techniques presented above, the principle of data flow matching tracking is to determine whether the incoming and outgoing data flows are the same by checking the inflow and outflow of the network nodes, determining the degree of correlation between the inflow and outflow traffic, and reconstructing the transmission path of the data flow in the network. According to the different data flow association objects, there are currently three types of tracking techniques based on data flow matching: data flow packet header based on features, data flow time features, and data flow content based on features.

- Based on packet header matching [15,16]: By checking the message header of the data flow in the network node, it is determined whether the data packet flowing into the node has an association relationship with the data packet flowing out of the node, thereby determining the direction of the data flow.
- Based on data flow time feature matching [17-19]: Whether or not the data flow is the same is determined by checking if the data flow has a causal relationship on different temporal characteristics. Since the matching object of the technique is based on a temporal feature, it is effective for encrypted data. It has been actively applied in scenarios, such as botnets and anonymous network traceability.
- Based on feature matching of data flow contents [20]: The correlation of the data flow entering and leaving the node is determined by checking the matching degree of the data flow contents in the network node. However, this technique currently loses usability for encrypted data flows.

*Attack Tracking based on Network Flow Watermarking*

The attack source tracing technique based on network flow watermarking is an improvement and extension of the above data flow matching [21,22]. Passive data flow matching is prone to generate a high false positive rate, has a long observation time, and is difficult to resist signal interference or even malicious attacks during data transmission. The traffic correlation based on network flow watermarking uses the active traffic analysis technique [23]. The principle is to determine whether the receiving end is on the communication path by actively embedding a digital watermark in the sender's network flow and

detecting whether the digital watermark appears at the receiving end. Through analysis, the complete attack path is reconstructed.

According to different flow watermark carriers, three main types of network flow watermarking are developed, namely packet payload, packet rate, and packet-time flow watermarking.

(1) Packet load-based flow watermarking [24,25]: The principle is to achieve the purpose of embedding digital watermarks by changing the load of the data packet in the data flow (such as the size or content of the data packet). A typical representative is the sleep watermark tracking method proposed by Wang et al. However, this method is difficult to apply to encrypted data traffic due to the exact modification of the contents of a packet.

(2) Packet rate-based flow watermarking [26,27]: The principle is to modulate the rate of the data flow to embed the watermark. The watermark is usually embedded at the sender by actively affecting the network flow rate.

(3) Packet time-based flow watermarking: The principle is to embed watermarks by adjusting the temporal characteristics of the data packets. Currently, there are three main types: time interval [28-33], interval arrival time [34], and the center of gravity of the interval [35-37]. Time-based flow watermarking refers to controlling the number of packets in a fixed time interval by dividing the data flow into fixed-length time intervals and adjusting the packet time. The flow watermark is based on the interval arrival delay. By modulating the time interval of each data packet in a data flow, the watermark can be effectively embedded. The flow center watermark is based on the center of gravity of the interval. Since the center of gravity of the interval represents the equilibrium point in the interval period, the modulation can effectively correlate the data flow.

IPD (inter-packet delay) is a classical time-based flow watermarking. The interval times between packets contain the information that are used to mark the data flow. It modulates the interval time between packets to embed the original watermark into the packet flow to the attacker, and the detecting node will match the watermark extracted from the data flow going through the node to determine whether the node is in the attacking path. RCIPD is an improved version of IPD that is more accurate, robust, and invisible.

## 3. Overall Architecture and Process

The paper assumes that the network flow watermarking system is deployed in a backbone network, such as the ISP network. The whole network is divided into two layers: a core layer and a large-area layer. The core layer is mainly composed of core nodes. The large-area layer that divides the whole network into several large areas is mainly composed of tandem nodes and service nodes. The communications between different large areas must go through the core nodes. The scenario of an attack in the network is shown in Figure 2.

The attacker launches an attack through a service node, and the attacked host communicates with the attacker through the ISP network. A honeynet with multiple honey-pots is deployed to attract the attacks from the attacker. After the attacker launches an attack, the honeynet manager collects the warnings of potential attacks from honey pots on the Web server, application server, and database server, and then activates the watermarking trigger that initiates an attack tracking. There will be a network flow from the attacked host to the attacker, and it is possible to embed a watermark into the data flow. Every node in the ISP network can undertake tracking as a detector of watermarking, and it will detect every data packet through the node. The detecting results will be reported to a control center of the whole system, which can rebuild the entire attacking path with these detecting reports from distributed detectors.

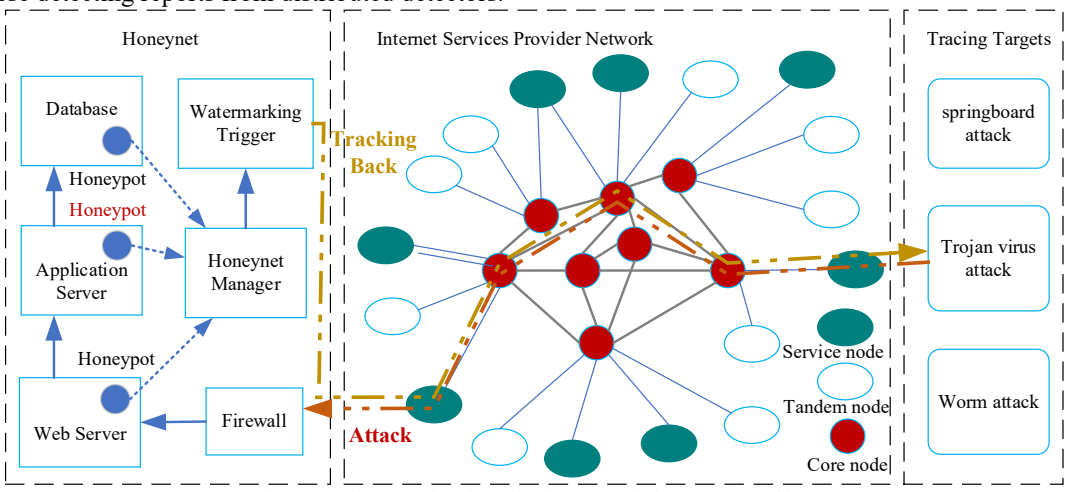


Figure 2. The scenario of attack with network flow watermarking

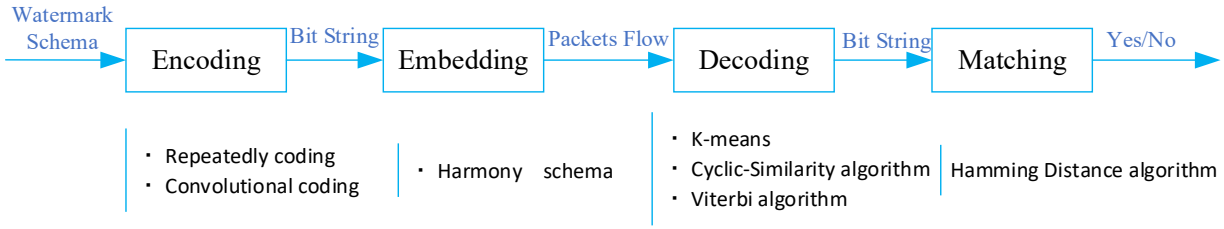


Figure 3. The main steps of RCIPD

The main steps of the proposed RCIPD are shown in Figure 3. Each step has key techniques to address difference issues, which are described as follows:

*Watermark Schema*

A special random algorithm is used to generate the original watermark. When the honeynet detects the attack, the watermarking trigger sends a tracking request to the control center of the tracking system. The control center receives the request and uses a random algorithm to generate a globally unique watermark bit sequence as an initial watermark schema. This generated watermark bit sequence is then transmitted along to the attacker, passing through all nodes on the path in the network.

*Encoding*

The input of this step is the random n-bit sequence from the control center. The original watermark is encoded as convolutional code, and then the watermark is encoded repeatedly according to the pre-determined redundancy value based on the RCIPD schema. In the system, the convolutional encoder is used to encode the sequence as shown in Figure 4.

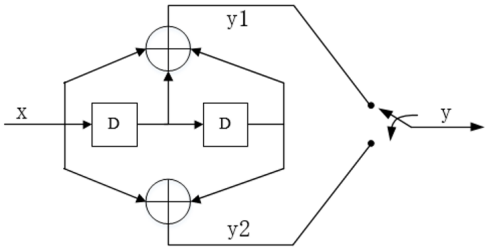


Figure 4. Convolutional encoder

Let x be an input bit sequence and $D$ be a shift register, where the left register is denoted as $D_1$ and the right as $D_2$. Let $y_1$ and $y_2$ be the output bits and Y be the combination of the two bits, which means that one bit input into convolutional encoder outputs two bits, and it is formulated as follows:

$$y_1 = x \oplus D_1 \oplus D_2$$
$$y_2 = x \oplus D_2$$

$D_1$ and $D_2$ move one bit to the right after each completion. Let m be the length of the convolutional encoded sequence. In this case, $m = 2n$. Then, the convolutional encoded sequence is repeated r times, where r is a parameter of RCIPD. The output of this step is a sequence with $r \times m$ bits.

*Embedding*

The input of this step is a sequence with $r \times m$ bits. RCIPD embeds the sequence into a packet flow based on a strategy, called the harmony schema.

As RCIPD only modulates the time interval between data packets in the data flow, to prevent the attacker from detecting the existence of the watermarking, a procedure needs to be performed on the packets of the data flow from the attacked host to the attacker to improve the system's invisibility.

- When generating a watermark schema, a special random algorithm is used to ensure that the number of bit 0s are equal

to that of bit 1s.

- All packets from the attacked host to the attacker face the same random delay.
- When the mathematical expectation of the delay between the packets containing the watermark is equal to that of the delay between the random packets, it will be difficult to find a statistical error between the two from the time statistical characteristics of the data flow. The output of this step is s packets flow to the attacker with the inter-packet delay modulated.

*Decoding*

The input of the decoding is s packets flow. The decoding can be divided into the following three steps:

(1) Extract from the packets flow a sequence with $r \times m$ bits by using K-means: In this step, the interval packet delay, *IPD*, is obtained by calculating the difference between the arrival times of two adjacent packets. Soon afterwards, *IPD* is converted to bit 0 or bit 1.

The traditional method uses a threshold value to decide, i.e., an IPD that is larger than the threshold value is converted to bit 0, and an IPD that is smaller than or equal to the threshold value is converted to bit 1. Due to the disturbance of the network environment, the time interval between all data packets has been extended. This results in a small IPD becoming larger than the threshold. This paper proposes to use the K-means algorithm to determine the partition criteria dynamically. By using the K-means algorithm, the values of IPD with similar values can be divided together. In other words, the threshold used to convert the IPD to bit 0 or bit 1 is dynamically changed according to the obtained distribution of the IPD.

(2) Delimit the sequence with $r \times m$ bits to a m-bits sub-sequence by using the cyclic-similarity algorithm (CSA): This paper proposes CSA to decode and restore a bit sequence containing multiple redundant watermarks and random errors to another bit sequence that is considered the possible convolutional coded watermark.

Assume that S is a bit sequence encoded by RCIPD, passing through a network that may incur packet splitting. S' is the bit sequence obtained by the decoder. CSA divides S into several subsequences with the same length, and the new sequence length is pre-defined in advance. There are two types of subsequences, subsequences without packet splitting and subsequences with packet splitting. The same or cyclic-similar subsequences are put into a same set. If sequence A' can be obtained by shifting around the first few bits of sequence A with the last of A, and sequence A' is equal to sequence B, then sequence A and sequence B are called cyclic-similar sequences. The targeted bit sequence can be obtained by only finding the set containing the most elements. Subsequences without packet splitting are in the same set, subsequences with packet splitting and the same packet splitting positions are in the same set, and subsequences with packet splitting but different packet splitting positions are in different sets. The pseudo code of CSA is shown in Algorithm (1).

```
Algorithm (1) CSA(S[], m)
Input: S[] is a decoded sequence, and m is the length of convolutional coded watermark
Output: The sequence delimited by CSA
1.  i←0
2. while i*pos < S.length do
3.      for j←0 to (m-1) do A[i][j]←S[i*pos + j]
4.      i←i+1
5. for k←0 to i do
6.      find all key of Map
7.      if the amount of key is zero
8.      then put (A[k], 1) into Map
9.      else if iscyclic-similar( A[k],key)=true
10.        then put (A[k],value+1) into Map
11.        else put (A[k], 1) into Map
12.  return key with maximum value in Map
```

```
Algorithm (2) iscyclic-similar (A[], B[])
Input: A[] is the sequence to identify, and B[]is the sequence in Map
Output: true or false, which stands for the correctness or incorrectness of the A[] is cyclic-similar sequence of B[]
1. length←A[].length
2. for i←0 to (length-1) do
3.       for j←0 to (length-1) do
4.           if B[j] != A[(i+j)/length] then break
```

5. else if j=(length-1) then return true
6. return false

(3) Convert the m-bits sequence to an n-bits sequence by using the Viterbi algorithm: The Viterbi decoding algorithm is applied to decode the bit sequence that has been encoded using a convolutional code. The Viterbi algorithm does the maximum likelihood decoding convolutional codes, with constraint lengths $k \leq 3$.

The output of this step is an n-bit sequence that is regarded as the detected watermark.

*Matching*

The input of this step is an n-bit decoded watermark. RCIPD uses the Hamming distance algorithm to match the decoded watermark with the original watermark to determine whether the node is in the attack path. Then, the matching result is reported to the control center of the whole system. The output of this step is the matching result.

## 4. Watermark Scheme Design and Analysis

*Watermarking Scheme Design*

The key to implement attack tracking based on the network flow watermark is to design a good watermark to generate, encode, and embed the watermark to the data flow and to extract the detection scheme corresponding to the detection node. This paper designs and implements a network flow watermark tracking model using the convolutional code, harmony algorithm, K-means algorithm, CSA, and Viterbi algorithm.

1. Network Flow Watermarking Model

The network flow watermark embedding component determines the data flow by extracting the quintuple from the whole attacker data flow, including the source IP, destination IP, source port, destination port, and protocol version number. The watermarks will be generated according to the watermarking schemas. The watermark embedding component embeds the watermark into the data flow by delaying the sending time to adjust the interval time between packets. The destination of the data flow will be the attacker.

Let *P* be the collection of data packets, $p_i$, (*i* =1, …, *n*), in an obtained attacker network data flow, where *n* is the number of data packets, denoted as follows:

$$\mathrm{P} = (p_1, p_2, \dots, p_n) \tag{1}$$

Let T be the time set of each data packet in the data flow P sending by the watermark embedding component.

$$\mathrm{T} = (t_1, t_2, \dots, t_n) \quad \{t_i < t_j | 1 \leq \mathrm{i} \leq \mathrm{j} \leq n\} \tag{2}$$

For any two consecutive data packets in the data flow P, let IPD be the set of the inter-packet delays between two consecutive data packets.

$$\mathrm{Ipd} = (ipd_1, \dots, ipd_i, \dots, ipd_n) \tag{3}$$

Where $ipd_i$ is the ith element of IPD, defined as follows:

$$ipd_i = t_{i+1} - t_i \qquad (1 \leq \mathrm{i} \leq \mathrm{n} - 1) \tag{4}$$

2. Watermark Generation

The watermark embedding component uses pseudo-random functions to generate globally unique bit sequences $w^o$.

$$w^o = \left(w_1, w_2, \dots, w_i, \dots, w_{L_{w^o}} | \, w_i = \{0, 1\}\right) \tag{5}$$

Where $L_{w^o}$ is the length of bits in the $w^o$ sequence. To enhance the anti-interference ability and robustness of watermark in a complex network environment, this paper uses convolutional code to encode $w^o$.

$$w^s = Encode(w^o) \tag{6}$$

Where $w^s$ is a 0-1 bit sequence after being encoded by a convolutional coder, and Encode() is an encoding function for convolutional code. Assume that the convolutional code rate is R, the length of the 0-1 bit sequence of $w^s$ after convolution encoding, and $L_{w^s}$, is computed as follows:

$$L_{w^s} = \left(\frac{1}{R}\right) * L_{w^o} \tag{7}$$

Let r be the repetition number of $w^s$ and $w^r$ be the 0-1 bit sequence after repeatedly coding, the final watermark, which is defined in Equation (8).

$$w^r = Repeat(w^s,\ r) \tag{8}$$

Where Repeat() is the repeated coding function. Let $L_{w^r}$ be the length of the 0-1 bit sequence of $w^r$, defined in Equation (9).

$$L_{w^r} = L_{w^s} * r \tag{9}$$

Where $w^r$ is the watermark bitflow used for embedding into the network flow.

## 3. Watermark Embedding

For a network flow with n number of packets, the length of its IPD is n-1, $ipd \in Ipd$. According to the captured n, the values of $L_{w^o}$, R and r in Equations (7) and (9) are adjusted, such that

$$L_{w^r} = n - 1 \tag{10}$$

For any watermark $w^r$ that is defined in Equation (11), $w_j^r \in w^r$ affects the IPD according to the following rules: let $\mu$ be the standard time of IPD and $\lambda$ be the multiplier of the IPD when the bit is 0. Let $\pi_j$ be the IPD between two adjacent packets. The rule for converting a bit sequence to IPD is shown in Equation (12).

$$w^r = \left(w_1^r,\ w_2^r,\ \dots,\ w_j^r, \dots,\ w_{L^{w^r}}^r\right) \tag{11}$$

$$\pi_j = \begin{cases} \mu & ,\ if\ w_j^r = 1 \\ \lambda \times \mu, & if\ w_j^r = 0 \end{cases} \tag{12}$$

That is, the inter-packet delay corresponding to any bit $w_j^s$ in watermark bit sequence $w^s$ to be transmitted is adjusted to $\pi_j$. Therefore, for packet $p_j \in P$, let $t_{p_j}$ be the time to transmit the jth packet, calculated as shown in Equation (13).

$$t_{p_j} = t_j + \sum_{i=1}^{j} \pi_i \tag{13}$$

The purpose of embedding the watermark into the data flow is achieved by modulating the interval between the data packets in the data flow, and after the watermark is embedded, the data flow is sent to the attacker.

To improve the invisibility, this paper proposes a harmony schema to prevent the attacker from detecting the existence of the watermark. The following two steps are performed:

- The watermark generated above is generated by a special random algorithm, which makes the number of bit 1s the same as that of bit 0s;
- The other packets without watermarks embedded are also delayed for a period of time, the same as the time of bit 1 or the time of bit 0 randomly.

The above two steps make the mathematical expectation of the delay time of the packets with watermarks embedded and the packets without watermarks the same. In this way, it will be difficult to find a statistical error between the two from the time statistical characteristics of the data flow. Thus, the system could be regarded as statistically invisible [4,10].

## 4. Watermark Extraction

The watermark extraction and detection component extracts the watermark by identifying the destination network flow, calculating the IPD of the network flow, recovering a bit sequence of length $L_{w^s}$, and decoding with the Viterbi algorithm.

The watermark extraction and detection component identifies destination network flow by verifying the quintuple. In consideration of the network disturbances, such as small-sized or medium-sized congestion, low communication delay, and other factors, some packets will arrive late and the received IPD will be bigger than the sent IPD. Therefore, the K-means algorithm is adopted in this scheme, which can better restore the IPD to the original bit sequence. The strategy of recovering the original bit sequence that uses the K-means algorithm is more accurate than the conventional schemas. By using the K-means algorithm, all IPDs will be divided into two groups. All IPDs in one group are smaller than those in another group. The group containing the small IPDs is recorded as $G_1$, and the group containing the large IPDs is recorded as $G_2$. For each bit of

bit sequence $w^{r'}$, this strategy can be described as Equation (14).

$$w_j^{r'} = \begin{cases} 1, ipd_j' \in G_1 \\ 0, ipd_j' \in G_2 \end{cases} \tag{14}$$

Where $ipd_j'$ is the difference in the arrival time between the (j+1)th packet and the jth packet of the received network flow.

Suppose $w^{r'}$ is the extracted watermark bit sequence. The bit sequence $w^{s'}$ is recovered from $w^{r'}$ by using CSA. The main steps of CSA include dividing $w^{r'}$ into multiple subsequences according to the $L_{w^s}$ defined in Equation (7), finding the most frequent cyclic-similar sequence, and recording it as $w^{s'}$.

For example, the sequence $w_1, w_2, \dots, w_i, \dots, w_n$ and the sequence $w_i, w_{i+1}, \dots, w_n, w_1, w_2, \dots, w_{i-1}$ are cyclic-similar sequences.

The bit sequence $w^{s'}$ can be considered as the most likely sequence after the convolutional coding of sequence $w^{o'}$. The watermark extraction and detection component can obtain $w^{o'}$ by decoding $w^{s'}$ with the Viterbi algorithm. In this process, the Viterbi algorithm failovers some errors caused by network transmission.

$$w^{o'} = decode(w^{s'}) \tag{15}$$

After a series of processing, the $w^{o'}$ obtained is the final restored watermark.

## 5. Watermark Correlation Detection

The correlation detection method adopted in the proposed schema is to calculate the Hamming distance to the watermark sequence $w^o$ before the convolutional coding and the watermark sequence $w^{o'}$ obtained by detecting.

$$\text{Distance} = \sum_{i=1}^{n} XOR(w_i^o * w_i^{o'}) \tag{16}$$

Where XOR() represents an exclusive OR operation. Let α be the threshold of correlation detection, and when the distance is less than α, it can be considered that the watermark bit sequence is the same watermark.

*Interference Analysis*

Packet-based network flow watermarking requires high stability of networks; however, packet merging or packet splitting may occur during the transmission of the network flow, which may reduce or increase the number of bits of the watermark bit flow. In addition to passive packet merging or packet splitting, there may be an attacker manipulating the controlled key route to perform packet merging or packet splitting on the network flow that may contain a watermark, thereby destroying the original watermark. This is called active packet merging or packet splitting. Regardless of whether it is active or passive packet merging or packet splitting, it will cause greater interference to watermark detection. This paper proposes a solution to packet merging and packet splitting that may occur during the transmission of network data flow.

### 1. Packets Merging

Passive packet merging occurs during the process of network flowing. The forwarding router may combine two smaller data packets into one data packet and then forward it, which causes the watermark detection extractor to miss several bits when detecting the watermark. This has a large impact on watermark correlation detection. Active packet merging refers to the phenomenon in which an attacker merges adjacent data packets in a network flow after discovering a network flow that may contain a watermark at a controlled critical route.

Due to the impact of packet merging, the total number of packets in the network flow may be reduced, and the number of IPDs will also be reduced. For example, if the jth packet and the (j+1)th packet are merged into one packet, the jth IPD will disappear. Therefore, the sequence obtained by the K-means algorithm will change from $w^r$ to $w^{r1'}$. $w^r$ is given by Equation (11), and $w^{r1'}$ is shown in Equation (17).

$$w^{r1'} = \left(w_1^{r1'}, \dots, w_{j-1}^{r1'}, w_{j+1}^{r1'}, \dots, w_{Lw^r}^{r1'} | \, w_i = \{0, 1\}\right) \tag{17}$$

A subsequence delimitated from $w^{r1'}$ is obtained using the CSA algorithm, the subsequence and its cyclic-similar sequence are decoded by the Viterbi algorithm, and the watermark detection and extraction component decides whether the matching is successful using the Hamming distance. Whether the watermark detector is on the path of the network flow transmission can be concluded based on the association relationship of the detection result and the watermark of the sender.

### 2. Packets Splitting

Passive packet splitting occurs during the process of network flowing. The forwarding router may split a large packet into several smaller packets and forward it, leading to an increase in the number of packet obtained by the watermark detection and extraction component. Active packet merging refers to the attacker splitting certain data packets in the network flow after detecting the network flow that may contain the watermark at the controlled key router, and dividing the original data packet into multiple adjacent packets. It also leads to an increase in the number of packets obtained by the watermark detection and extraction component.

Suppose that if the packet splitting occurs in the jth data packet, there will be a very small IPD added. The sequence obtained by the K-means algorithm will change from $w^r$ to $w^{r2'}$. The $w_k^{r2'}$ in this sequence is the embodiment of packet splitting. Because the IPD generated by packet splitting is very small, generally less than the threshold value, the IPD will be converted to bit 1. The value of $w_k^{r2'}$ depends on the size of the added IPD. $w^{r2'}$ is shown in Equation (18).

$$w^{r2'} = (w_1^{r2'}, \ \ldots, w_j^{r2'}, w_k^{r2'}, w_{j+1}^{r2'}, \ldots, w_{Lw^r}^{r2'}) \tag{18}$$

A subsequence delimitated from $w^{r2'}$ is obtained by using CSA, the subsequence and its cyclic-similar sequence are decoded by the Viterbi algorithm, and the watermark detection and extraction component determines whether the matching is successful using the Hamming distance. The detection result and the watermark of the sender are used to determine if there is an association relationship and conclude whether the watermark detector is on the path of the network flow transmission.

## 3. Correctness of CSA

In this section, the correctness of the cyclic-similarity algorithm will be proven. The original sequence can be recovered correctly when some errors occur in the repeated sequence generated by the original sequence.

CSA generates a set of sets containing cyclic-similar sequences, and the elements in different sets are not cyclic-similar sequences. Therefore, if it can be proven that the number of elements in the set containing the cyclic-similar sequences of original sequences is greater than the number of elements in the set containing error sequences, then the correctness of the algorithm will be proven.

This paper sets the IPD with a duration of $\lambda \times \mu$ to bit 0 and the IPD with a duration of $\mu$ to bit 1. When packet splitting occurs, the IPD between the split packet and the previous packet is very small. Due to the property of the K-means algorithm, this IPD is converted to bit 1. Due to the nature of CSA, the number of subsequences without packet splitting is greater than that of any kind of subsequences with packet splitting. In other words, the value of 0 or 1 for the added bit does not affect the final result.

Therefore, when the packet in the network flow splits, the number of packets in network flow will increase. In other words, if the network flow watermarking is converted to 0-1 bit sequences, the number of bits will increase, an extra bit 0 or bit 1 will be added to the sequence, and the bit sequence will increase by 1 bit, i.e., a bit sequence has been polluted.

Suppose the length of the bit sequence that is extended by the Viterbi algorithm is m bits and the bit sequence was repeated r times. According to the decoding schema of RCIPD, as long as the number of sequences that are not polluted is larger than the mathematical expectation of the number of one kind of the polluted sequences, the original watermark can be recovered through the bit sequences that are not polluted. Assume that the number of packet splitting is set to k. Through the relationship between k and r, the mathematical expectation of the number of unpolluted sequences can be obtained.

The random variable $X_i$ is defined in Equation (19).

$$X_i = \begin{cases} 1, & \text{when the } i\text{th sequence is polluted} \\ 0, & \text{when the } i\text{th sequence is not polluted} \end{cases} (\mathrm{i} = 1,2,3,\ldots, \mathrm{r}) \tag{19}$$

Let $Y = X_1, X_2, \ldots\ldots, X_{\mathrm{r}}$ be the number of polluted sequences. For the ith sequence, when the packet is split, the probability of occurrence in the sequence is $\frac{1}{r}$, and the probability of un-occurrence in the sequence is $(1 - \frac{1}{r})$. The events that each packet is split in each sequence are independent of each other, so the probability that k packets are not split in the sequence is $\left(1 - \frac{1}{r}\right)^k$. The probability of at least one packet splitting in the sequence can be expressed as $1 - \left(1 - \frac{1}{r}\right)^k$.

The probability of $X_i$ can be represented as Equation (20).

$$P\{X_i = 0\} = \left(1 - \frac{1}{r}\right)^k$$
$$P\{X_i = 1\} = 1 - \left(1 - \frac{1}{r}\right)^k \tag{20}$$

The mathematical expectation of variables $X_i$ and $Y$ can be represented as Equation (21).

$$E(X_i) = 0 \times \left(1 - \frac{1}{r}\right)^k + 1 \times \left(1 - \left(1 - \frac{1}{r}\right)^k\right) = 1 - \left(1 - \frac{1}{r}\right)^k$$

$$E(Y) = E(X_1) + E(X_2) + E(X_3) + \cdots + E(X_r) = r \times \left(1 - \left(1 - \frac{1}{r}\right)^k\right) \tag{21}$$

$E(Y)$ is the mathematical expectation of the number of sequences with packet splitting. Therefore, the number of sequences with packet splitting can be obtained, represented as Equation (22).

$$r - E(Y) = r \times \left(1 - \frac{1}{r}\right)^k \tag{22}$$

Let w be the mathematical expectation of the number of sequences with packet splitting, expressed as Equation (23).

$$w = r - r \times \left(1 - \frac{1}{r}\right)^k \tag{23}$$

For packet splitting that has occurred k times, it is assumed that the number of packet splitting on each sequence is the same. Consequently, the mathematical expectation of the number of packets splitting on one sequence can be calculated as $\frac{k}{w}$. Since each segment contains m bits, the number of all cases where packet splitting can occur is $m^{\frac{k}{w}}$. The mathematical expectation of one type of packet splitting can be calculated as $\frac{w}{m^{\frac{k}{w}}}$. The correctness of the algorithm can be proven as long as Inequality (24) holds.

$$\frac{w}{m^{\frac{k}{w}}} \leq r - w \tag{24}$$

Taking Equation (23) into Inequality (24) leads to Inequality (25).

$$\frac{r - r \times \left(1 - \frac{1}{r}\right)^k}{m^{\frac{k}{r - r \times \left(1 - \frac{1}{r}\right)^k}}} \leq r \times \left(1 - \frac{1}{r}\right)^k \tag{25}$$

After simplification and inequality zoom (see Appendix A for details), this conclusion can be drawn. When the variables m and $r$ are fixed, Inequality (25) holds for any nonnegative variable k.

The algorithm is correct under the condition above. According to the algorithm proposed in this paper, the original watermark can be recovered completely and correctly so long as only two pieces of sequence without packet splitting can be obtained. Calculation shows that if the mathematical expectation of the total number of sequences without packet splitting is made greater than 1, $r$ and $\frac{r}{k}$ are positively correlated (i.e., the growth of the variable $\frac{r}{k}$ increases with the growth of the variable $r$). Therefore, when the probability of packet splitting is invariant, the number of sequences can be obtained without packet splitting, and the variable $r$ is positively correlated. Therefore, the conclusion can be drawn that the higher the redundancy, the higher the possibility of the algorithm.

## 5. Experiment Results & Evaluation

*Experiment Design*

The experiment designs and implements the watermark sender and the watermark detector to generate, transmit, receive, and recover the network flow watermark.

The watermark sender implements the random generation of the original watermark, expands it by the Viterbi algorithm, repeats it according to the pre-defined redundancy number, and sends the data packet by controlling the packet interval time according to the pre-defined rules. The number of packet splitting is randomly added to the corresponding number of data

packets, thereby implementing an operation in which the attacker randomly performs a packet split operation on the controlled router to destroy the watermark. The key router with the detection component is simulated as the watermark detector. The watermark detector obtains the packet attributes by recording the arrival time of the packets of different source IPs and destination IPs and obtaining the IPD.

The K-means algorithm is applied to the IPD. The watermark is obtained, and the original watermark is restored by the algorithm described in the paper and compared with the watermark database to determine whether the watermark is correctly restored.

In the experiment, the original watermark bit, redundancy, number of packet splitting, and other parameters can be controlled to conduct multiple experiments, obtaining the influence of specific variables on the RCIPD schema through the control variable method.

In addition, three classical network flow watermarking schemas, IPD, IBW (interval-based watermark), and ICBW (interval centroid-based watermark), are implemented to conduct experimental comparison to observe the differences between classical network flow watermarking schemas and the proposed RCIPD schema in the case of packet splitting.

*Experiment Results*

This paper proposes a packet-based network flow watermarking schema to achieve attack source tracing. Initially, the network situation of packet loss is simulated. By setting the number of packet loss per 100 packets and the redundancy of RCIPD, the percentage of watermarks that can be recovered correctly can be measured. The results are used to evaluate the robustness of the RCIPD schema.

In the experiment of simulated packet loss, the original watermark of 10 bits is used for the RCIPD schema, which is expanded to 20 bits by the Viterbi algorithm, adopting the redundancy of 100. The numbers of packet loss are set to 0, 100, 200, 300, 400, and 500. The redundancies are set to 10, 30, 50, and 100, respectively. There are 1,000 times of experiments carried out in each redundancy, obtaining the correct rate of watermark recovery. Figure 5 shows the correlation between the number of packet loss and correctly recovering RCIDP in different repetition times.

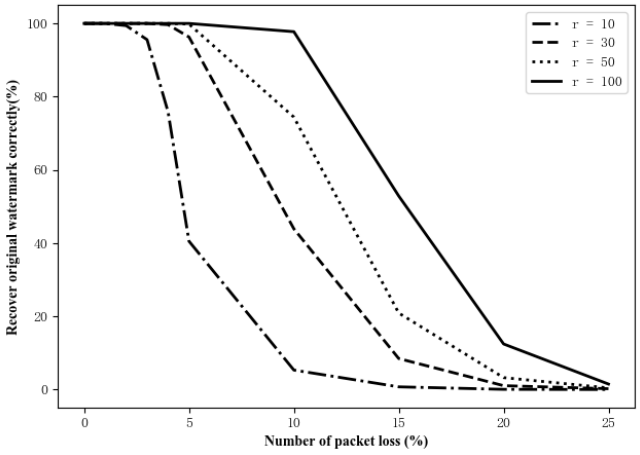


Figure 5. Capability of correctly recovering watermarking with packet loss

It is observed that when the redundancy is higher, the robustness of RCIPD schema is higher. When less than three packets in every 100 packets are lost, the RCIPD schema can recover the original watermark with 100%.

Afterwards, the experiment of restoring the original watermark's accuracy by using different redundancies and the number of packets splitting per 100 packets is carried out. As the number of packet splitting is increased gradually, multiple experiments are performed under different packet splitting times, recording the number of times the original watermark is correctly restored in the current situation.

Different redundancies of the RCIPD schema are tested to explore the relationship between different redundancy and the rate of successful restoration of the original watermark. In the experiments, 10 bits original watermark is used, and through the

Viterbi algorithm it becomes the length of 20 bits. An experiment that uses the redundancies of 10, 30, 50, and 100 is carried out. The accounts of packet number for the proportion of the total number of packets is the independent variable. The independent variable values are set to 0%, 1%, 2%, 3%, 4%, 5%, 9.5%, 10%, 12.5%, 15%, 17.5%, 20%, 22.5%, and 25%. There are 1,000 times of experiments for each independent variable. The number of successfully restored original watermarks is converted to accuracy and plotted in Figure 6.

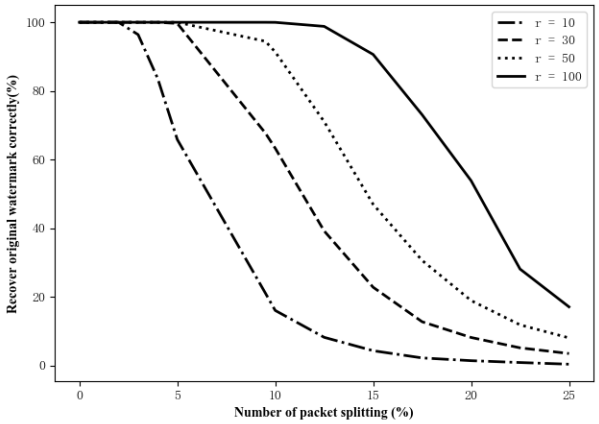


Figure 6. Comparison of different watermarking with packets splitting

The broken line graph shows that with the increase in the number of packet splitting, the probability that the RCIPD schema can recover the original watermark correctly decreases. When the number of packet splitting is fixed, the higher the number of repeated encoding, the higher the probability of recovering the original watermark correctly. The higher the number of repeated encoding, the higher the ratio of packet splitting to the total number of packets, and the better the RCIPD schema can recover the original watermark with a probability of 100%.

It is observed that the higher the redundancy, the more robust the RCIPD schema is, and the stronger the ability of the RCIPD schema to resist packet splitting. In addition, a comparative experiment is also carried out. The performance of the RCIPD schema is verified by comparing the RCIPD schema with the IPD schema, IWB schema, and ICBW schema to recover the original watermark correctly in the case of simulated packet loss and packet splitting.

In the experiment of simulated packet loss, the original 10 bits watermark is used for the RCIPD schema that expands it to 20 bits by the Viterbi algorithm, adopting a redundancy of 100. The number of packet loss is 0, 100, 200, 300, 400, and 500. A redundancy of 100 is used in RCIPD. The comparison of these network flows in packet splitting is shown in Figure 7.

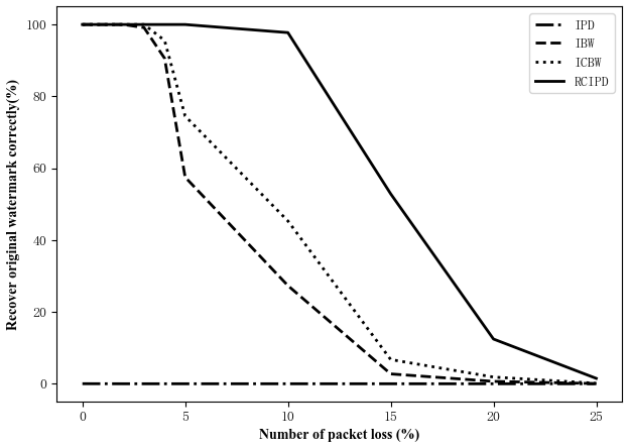


Figure 7. Comparison of RCIPD in different repetition times with packets loss

The broken line graph shows that the ability of the RCIPD schema with a redundancy of 100 to recover the watermark correctly is much higher than other schemas in the case of packet loss. By comparing with the previous figure, it is observed that the accuracy of the RCIPD scheme with a redundancy of 30 is similar to that of the IWB schema and ICWB schema.

In the experiment of simulated packet splitting, the original watermark of 10 bits is used for the RCIPD schema, expanding it to 20 bits by the Viterbi algorithm and adopting a redundancy of 100. The number of packet splits is set to 0, 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500. Each group of experiments is performed 1,000 times, and the number of times the watermark is correctly restored in each group of experiments is recorded. At the same time, the proportion of the number of packet splits in the packet flow to the total number of packets including the watermark and the probability of correctly recovering the watermark are recorded. The relevant situation is recorded for comparison.

The total number of IBW schemas and ICBW schemas are set to 2,000, so that they have the same total number as in the RCIPD schema experiment. After that, the random packet splitting simulation is carried out. The number of packet splitting is set to 0, 50, 100, 150, 200, 250, 300, 350, 400, and 450. The number of correct recoveries of watermark in each group of experiments is recorded, and the corresponding accuracy is observed. The comparison of these network flows in packet splitting is shown in Figure 8.

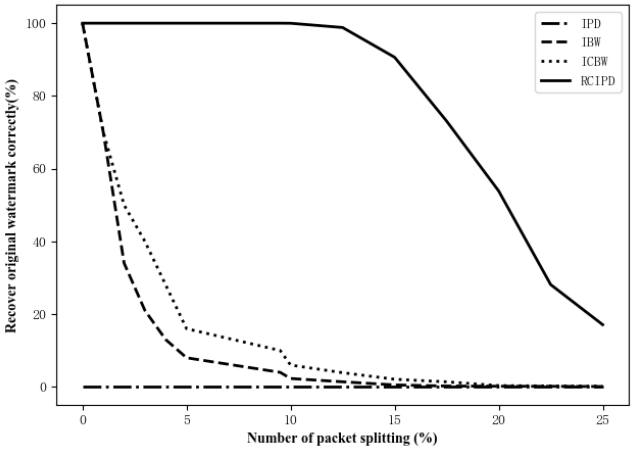


Figure 8. Comparison of different watermarking with packets splitting

The experimental data shows that when the number of packet splits in the packet flow accounts for less than 10% of the total number of packets containing the watermark, the watermark schema of this paper can recover the watermark by almost 100%. When the number of packet splits accounts for 9.5% of the total number of packets including the watermark, 10,000 experiments are performed, and only the case when the original watermark could not be correctly restored occurs twice. In the actual situation, the attacker does not perform excessive packet splitting operations on the network flow, so the superiority of the network flow watermarking schema can be observed.

By comparing the results of many experiments, the performance of the RCIPD schema is verified, especially the performance of the RCIPD schema in the aspect of resisting packet splitting compared with other classical schemes. The experiment compares the performance of different network flow watermarking in resisting packet splitting by completely and correctly recovering the accuracy of the original watermark. The experiment shows that in the case of packet splitting, the IPD schema is devastated, and the receiver of the watermark will not be able to recover the original watermark correctly, while other schemas will resist different degrees of packet splitting according to their characteristics. With the increase in packet splitting times, the IBW schema first fails to recover the original watermark correctly, and the ICBW schema will not recover the original water correctly. However, the RCIPD schema performs well. Only when the number of packet splitting in the data flow accounts for about 10% of the total packets containing the watermark can the original watermark not be recovered completely.

Robustness refers to the property of watermark resistance to active noise added by an attacker to alter the watermark carrier. It is observed from the above results that RCIPD affected by packet splitting or merging is low; therefore, RCIPD is still efficient in complex network environments, indicating the robustness is high.

- RCIPD is efficient when packet loss occurs. The result can be observed in Figure 5. \When $r = 100$, RCIPD is efficient with extremely high correctness even when the percentage of packet loss is 10%.
- RCIPD is efficient when packet splitting occurs. The result can be observed in Figure 6. When $r = 100$, the correctness is still high even when the percentage of packet splitting reaches 15%.
- In Figures 7 and 8, it is observed that the RCIPD performance is better than others when packet loss and packets splitting occur.

Therefore, the robustness of the RCIPD is higher than the robustness of other methods.

In the experiment of the invisibility of RCIPD, the original 10 bits watermark is used and expanded to 20 bits by the Viterbi algorithm, adopting a redundancy of 100. Because of the harmony algorithm, the closest mathematical expectation of the random sequence and mathematical expectation of the detected watermarking sequence can make the watermark have the highest invisibility.

100 RCIPD experiments are carried out, and the average of the mathematical expectation of the detected watermarking sequence is obtained. Let $E(Y')$ be the average of the mathematical expectation of the detected watermarking sequence. Then, let $E(X)$ be the mathematical expectation of the random sequence. Different values of $E(X)$ are used in the watermark embedding party, that is, 100 experiments are carried out on the random sequence with the mathematical expectation of 160, 180, 200, 220, 240, 260, 280, and 300, and the average of the mathematical expectations of those sequences in the watermark extraction and detection party is recorded and represented by $E(X')$. The relationship between $E(X)$ and $\|E(Y') - E(X')\|$ is shown in Figure 9.

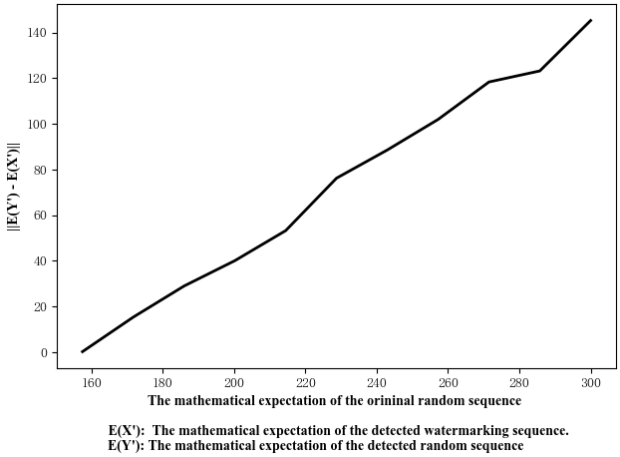


Figure 9. The influence of the mathematical expectation of the random sequence

$E(X)$ is the mathematical expectation of the delaying time of the original watermark, and $E(\mathrm{Y})$ is for the original random sequences. $E(X')$ is the mathematical expectation of the detected random sequence, and $E(Y')$ is for the detected random sequences. When the difference of the mathematical expectation between the original bit sequence embedded watermark $E(\mathrm{X})$ and the original random bit sequence $E(\mathrm{Y})$ is greater, the difference of $E(X')$ and $E(Y')$ is also becoming greater. When the difference of the mathematical expectation between E(X) and E(Y) is ignorable, the difference between $E(X')$ and $E(Y')$ is also ignorable. In particular, when $E(X)$ is equal to 160, $\|E(Y') - E(X')\|$ is approximately equal to 0. Thus, while the mathematical expectation of the random sequence and the mathematical expectation of the detected watermarking sequence are close to each other, RCIPD can be regarded as a watermarking scheme with excellence invisibility.

At the same time, the RCIPD schema is qualitatively compared with other three schemas, i.e., IPD, IBW, and ICBW, based on the metrics for network flow watermarking tracking schemas, including robustness, invisibility, watermark capacity, embedding overhead, extraction overhead, number of packets, and resisting of packet splitting, as shown in Table 1.

Table 1. Comparison of different watermarking schemas

| | RCIPD | IPD | IBW | ICBW |
|---|---|---|---|---|
| Robustness | High | Low | Low | Low |
| Invisibility | Mid | Low | Low | Low |
| Watermark capacity | Low | Mid | Low | Low |
| Embedding overhead | Mid | Low | Mid | Mid |
| Extraction overhead | Mid | Low | Mid | Mid |
| Number of packages | High | Low | Mid | Mid |
| Resisting PS | High | Low | Mid | Mid |

## 6. Discussions

This paper proposes a time-based network flow watermarking scheme. The experimental results show that the detection accuracy of RCIPD scheme proposed is high. The original watermark can be recovered very effectively when the packet loss is small. RCIPD also exhibits high detection accuracy even when faced with packet merging and packet splitting, which impact

network flow watermarking. In the analysis of the packet splitting, it is observed that the higher the number r of repeated watermarks, the higher the ratio of recovering the original watermark, and the higher the detection accuracy when the number of packet splitting is lower. In the analysis of the packet merging, it is observed that compared with IPD, IBW, and ICBW, the accuracy of RCIPD to restore the original watermark is obviously improved.

During the process of generating and encoding watermark, the convolutional encoding algorithm is used to improve the accuracy of transmission, which alleviates the influence of network disturbances, but it will also increase the number of bits needed to be forwarded.

When the watermark bits are embedded to the data packets and sent to the attacker, the harmony schema is used to make the mathematical expectation of the flows containing watermarks and no-watermarks the same. This approach makes it difficult for the attacker or third party to realize the existence of the watermark, and the invisibility is improved. However, the side effect is that it may cost more time to send the packets.

The K-Means algorithm is used to determine the new criterion to differentiate 0 and 1, which may improve the accuracy because of the delay of the data packets. The cyclic-similarity algorithm is used to obtain the convolutional code of the original watermark with maximum likelihood, and the cost is that it will token more bits.

The Viterbi decoding algorithm is used in the detecting node to obtain a bit sequence that may be the original watermark. This method decreases the error rate during transmission, but it cost more than twice the number of bits to achieve the performance.

From the discussion above, one can see that the high invisibility, accuracy, and robustness are at the expense of a relatively high cost of time. Hence, RCIPD is slower than other classical watermarking schemes. Therefore, in practical applications, it needs to comprehensively consider various factors to decide which one to adopt.

## 7. Conclusions

This paper proposes an improved time-based network flow watermarking scheme. The scheme will transmit the watermark information that needs to be embedded in the network flow for n times in the embedded network flow. In this way, when the detection node detects, the redundant watermark information can be used to recover the original watermark information. At the same time, the problems of packet splitting and packet merging can be solved, and the detection accuracy and the system robustness are effectively improved. The problem that the scheme is time-consuming needs further improvement.

This paper proposes an improved time-based network flow watermarking scheme. Several techniques are proposed at each stage of the watermark scheme. In the coding phase, convolutional code and repeatedly coding are used. In the embedding phase, the harmony algorithm is proposed. In the decoding phase, the K-Means algorithm and Viterbi algorithm are applied, and CSA is proposed. In the matching phase, the Hamming distance algorithm is introduced. The K-means algorithm, CSA, and Viterbi algorithm improve the robustness of the algorithm, and the harmony schema improves the invisibility of the algorithm.

Five experiments are designed and implemented. The accuracy of RCIPD under different redundancies is compared. In addition, RCIPD is compared with IPD, IBW, and ICBW.

To verify the robustness of RCIPD, the experiments with different redundancies are implemented in the case of packet loss. To verify the ability of RCIPD to resist packet splitting, experiments with different redundancies are implemented in the case of packet loss. To verify the performance difference between RCIPD and classical watermarking schemes, the comparative experiments of RCIPD, IPD, IBW, and ICBW are implemented under the condition of packet loss and packet splitting. To verify the influence of the mathematical expectation of the random sequence in the harmony algorithm, experiments with different mathematical expectations of the detected watermarking sequence are implemented. Based on the experimental results, the robustness and invisibility of RCIPD are quantitatively analyzed. In addition, such indicators as the watermark capacity and extraction difficulty of RCIPD are qualitatively analyzed.

RCIPD takes a relatively longer time to fulfill its task and needs further improvement in future work.

## Acknowledgements

This work is supported by the Ministry of Education-China Mobile (No. MCM20170406) and the National Natural Science

Foundation of China (No. 61872011). The authors would also like to thank the anonymous reviewers for their invaluable comments.

## References


1. G. Forney, "Convolutional codes I: Algebraic structure," Journal of IEEE Transactions on Information Theory, Vol. 16, No. 6, pp. 720- 738, December 1970
2. J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," Journal of Computer Networks, Vol. 28, No. 1, pp. 100- 108, April 2013
3. A. J. Viterbi, "Viterbi Algorithm," 1st Edition, Encyclopedia of Telecommunications, 2003
4. R. W. Hamming, "Error Detecting and Error Correcting Codes," Journal of Bell System Technical Journal, Vol. 29, No. 2, pp. 147- 160, April 1950
5. L. Zhang, Y. Kong, Y. Guo, J. Yan and Z. Wang, "Survey on network flow watermarking: model, interferences, applications, technologies and security," Journal of IET Communications, Vol. 12, No. 14, pp. 1639- 1648, May 2018
6. Z. Yi, L. Pan and X. Wang, "IP Traceback Using Digital Watermark and Honeypot," in Proceedings of the International Conference on Ubiquitous Intelligence and Computing, pp.426-438, Berlin, Heidelberg, 2008
7. X. Wang, J. Jiang and L. Bai, "SWATS: A Lightweight VANET Anonymous Traceback System Based on Random Superposition Watermarking," in Proceedings of the 2018 IEEE International Conference on SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, pp.748-755, Guangzhou, China, October 2018
8. Y. Jing, P. Tu, X. Wang and G. Zhang, "Distributed-log-based scheme for IP traceback," in Proceedings of the 5th ACM International Conference on Computer and Information Technology, pp.711-715, Shanghai, China, November 2005
9. E. Hilgenstieler, E. P. Duarte, G. Mansfield-Keeni and N. Shiratori, "Extensions to the source path isolation engine for precise and efficient log-based IP traceback," Journal of Computers & Security, Vol. 29, No. 4, pp. 383- 392, June 2010
10. J. Shi, L. Zhang, S. Yin, W. Liu, J. Zhai, G. Liu and Y. Dai, "A Comprehensive Analysis of Interval Based Network Flow Watermarking," in Proceedings of the 4th International Conference on Cloud Computing and Security, pp.72-84, Haikou, China, June 2018
11. A. Iacovazzi, S. Sarda and D. Frassinelli, "DropWat: An Invisible Network Flow Watermark for Data Exfiltration Traceback," Journal of IEEE Transactions on Information Forensics and Security, Vol. 13, No. 5, pp. 1139-1154, May 2017
12. R. Stone, "CenterTrack: An IP overlay network for tracking DoS floods," in Proceedings of the 9th Conference on USENIX Security Symposium, pp.114, Colorado, USA, August 2000
13. T. Baba and S. Matsuda, "Tracing network attacks to their sources," Journal of IEEE Internet Computing, Vol. 6, No. 2, pp. 0-26, April 2002
14. D. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in Proceedings of the 12th IEEE International Conference on Computer and Communications Society, pp.878-886, Alaska, USA, Februray 2001
15. A. Belenky, "IP traceback with deterministic packet marking," Journal of IEEE Comm Letters, Vol. 7, No. 4, pp. 162- 164, May 2003
16. M. T. Goodrich, "Efficient packet marking for large-scale IP traceback," in Proceedings of the 9th ACM International Conference on Computer and Communications Security, pp.18-22, Washington, USA, November 2002
17. A. Lazarevic, L. Ertoz and V. Kumar, "A comparative study of anomaly detection schemes in network intrusion detection," in Proceedings of the 2003 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, pp.25-36, San Francisco, USA, May 2003
18. Lopez, R. Diego, John, Wolfgang, Olovsson and Tomas, "Detection of malicious traffic on back‐bone links via packet header analysis," Journal of Campus-Wide Information Systems, Vol. 25, No. 5, pp. 342- 384, January November 2008
19. N. Venkatesu, V. D. Chakravarthy and D. Sathya, "An Effective Defense Against Distributed Denial of Service in GRID," in Proceedings of the 1st IEEE International Conference on Emerging Trends in Engineering and Technology, pp.373-378, Maharashtra, India, July 2008
20. N. Wu, Z. Mu, and L. Zhang, "Distributed Denial of Service Covert Flow Detection Based on Data Stream Potential Energy Feature," Journal of Computer Engineering, Vol. 41, No. 3, pp. 142- 146, March 2015
21. W. Cheng, J. Zhao and P. Wu, "Network traffic detection and analysis based on big data flow," Journal of Journal of Nanjing University of Science & Technology, Vol. 41, No. 3, pp. 294- 300, June 2017
22. Y. Zhang and Q. Shen, "Data Flow Control Algorithm based on Feature Matching in Mobile P2P Network," Journal of Journal of Networks, Vol. 8, No. 5, pp. 1146, May 2013
23. B. Jonsson, "A fully abstract trace model for dataflow and asynchronous networks," Journal of Distributed Computing, Vol. 7, No. 4, pp. 197-212, May 1994
24. A. Iacovazzi and Y. Elovici, "Network Flow Watermarking: A Survey," Journal of IEEE Communications Surveys & Tutorials, Vol. 19, No. 1, pp. 512- 530, September 2016
25. T. Lu, R. Guo, L. Zhao and Y. Li, "A Systematic Review of Network Flow Watermarking In Anonymity Systems," Journal of International Journal of Security and Its Applications, Vol. 10, No. 3, pp. 129- 138, October 2016
26. X. Wang, D. S. Reeves and S. Wu, "Sleepy watermark tracing: An active network-based intrusion response framework," in Proceedings of the 22nd International Conference on International Information Security Conference, pp.369-348, Boston, USA, June 2001
27. S. Kent and R. Atkinson, "IP encapsulating security payload (ESP)," 1st Edition, RFC Editor, 1995
28. A. Iacovazzi and A. Baiocchi, "Optimum packet length masking," in Proceedings of the 22nd International Conference on Teletraffic Congress, pp.1-8, Amsterdam, The Netherlands, October 2010
29. L. Zhang, Z. Wang, Q. Wang, and F. Miao, "MSAC and multiflow attacks resistant spread spectrum watermarks for network flows," in Proceedings of the 2nd IEEE International Conference on Information & Financial Engineering, pp.438-441, Chongqing, China, September 2010
30. Z. Liu, J. Jing and P. Liu, "Rate-based watermark traceback: A new approach," in Proceedings of the 6th International Conference on Information Security Practice & Experience, pp.172-186, Seoul, South Korea, May 2010

31. 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 International Conference on Computer and Communications Security, pp.20-29, Washington, DC, USA, October 2003
32. Y. H. Park and D. S. Reeves, "Adaptive timing-based active watermarking for attack attribution through stepping stones," in Proceedings of the 2nd International Conference on Security in Distributed Computing Systems, pp.107-113, Washington, USA, June 2005
33. Z. Pan, H. Peng and X. Long, "A watermarking-based host correlation detection scheme," in Proceedings of the 2009 International Conference on Management of e-Commerce and e- Government, pp.393-497, Nanchang, China, September 2009
34. X. Wang, S. Chen and S. Jajodia, "Tracking anonymous peer-to-peer voip calls on the internet," in Proceedings of the 12th ACM International Conference on Computer and Communications Security, pp.81-94, Alexandra, USA, November 2005
35. A. Houmansadr, N. Kiyavash and N. Borisov, "Non-blind watermarking of network flows," Journal of IEEE/ACM Transactions on Networking, Vol. 22, No. 4, pp. 1232- 1244, August 2014
36. A. Houmansadr and N. Borisov, "Towards improving network flow watermarking using the repeat-accumulate codes," in Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1852-1855, Prague, Czech Republic, May 2011
37. Y. J. Pyun, Y. H. Park, X. Wang, D. S. Reeves and P. Ning, "Tracing traffic through intermediate hosts that repacketize flows," in Proceedings of the 26th IEEE International Conference on Computer Communications, pp.634-642, Anchorage, USA, May 2007
38. Y. J. Pyun, Y. H. Park, D. S. Reeves, X. Wang and P. Ning, "Interval-based flow watermarking for tracing interactive traffic," Journal of Computer Networks, Vol. 56, No. 5, pp. 1646- 1665, January 2012

## **Appendix** A

This appendix provides the details of the proof of the constancy of Inequation (25).

Let r be an integer indicating the redundancy degree, and $r \geq 1$. Let k be an integer indicating the number of packet splitting, and $k \geq 0$. Let m be an integer indicating the number of bits of watermark, and $m \geq 1$.

The reduction process of the inequality is as follows:

$$\frac{r - r \times \left(1 - \frac{1}{r}\right)^k}{\sqrt[k]{m^{r - r \times \left(1 - \frac{1}{r}\right)^k}}} \leq r \times \left(1 - \frac{1}{r}\right)^k$$

Both sides of the inequality eliminate r at the same time.

$$\frac{1 - \left(1 - \frac{1}{r}\right)^k}{\sqrt[k]{m^{r - r \times \left(1 - \frac{1}{r}\right)^k}}} \leq \left(1 - \frac{1}{r}\right)^k$$

Reduce the fraction to a common denominator and transpose.

$$1 - \left(1 - \frac{1}{r}\right)^k - \sqrt[k]{m^{r - r \times \left(1 - \frac{1}{r}\right)^k}} \times \left(1 - \frac{1}{r}\right)^k \leq 0$$

Transpose and unite like terms.

$$1 \leq \left(1 + \sqrt[k]{m^{r - r \times \left(1 - \frac{1}{r}\right)^k}}\right) \times \left(1 - \frac{1}{r}\right)^k$$

As $0 < \left(1 - \frac{1}{r}\right)^k$, the following inequality can be obtained by using the inequality zoom:

$$1 \leq \sqrt[k]{m^{r - r \times \left(1 - \frac{1}{r}\right)^k}} \times \left(1 - \frac{1}{r}\right)^k$$

As $0 < \frac{k}{r - r \times \left(1 - \frac{1}{r}\right)^k}$, the following inequality can be obtained by multiplying both sides by $m^{-\frac{k}{r - r \times \left(1 - \frac{1}{r}\right)^k}}$:

$$\sqrt[k]{m^{r \times \left(1 - \frac{1}{r}\right)^k - r}} \leq \left(1 - \frac{1}{r}\right)^k$$

Both sides of the equation eliminate the exponent k at the same time.

$$\sqrt[1]{m^{r \times \left(1 - \frac{1}{r}\right)^k - r}} \leq \left(1 - \frac{1}{r}\right)$$

As $1-\frac{1}{r}<1$, there is the following inequality:

$$m^{\frac{1}{r\times\left(1-\frac{1}{r}\right)^{k}-r}}<1$$

As $m^0=1$ and $1\le m$, there is the following inequality:

$$\frac{1}{r\times\left(1-\frac{1}{r}\right)^{k}-r}<0$$

It would be easy to conclude the following inequality:

$$r\times\left(1-\frac{1}{r}\right)^{k}<r$$

Both sides of the inequality eliminate r at the same time.

$$\left(1-\frac{1}{r}\right)^{k}<1$$

For all positive integers k, this inequality is always true.
When it is equal to zero , Inequation (25) can be written as the following inequality:

$$\frac{r}{m^0}\le r$$

Thus, the conclusion that Inequation (25) always stands up can be drawn.