Watermark Based Tor Cross-Domain Tracking System for Tor Network Traceback

Jianwei Ding(✉) and Zhouguo Chen

30th Research Institute of China Electronics Technology Group Corporation, No. 8 Adventure Road, High-tech Zone, Chengdu, China
mathe_007@163.com, czgexcel@163.com

Abstract. Anonymous network is widely used to access the Internet, causing varieties of cyber security incidents because of its anonymity, which increasingly affects the security of cyberspace. How to detect anonymous network flow to position the anonymous users, is becoming to a research hotspot. However, with rapid development of the encryption and network technology, it is a nontrivial task to detect and position the anonymous user in such a complex network environment.

In this paper, we design a prototype system called Watermark based Tor Cross-domain Tracking System that is effectively detects and determine the sender and the receiver on the real Tor network to testify its function. Moreover, instead of conventional passive network flow analysis, this paper learns from active network flow analysis to design three digital watermark models to implement the embedding, extracting and matching of watermark information, and meanwhile it will not affect the network flow's content and transmission. Experimental results on the real data sets show that when embedding the three watermark models on the sender, watermark based Tor cross-domain tracking system indeed yields the positioning function.

Keywords: The router onion · Watermark model · Tracking system

1 Introduction

In recent years, more and more anonymous networks appears with the rapid development of the Internet and encryption technology. After the emergence of anonymous networks such as the router onion (Tor), users can communicate anonymously on the Internet through anonymous networks, which also brings network security issues [13,25]. Using anonymous communication systems, cyber attackers can hide their identities. Cyber attackers usually join multiple intermediate springboard hosts into anonymous networks and use these springboards

Supported by: National Key R&D Program of China (No. 2016YFE0206700); Sichuan Science and Technology Program, NO. 2018HHO115.

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 54–73, 2021.
https://doi.org/10.1007/978-3-030-66922-5_4

to attack, intended to make network tracing and network supervision more difficult, which not only threatens the privacy of users, but also causes users to suffer economic losses.

The traffic of anonymous network is encrypted, which make it difficult to analyze. In this case, traditional intrusion detection technology has obvious shortcomings. Conventional passive network traffic analysis [4,6,19] can not confirm the communication relationship between two parties in communication, track anonymous attackers or find intermediate proxy hosts, which is difficult to be applied in a large-scale, high-bandwidth network environment, such as Tor. The controlled environment communication entity correlation technology that incorporates the idea of digital watermarking into network traffic analysis [17] can be applied to a variety of network environments, and has the advantage of high detection rate, low false alarm rate, strong concealment and short detection time.

This paper applies correlation positioning technology to track the illegal users in the Tor, which applies actively traffic analysis technology to track users who attack a certain party through a series of intermediate springboard hosts or users who communicate illegally through anonymous channels. It draws on the idea of digital watermark technology [8–10,24], and embeds watermark information by actively changing certain characteristics of the network flow (such as packet length, network flow rate, or packet sending time interval, etc.) generated by suspicious senders. The watermark information extracted by the suspicious receiver is compared with the original watermark information. If a certain detection rate can be achieved, it can be determined that there is a communication relationship between the sender and the receiver. At the same time, this paper designs a cross-domain collaborative tracking architecture based on network watermarks, support hidden signal detection, tracking and positioning, and path construction, in response to the problem that watermark transmission may cross multiple autonomous domains (ASs), construct a cross-domain collaborative flow watermark tracking architecture.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 gives the design details of our cross-domain collaborative tracking architecture based on network watermarks. In Sect. 4, we design three kinds of watermark models, including IPD model, IW model and IWG model. Section 5 presents experimental results to validate our approach. Section 6 gives the conclusions of the research.

2 Related Work

The onion router (Tor) [5] is a world-renowned anonymous network. The core of the Tor network's anonymity is rerouting technology. After multi-layer routing and forwarding, each layer of routing only knows the upper-level node, so it is difficult to trace the source of the traffic. In addition, the bridge protocol will disguise its source as other data traffic, intended to distinguish the third party from the content of the message.

Hence, there are many studies on the traffic analysis of Tor [2,3,22], whose idea mostly is that analyzing the Tor's protocol and then extracting characteristics from network flow to train a machine learning model like SVM for Tor traffic analysis [14,26]. However, with Tor protocol's upgrading, pluggable transmission protocol, such as Meek, obfsproxy and FTE, are introduced into Tor. With the introduction of protocol hiding technology, the previous fingerprint characteristics methods are no longer applicable. Based on the new generation of Tor protocol, there are also some research studies on the Tor traffic analysis [7,15,21], but the recognition rate is not high.

There are a few studies on traceability technology, which are also affected by the pluggable transmission protocol. At present, there is no research on the traffic tracking technology for the Tor protocol. Traditional traffic tracking technologies [27] are generally divided into active traffic analysis and passive traffic analysis. Passive traffic analysis [1] will not have any impact on the anonymous communication process. It only observes the communication process to infer the relationship between users. Active traffic analysis [11,12,16,18,20,28] is mainly to artificially interfere with the traffic, and to achieve neither the purpose of exposure nor the means of intervention. There are some typical digital watermark model, such as packet transmission rate based watermark, inter packet watermark and packet sending interval watermark.

3 Watermark Based Tor Cross-Domain Tracking System

Watermark based Tor Cross-domain Tracking System is illustrated in Fig. 1. The overall structure of the Tor tracking system is based on the idea of SDN's (software defined network) data and control separation. SDN switches do not run any protocol between them, which are only responsible for forwarding data packets, and it is controlled entirely by the upper controller. Developer is able to customize routing decisions and transmission rules in the controller.

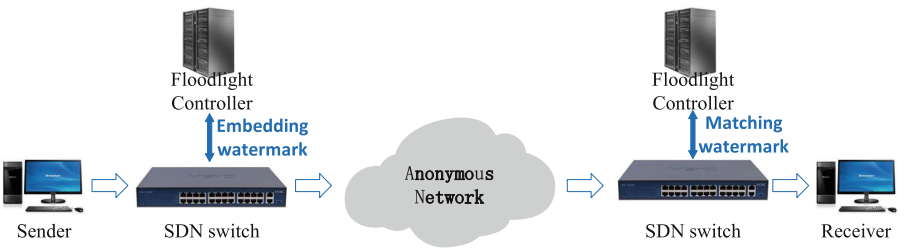

Fig. 1. Architecture of Watermark based Tor Cross-domain Tracking System.

Hence, this paper adopts the secondary development of the Floodlight controller, intended to analyze OpenFlow protocol and issue flow table. The SDN switch sends the matching data packet to the controller by matching the flow

table. The controller embeds digital watermark into the target flow and then sends flow's data packets back to the SDN switch in the sender. The target flow passes through Tor network, and the controller matches digital watermark of target flow in the receiver. This architecture design only needs to modify the codes on the controller, and does not need to modify the OpenFlow protocol in the underlying switch, which make the implementation is more flexible.

The Floodlight controller communicates with the SDN switch through the OpenFlow protocol. The controller can add, update, and delete flow table. The switch selects and processes data packets according to flow table. OpenFlow protocol supports three message types in the workflow of the entire architecture, which include Flow-mod messages, Packet-in messages, and Packet-out messages. The controller uses the Flow-mod message to deliver flow table to the switch for updating flow table. If the data packet and flow table are inconsistent or the operation defined in the flow table is to forward the data packet to the controller, the switch sends a Packet-in message to hand the data packet processing right to the controller. The controller uses Packet-out messages to send data packets to the switch via the data channel, and hands the operation to the switch for execution.

All message processing modules in Floodlight controller need to listen to OpenFlow messages, and these modules must follow a certain calling sequence. After each module processes them in sequence, the packet-in message will eventually reach the forwarding module, which is the key to processing the forwarding of data packets between the sending device and the receiving device. Therefore, we modify the relevant code for processing data packets in the forwarding module to implement the digital watermark embedding and detection functions. The work process of watermark based Tor cross-domain tracking system is illustrated in Fig. 2.

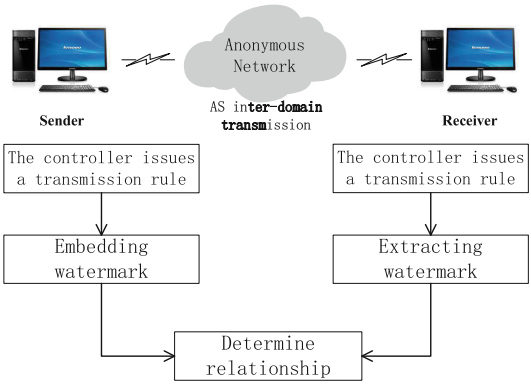

Fig. 2. Work Process of Watermark based Tor Cross-domain Tracking System.

4 Design of Digital Watermark Model

In this paper, we refer to the previous watermark model to design Tor's watermark model in the watermark based Tor cross-domain tracking system. The design of watermark model need to follow the principle of not affecting the contents of the packet. Hence, we design three watermark models, that is inter packet delay based watermark model, interval based watermark model and interval gravity based watermark model.

4.1 Inter Packet Delay Based Watermark Design

The Inter Packet Delay (IPD) based watermark model encode watermark by fine-tuning the timing of the selected packet. It is necessary to ensure that there are enough packets in the watermarked network flow, and the watermark is embedded only on the selected IPD. In order to make it difficult for an attacker to detect the existence of a watermark without knowing the IPD selection function and other watermark embedding parameters, it is necessary to obtain an IPD on the basis of randomly selecting a packet set and randomly pairing.

Watermark Model. Suppose that network flow f has $n(n > 1)$ packets, and $<P_i, P_j>$ $(0 \leq i < j \leq n-1)$ are two successive packet pair on the embedded side. The sending time of packet P_i and P_j is t_i and t_j respectively. The sending time interval $dipd_{i,j}$ of P_i and P_j is calculated as follows:

$$dipd_{i,j} = t_j - t_i \tag{1}$$

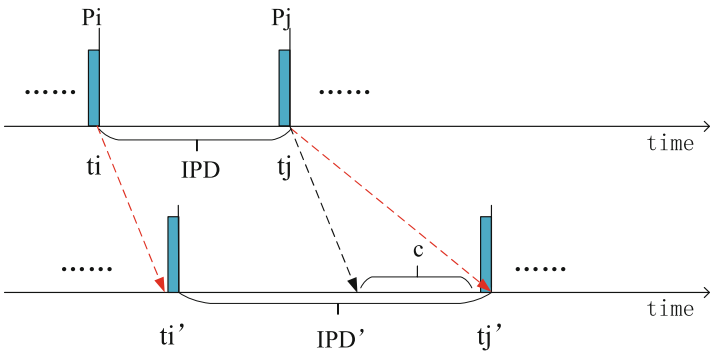

Fig. 3. Schematic diagram of embedded watermark information bits in one IPD.

For one inter-packet delay to be embedded one watermark bit, we need to add extra delay before sending the packet P_i. The watermark bit embed equation is shown as follows:

$$dipd'_{i,j} = dipd_{i,j} + c \tag{2}$$

where $dipd'_{i,j}$ is the time interval after embedding watermark, and the extra delay c is calculated as follows:

$$c = \left(round\left(\frac{dipd_{i,j}}{s} + \frac{1}{2}\right) + \left(w - round\left(\frac{dipd_{i,j}}{s} + \frac{1}{2}\right) \% \ 2 + 2\right) \%2\right) * s \tag{3}$$

where w is the binary watermark information bit '1' or '0', and s is the selected reference time length. The larger the value of s, the less the watermark bit is embedded, but the longer it takes. A schematic diagram of embedding watermark information bits in a single IPD is illustrated in Fig. 3.

Because of the randomness for embedding watermark bits in one inter-packet delay, the success rate of watermark embedding is not high, and it is easily affected by network disturbances. Therefore, the distribution of IPDs carrying watermarks in the longer duration of the network flow is considered. A single watermark bit is embedded in the average of multiple IPDs. As shown in formula (4), in addition to the reference time length s, the number of data packet pairs (redundancy amount) r needs to be selected.

$$dipd_{avg} = \frac{1}{r}\sum_{k=1}^{r} dipd_k \tag{4}$$

$<P_{i,k}, P_{j,k}>$ $(0 \leq< j \leq n-1, 1 \leq k \leq r)$ is the k-th packet used to embed the watermark bit, and the packet $P_{i,k}$ is sent $t_{i,k}$, and packet $P_{j,k}$ is sent $t_{j,k}$, where $dipd_k = t_{j,k} - t_{i,k}$. A schematic diagram of embedding watermark information bits in multiple IPDs is shown in Fig. 4.

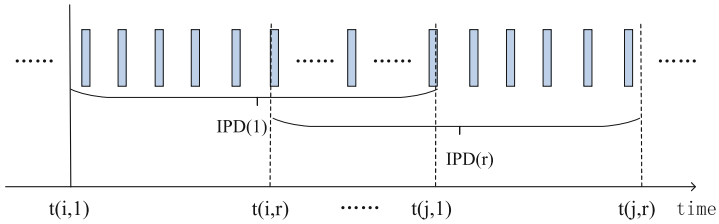

Fig. 4. Schematic diagram of embedded watermark information bits in multiple IPDs.

For the watermark information fw with a watermark bit number of l, $(l+1)*r$ random packets can be selected. Applying l times to embed l-bit watermark information in the selected network flow f.

Watermark Detection. Watermark detection is the process of determining whether a given watermark information is embedded in the IPD of the selected Flow. For the network flow f with l-bit watermark information arriving at the detection end, $<P_i, P_j>$ are two packets that have arrived one after another,

and the arrival time of packet P_i is $t_i\prime$, and the arrival time of packet P_j is $t_j\prime$. The interval $aipd_{i,j}$ between the arrival of P_j and P_i is calculated as shown as follows:

$$aipd_{i,j} = t_j\prime - t_i\prime \tag{5}$$

The watermark information is extracted according to the watermark bit decoding function w' is calculated as follows:

$$w' = round\left(\frac{aipd_{i,j}}{s}\right)\%2 \tag{6}$$

Where w' is the extracted one-bit binary watermark information bit '1' or '0'.

Let the l-bit binary watermark information decoded from the watermark flow f be fw', and set the error threshold $h(1 \leq h \leq l)$. Compare fw' with the original watermark information fw. If fw' and fw have different digits less than or equal to h, then the watermark information fw can be considered to be detected in the flow f, so that it can be determined that there exists communication relationship between the receiver and the sender.

It should be noted that if the value of h is set smaller, the watermark detection rate is lower; if the value of h is set too large, even if the detection result is also with the error range, an un-watermarked network flow may be misunderstood that there exists communication relationship between the receiver and the sender. This situation is called system false positive. Hence, the error threshold h needs a tradeoff to satisfy both a higher watermark detection rate and a lower watermark false positive rate.

4.2 Interval Based Watermark Design

The Interval based Watermark (IW) model uses the characteristics of the invariable duration of the network flow to divide the duration of the selected network flow into fixed-length intervals, and adjusts the number of packets in a specific interval according to the watermark information bits to be embedded to achieve the embedded watermark information bit. In this design, the time interval is used as the carrier, so that the watermark is not affected by the change of the number of packets.

Watermark Model. Suppose that, a random time offset σ is set from the beginning of the selected network flow f, and it is watermarked after the time ? has elapsed. Divide the network flow into multiple time intervals I_i of length T. Each I_i contains $X_i(0 \leq i \leq n-1)$ consecutive packets. These packets X_i is independent and identically distributed with respect to the network flow f.

For the binary watermark information fw with a watermark bit number l, divide every three intervals of the flow f into one group. Randomly select n consecutive pairs of intervals, and randomly assign these consecutive pairs of pairs so that each r group of consecutive intervals is used in pairs. To encode a bit of watermark, where $r = \frac{n}{l}$.

The random allocation strategy for n sets of consecutive interval pairs is that: the $<3i, 3i+1, 3i+2>$, $<3(i+l), 3(i+l)+1, 3(i+l)+2>$, $<3(i+2l), 3(i+2l)+1, 3(i+2l)+2>$, ..., $<3[i+(r-1)l], 3[i+(r-1)l]+1, 3[i+(r-1)l]+2>$ consecutive interval pairs are used to encode the i-th watermark bit $(0 \leq i \leq l-1)$.

In order to avoid conflicts between consecutive interval pairs when embedding watermark information bits, every two pairs of interval pairs used to embed watermark information bits need to be inserted as a buffer without using watermark information bits. Therefore, in the selected n sets of consecutive intervals $<I_{i,j,1}, I_{i,j,2}, I_{i,j,3}>$ $(0 \leq i \leq l-1, 0 \leq j \leq r-1)$, $I_{i,j,1}$ is the buffer interval, and $<I_{i,j,2}, I_{i,j,3}>$ is used as the i-th watermark of the encoding the j-th group of bits embeds a space pair. Each interval contains $X_{i,j,k} (0 \leq i \leq l-1, 0 \leq j \leq r-1, k = 1, 2, 3)$ consecutive data packets.

Without artificial interference, the arrival times of the data packets are evenly distributed in each time interval, so the exception u of the number of packets contained in each interval is the same. The model chooses to encode each watermark bit as $I_{i,j,2}$ and $I_{i,j,3}$ with a difference in the number of packets $Y_{i,j} (0 \leq i \leq l-1, 0 \leq j \leq r-1)$, as shown as follows:

$$Y_{i,j} = \frac{X_{i,j,2} - X_{i,j,3}}{2} \tag{7}$$

The average of all the r packet amount deviations $Y_{i,j}$ for encoding the i-th watermark bit $\bar{Y_{i,r}}$ is calculated as follows:

$$\bar{Y_{i,r}} = \frac{1}{r} \sum_{j=0}^{r-1} Y_{i,j} \tag{8}$$

The expectation of the number of packets in each interval is u, and the number of packets X_i is independent and identically distributed with respect to the network flow f, the expectation of $\bar{Y_{i,r}}$ is calculated to be 0. Therefore, the binary watermark information bit '0' or '1' can be encoded through increasing or decreasing $\bar{Y_{i,r}}$ by u, that is, by adjusting the number of packets $X_{i,j,2}$ and $X_{i,j,3}$ within $I_{i,j,2}$ and $I_{i,j,3}$ respectively to makes the distribution of $\bar{Y_{i,r}}$ shift by u to the right or the left.

When the watermark to be embedded is '0', increasing $\bar{Y_{i,r}}$ by u can be achieved by increasing $Y_{i,j}$ by u, that is, increasing r pieces of $X_{i,j,2}$ by u and decreasing r pieces of $X_{i,j,3}$ by u. The former is accomplished by adding packets to the interval $I_{i,j,2}$: adding a delay T to all packets $X_{i,j,1}$ in the buffer interval $I_{i,j,1}$ of the current interval pair, and moving all packets in the interval $I_{i,j,1}$ to the current interval $I_{i,j,2}$. The latter is accomplished by clearing all packets in the interval $I_{i,j,3}$: adding a delay T to all packets $X_{i,j,3}$ in the current interval $I_{i,j,3}$, and moving all packets in the interval $I_{i,j,3}$ to the buffer interval $I_{i,j,1}$ of the next interval pair. When the watermark to be embedded is '1', $\bar{Y_{i,r}}$ is reduced by u, the method is the opposite of the above. The schematic diagram of the embedded watermark information bit is shown in Fig. 5.

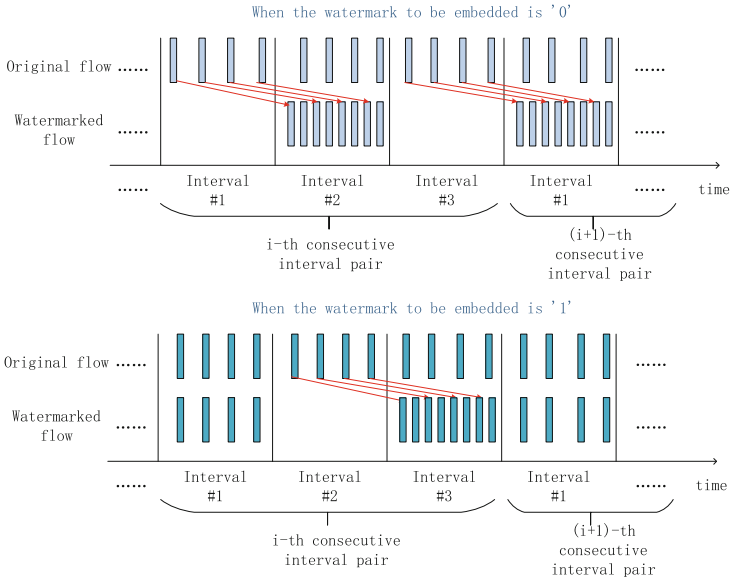

Fig. 5. Schematic diagram for embedding watermark in the IW model.

Watermark Detection. For the watermark flow f embedded with l-bit watermark information arriving at the detection end, by recording the number of packets $X_{i,j,2}$ and $X_{i,j,3}1, 0 \leq j \leq r-1)$ for the embedding interval pair $<I_{i,j,2}, I_{i,j,3}>$ $(0 \leq i \leq l-1, 0 \leq j \leq r-1)$, the i-th watermark bit is extracted according to the following strategy:

- Step 1: Let $w_{i,0} = 0$, and calculate the average value of the number of packets $\bar{X_{i,j,2}}$ and $\bar{X_{i,j,3}}$ in the embedding interval pair as follows:

$$\bar{X_{i,j,2}} = (j+1)^{-1} \sum_{k=0}^{j} X_{i,k,2} \tag{9}$$

$$\bar{X_{i,j,3}} = (j+1)^{-1} \sum_{k=0}^{j} X_{i,k,3} \tag{10}$$

- Step 2: When $\bar{X_{i,j,2}} > \bar{X_{i,j,3}}$, let $w_{i,j} = 0$;
- Step 3: When $\bar{X_{i,j,2}} < \bar{X_{i,j,3}}$, let $w_{i,j} = 1$;
- Step 4: When $j > 1$ and $\bar{X_{i,j,2}} = \bar{X_{i,j,3}}$, let $w_{i,j} = w_{i,j-1}$;
- Step 5: Return to Step (1), repeat r times, and finally get $w_{i,r-1}$ is the i-th watermark bit obtained by decoding.

Repeat the above strategy l times to decode the complete l-bit binary watermark information fw' from the watermark flow f. Set the error threshold $h(1 \leq h \leq l)$ and compare fw' with the original watermark fw. If the number of different bits

of fw' and fw is less than or equal to the error threshold h, then It is determined that there is a communication relationship between the receiver and the sender; otherwise, it can be considered that there is no communication relationship between the receiver and the sender.

4.3 Interval Gravity Based Watermark Design

According to the characteristics of the invariable constant duration of the network flow, the Interval Gravity-based Watermark (IGW) model divides the duration of the selected network flow into fixed-length intervals, and adds delay to change the time of the packet arrival interval, intended to make the time distribution gravity of the packet arrival time shift to achieve the purpose of embedding the watermark information bits.

Watermark Model. Suppose that there is a random time offset σ for a given network flow f, and the constant duration is defined as T_d after σ. There are X packets added watermark in this selected network flow, and he time stamp of the starting point of the watermark is t_0.

For binary watermark information fw with a watermark bit number of I, T_d is divided into $2n$ intervals with I_i of length T, and each I_i contains X_i consecutive packets. The sending timestamp of the data packet $P_{i,j}(0 \leq i \leq 2n-1, 0 \leq j \leq X_{i-1})$ is $t_{i,j}$, which is time-lag relative to the start point of the interval I_i. The time offset of $P_{i,j}$ from the starting time in the time interval I_i is shown as follows:

$$\Delta t_{i,j} = \{t_{i,j} - t_0\}\%T \tag{11}$$

Choose n intervals from the $2n$ intervals in the I_i at random to construct a new interval named Interval Group A, denoted as $I_k^A(0 \leq k. \leq n-1)$, and the other n intervals in the I_i form a new interval named Interval Group B, denoted as I_k^B. Then randomly assign intervals for groups A and B respectively, intended to make every $2r$ intervals to build a watermark bit, where $r = \frac{n}{l}$.

The randomly assignment strategy of the $2n$ intervals is that: set $x(0 \leq x \leq 2n-1)$ to denoted as interval number, and then choose number $i?i + l?i + 2l????i + (2r-1)l$ interval to encode the i-th $(0 \leq i \leq l-1)$ watermark bit respectively. Assign the x-th interval to Interval Group A if $\frac{x-i}{l}\%2 = i\%2$; otherwise, assign the x-th interval to Interval Group B.

$I_{i,j}^A$ and $I_{i,j}^B$ are represented as the j-th $(i \leq j \leq r-1)$ interval in the i-th$0 \leq i \leq l-1$ encoded watermark bit for Interval Group A and B respectively. $X_{i,j}^A$ and $X_{i,j}^B$ are the packet amount for the interval $I_{i,j}^A$ and $I_{i,j}^B$ respectively, and X_i^A and X_i^B represent the packet amount in the i-th encoded watermark bit respectively.

$$X_i^A = \sum_{j=0}^{r-1} X_{i,j}^A \tag{12}$$

$$X_i^B = \sum_{j=0}^{r-1} X_{i,j}^B \tag{13}$$

According to the Eq. (11), we can calculate the time offset $\Delta t_{i,j,k}^A$ and $\Delta t_{i,j,k}^B (0 \le i \le l-1, 0 \le j \le r-1, 0 \le k \le X_{i,j}-1)$ for the k-th packet of the interval $I_{i,j}^A$ and $I_{i,j}^B$ respectively. Aggregate r timestamps in interval Group $A$and B respectively, and the time offset gravity of packet for Interval Group A and B is respectively calculated as follows:

$$A_i = \frac{\sum_{j=0}^{r-1} \sum_{k=0}^{X_{i,j}^A - 1} \Delta t_{i,j,k}^A}{X_i^A} \tag{14}$$

$$B_i = \frac{\sum_{j=0}^{r-1} \sum_{k=0}^{X_{i,j}^B - 1} \Delta t_{i,j,k}^B}{X_i^B} \tag{15}$$

As the time offset of the arrival interval for the packet $P_{i,j,k}$ evenly distributed over $[0, T)$, then we can calculate the time offset gravity of the packet arrival interval is $\frac{T}{2}$ for Group A and B respectively. Hence, the interval based watermark assignment chooses the time offset gravity deviation of each pair encoded watermark bit A_i and B_i, denoted as follows:

$$Y_i = A_i - B_i \tag{16}$$

According to adjust the time offset gravity of Group A or B, we can make the distribution of Y_i pan right or pan left to embed binary watermark information bit 1 or 0. Suppose that the maximum manual adding delay $c (c < c < T)$, when the encoded binary watermark bit is 1, then we can make the distribution of Y_i pan right $\frac{c}{2}$ by adding A_i, which means that manual adding extra delay $c_k (0 < c_k < c)$ for each packet $P_{i,j,k} (0 \le i \le l-1, 0 \le j \le r-1, 0 \le k \le X_{i,j}^A - 1)$ in the r intervals $I_{i,j}^A$. The packet $P_{i,j,k}$ delay strategy is follow the equation shown as follows:

$$\Delta t_{i,j,k}^{A\,\prime} = c + \frac{T}{T-c} \Delta t_{i,j,k}^A \tag{17}$$

where $\Delta t_{i,j,k}^{A\,\prime}$ represents the time offset of the corresponding interval after adding delay, and its calculation is shown as follows:

$$\Delta t_{i,j,k}^{A\,\prime} = (\Delta t_{i,j,k}^A - t_0 + c_k) \% T \tag{18}$$

Then the adding extra delay c_k for each packet $P_{i,j,k}$ is calculated as follows:

$$c_k = c - \frac{c}{T} [(\Delta t_{i,j,k}^A - t_0) \% T] \tag{19}$$

After adding delay, the time offset of the arrival interval for the packet $P_{i,j,k}$ evenly distributed from $[0, T)$ to $[c, T)$, illustrated as Fig. 6. The time offset gravity of the arrival interval in Group A_i is $\frac{T+c}{2}$, and the distribution of Y_i pans right $\frac{c}{2}$, intended to embed the binary watermark information bit '1'.

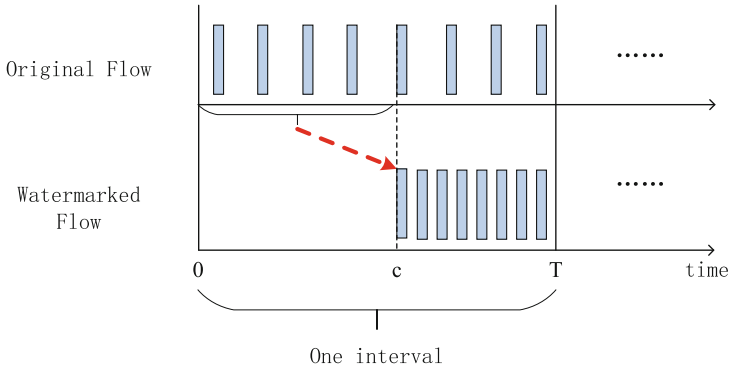

Fig. 6. Schematic diagram for altering the distribution of the packet arrival time.

Similarly, when the watermark bit to be encoded is '0', then we can make the distribution of Y_i pan left $\frac{c}{2}$ by adding B_i, which means that manual adding extra delay $c_k (0 < c_k < c)$ for each packet $P_{i,j,k} (0 \leq i \leq l-1, 0 \leq j \leq r-1, 0 \leq k \leq X^B_{i,j} - 1)$ in the r intervals $I^B_{i,j}$, intended to embed the binary watermark information bit '0'.

Watermark Detection. For the watermark flow f embedded with l-bit watermark information arriving at the detection end, we can record the arrival time offset of all the packets in every interval to calculate the arrival time offset gravity of Group A and B respectively, and then calculate the gravity deviation Y_i.

If Y_i is larger than 0, then the binary watermark information bit is determined to be '1'; otherwise, the binary watermark information bit is determined to be '0'.

Suppose that the complete l-bit binary watermark information decoded from the watermark flow f is fw', and at meanwhile the error threshold is set to $h (1 \leq h \leq l)$. Compare fw' with the original watermark information fw, if fw' and fw are not the same number of bits less than or equal to the error threshold h, it can be determined that there is a communication relationship between the receiver and the sender; otherwise, It can be considered that there is no communication relationship between the receiver and the sender.

5 Experimental Results

5.1 Experiment Design

The implementation of watermark based Tor cross-domain tracking system needs to reduce the possibility of watermark information being discovered, that is, to achieve high concealment, without affecting the service quality of the user as much as possible. Therefore, in the experiment, the controller of the watermark embedding end performs protocol analysis on the arriving data packets, and only the watermark embedding is performed on the TCP data packets. Depending on the selected watermark model, watermark the data packets returned from the server to adjust the server-side traffic, or add watermarks to the data packets sent from the client to adjust the client-side traffic. It ensures that the loading time of the webpage can be affected as little as possible when the watermark is embedded, thereby ensuring the user's access efficiency and improving the concealment of the watermark.

In the experiment, the three designed watermark models and detection schemes are respectively applied to watermark based Tor cross-domain tracking system, and the Eclipse software is used in the Floodlight controller to implement the programming in Java. In the experiment, a file with a size of 1G was placed in the Apache webpage set up by the server, and the client generated traffic by using the wget command on the terminal to download the file.

The experiment applies detection precision rate and false alarm rate [23] to estimate the results for the three watermark models. Meanwhile, the three watermark models are tested in an experimental environment without going through the intermediate springboard host, and the client directly accesses the server.

5.2 Results Analysis of IPD Model

Set the length l of the watermark information to 20 bits, the size of the watermark information bit interval to 2, the number of delayed packets embedded between different watermark bits to 5, and the error threshold h to 3. Adjust the value of the redundancy amount r and the additional delay time c to perform multiple experiments on the network flow. The shortest test duration is about 2 min.

First, test the effect of the additional delay c on the watermark detection precision rate and the system false alarm rate. Set the number of redundancy $r = 10$ and adjust the additional delay c. The test results are shown in Table 1.

Table 1. Effect of additional delay c on the watermarking system ($r = 10$).

Additional delay c(ms)	$c = 5$	$c = 10$	$c = 15$	$c = 25$	$c = 50$
Detection precision rate	84.10%	92.70%	93.20%	93.70%	94.00%
False alarm rate	9.10%	6.60%	7.30%	6.10%	4.10%

According to the experimental result shown in Table 1, when the redundancy amount r is Invariant, the detection precision rate will increase as the additional delay c increases, and the false alarm rate will decrease as the additional delay c increases. When error threshold h is set increased, both detection precision rate and false alarm rate will increase. Hence, the system need to set a reasonable error threshold to make that detection precision rate reach to 100% and false alarm rate can ben guaranteed to be below 5%.

Table 2 and 3 show the effect of redundancy amount r on the detection precision rate and the false alarm rate. The error threshold is set to $h = 0$, and additional delays $c = 5\,\text{ms}$ and $15\,\text{ms}$, respectively, and adjust the number of redundancy r.

Table 2. Effect of redundancy amount r on the watermarking system ($c = 5\,\text{ms}$)

Redundancy amount r	$r = 5$	$r = 10$	$r = 15$	$r = 20$	$r = 25$
Detection precision rate	80.50%	84.10%	90.00%	98.10%	99.00%
False alarm rate	7.20%	6.10%	4.30%	4.10%	3.90%

Table 3. Effect of redundancy amount r on the watermarking system ($c = 15\,\text{ms}$).

Redundancy amount r	$r = 5$	$r = 10$	$r = 15$	$r = 20$	$r = 25$
Detection precision rate	94.70%	93.20%	97.70%	99.30%	98.90%
False alarm rate	4.20%	3.30%	4.00%	2.00%	3.10%

According to the experimental result shown in Table 2 and 3, when the additional delay c is Invariant, as the redundancy amount r increases, the detection precision rate basically increases and the false alarm rate basically decreases. When $c = 5\,\text{ms}$ and the error threshold h is set to 11, the detection precision rate can reach to 100%, but the false alarm rate will also reach to 10%. When setting the error threshold to $h = 5$, the detection precision rate is above 90% and the false alarm rate is guaranteed to be below 5%. When $c = 15\,\text{ms}$ and the error threshold $h = 3$, detection precision rate reach to 100% and the false alarm rate is below 5%.

5.3 Results Analysis of IW Model

Set the length of the watermark information to $l = 24$ bits, the time offset to $\sigma = 6\,\text{s}$, the redundancy amount to $r = 12$, and the error threshold to $h = 3$. Adjust the value of the time interval length T to perform multiple experiments on the network flow, and test the effect of the length of the time interval T on the detection precision rate and the false alarm rate. The results are shown in

Table 4. As T increases, the test duration also increases, and the test takes about 15 min at one time.

Table 4. Effect of time interval T on the watermarking system.

Time interval T(ms)	$T = 10$	$T = 50$	$T = 200$	$T = 500$
Detection precision rate	65.20%	83.50%	100%	100%
False alarm rate	12%	3%	1%	1%

Analyzing the results in Table 4, it is known that with the increase of the time interval T, the detection precision rate is increasing, and the false alarm rate is decreasing. When the time interval T is set to 50 ms, setting the error threshold $h = 5$ can make the detection precision rate reach to 100% and the false alarm rate is lower than 1%.

The above test is performed without the system passing through the intermediate springboard host. The client directly accesses the server. Both are located on the same network segment, and the delay of network interference is very small. Therefore, in the experiment, a network interference delay D was artificially added to test the influence of the network interference delay on the detection precision rate and the false alarm rate. The fixed time interval is set to $T = 200$ ms.

Table 5. Effect of network interference delay D on the watermarking system.

Network interference delay D	$D = 0$	$D = 50$	$D = 100$	$D = 200$	$D = 300$
Detection precision rate	100%	100%	87.50%	58.30%	43.50%
False alarm rate	1%	3%	6%	15%	27%

Analysis of the results in Table 5 shows that when the time interval T is Invariant, as the network interference delay D increases, the detection precision rate decreases and the system false alarm rate increases. In particular, when the network interference delay D reaches the same as or exceeds the time interval T, the detection precision rate and false alarm rate of the system will be greatly affected.

The experiment also simply tested the effect of the length l and the redundancy amount r of the watermark information on the detection precision rate and the false alarm rate of the system. The results show that the larger the redundancy amount r, the higher the detection precision rate and the lower the false alarm rate. The length l of the watermark information has no significant effect on the system. But the larger both r and l, the longer the test duration of the experimental process.

5.4 Results Analysis of IWG Model

Set the length of the watermark information to $l = 32$ bits, the time offset to $\sigma = 10$ s, the redundancy amount to $r = 14$, and the error threshold to $h = 3$. Adjust the value of the time interval T and the artificially added maximum delay c to perform multiple experiments on the network flow. The test results are shown in Table 6. The test time will increase with the increase of T, and the test time is about 10 min.

Table 6. Effect of time interval T and maximum delay c on the watermarking system.

Interval T(ms)	$T = 10$		$T = 50$		$T = 200$		$T = 500$	
Max delay c(ms)	5	7	25	35	80	150	200	350
Precision rate	75.20%	79.50%	92.10%	94.30%	98.10%	100%	100%	100%
False alarm rate	4.80%	3.60%	3.10%	3.00%	1.60%	2%	0.70%	0.80%

The results in Table 6 shows that, as the time interval T increases, the detection precision rate increases, and the false alarm rate decreases. When the time interval T is invariant, the larger the ratio of the time interval T occupied by the maximum delay c, the higher the detection precision rate. When the time interval is set to $T = 50$ ms, setting the error threshold $h = 5$ can make the detection precision rate reach to 100% and the false alarm rate drops below 3%.

The experiment tested the effect of changing the value of the redundancy amount r on the system when the time interval length is set to $T = 50$ ms and the maximum delay is set to $c = 35$ ms. The test results are shown in Table 7.

Table 7. Effect of redundancy amount r on the watermark system ($T = 50$ ms, $c = 35$ ms)

Redundency amount r	$r = 14$	$r = 16$	$r = 18$	$r = 20$
Detection precision rate	94.30%	95.30%	94.60%	96.80%
False alarm rate	3.00%	2.80%	2.60%	2.70%

The results in Table 8 show that the larger the redundancy amount, the higher the detection precision rate and the lower the system false alarm rate.

The experiment also tested the effect of changing the value of the redundancy number r when the time interval length is set to $T = 500$ ms and the maximum additional delay is set to $c = 350$ ms. The test results were that the watermark detection rate reached to 100% and the false alarm rate is lower than 1%. It shows that when the time interval length T is sufficiently large, the effect of the redundancy number on the system is small. At the same time, the larger the number of redundancy r, the longer the test time required for the experiment.

Table 8. Comparison of three watermark models on the system.

Watermark model	IPD	IW	IWG
Min test time	2 min	15 min	10 min
Average of detection precision rate	about 80%	about 95%	about 95%
Average of false alarm rate	about 10%	about 3%	about 3%
Anti-interference ability	weak	strong	strong

5.5 Results Analysis

Compare the results of the three watermark models in Table 8, there are some conclusions:

- Among the three model, IPD-based watermark model has the simplest embedding and detection methods, the shortest test time, and the number of data packets in the selected network flow is not high. However, the anti-interference ability of the system is weak, and the watermark is not robust. Attackers can easily recover or modify the watermark information.
- Among the three model, IW model and IWG model's average detection precision rate is the highest and the false alarm rate is the lowest. When the time interval length T is longer, the anti-interference ability of the system is stronger, but at the same time, the service quality of the user is reduced. Both two models use a method to bind the watermark to a specific interval, so that the embedded watermark is not affected by changes in the number of data packets. However, these two solutions require a longer test time and require that the selected network flow has enough data packets.

6 Conclusion

In this paper we design watermark based Tor cross-domain tracking system and three watermark models on the Tor network tracking: IPD model, IW model and IWG model. The watermark model and detection system are designed separately. According to the test results, the impact of each watermark models on system performance is analyzed, and the availability of Tor tracking system is verified. The work of this paper is shown as follows:

- Design the watermark based Tor cross-domain tracking system for Tor's network communication entity based on the SDN, so that the control right is completely managed by the upper-level controller, without adding configuration information to the underlying network equipment, which is easy to operate and implement.
- Design three watermark models to design watermark schemes. The system can use different watermark schemes in different network situations, making the system more flexible.

The deficiencies in the watermark based Tor cross-domain tracking system designed in the paper are shown as follows:

- In the network transmission, there is only one intermediate springboard host between the client and the server. The network test environment is relatively simple, and the watermarking scheme is not tested in a more complicated network environment.
- The system uses the client to access the webpage set up by the server to generate traffic. In the three watermarking schemes that require artificial delay, the rate of the user accessing the webpage will be significantly slower, which affects the user's service quality.
- In the scheme that uses the time interval as the watermark carrier, the selected interval time is relatively long, so the number of data packets in the network flow is required, and the system does not have strong concealment.
- The offline detection method is used in watermark detection. Therefore, for a large number of network flows, the system cannot extract watermark information in time, and cannot quickly determine the communication relationship between the sender and receiver to locate the attacker.

In view of the shortcomings of the system designed in this paper, our future work contains the follows aspects: (1) Several intermediate springboard hosts are set up between the client and the server, so the network flow can go through a complex network environment after watermarking. Improve anti-interference ability of the system, the solution based on the test results, and enhance the robustness of the system. (2) To improve the method of generating traffic, we consider adding watermarks to the traffic generated by webpage advertisements, and try to optimize the active network watermark system without affecting users. (3) Improve the watermark detection method, so that the system can quickly determine the communication relationship between the sender and receiver to determine the location of the attacker, based on the large number of incoming network flows.

References

1. Agrawal, D., Kesdogan, D., Penz, S.: Probabilistic treatment of mixes to hamper traffic analysis. In: 2003 Symposium on Security and Privacy, pp. 16–27. IEEE (2003)
2. Cai, X., Zhang, X.C., Joshi, B., Johnson, R.: Touching from a distance: website fingerprinting attacks and defenses. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 605–616 (2012)
3. Cuzzocrea, A., Martinelli, F., Mercaldo, F., Vercelli, G.: Tor traffic analysis and detection via machine learning techniques. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 4474–4480. IEEE (2017)
4. Das, A.K., Pathak, P.H., Chuah, C.-N., Mohapatra, P.: Contextual localization through network traffic analysis. In: IEEE Conference on Computer Communications, IEEE INFOCOM 2014, pp. 925–933. IEEE (2014)
5. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. Technical report, Naval Research Lab Washington DC (2004)

6. He, G.-F., Yang, M., Luo, J.-Z., Zhang, L.: Online identification of tor anonymous communication traffic. Ruanjian Xuebao/J. Softw. **24**(3), 540–556 (2013)
7. He, Y.Z., Li, X., Chen, M.L., Wang, W.: Identification of tor anonymous communication with cloud traffic obfuscation. Adv. Eng. Sci. **49**(2), 121–132 (2017)
8. Hou, X., Chen, Y., Tian, H., Wang, T., Cai, Y.: Network watermarking location method based on discrete cosine transform. In: 3rd International Conference on Materials Engineering, Manufacturing Technology and Control. Atlantis Press (2016)
9. Houmansadr, A., Kiyavash, N., Borisov, N.: Rainbow: a robust and invisible non-blind watermark for network flows. In: NDSS (2009)
10. Iacovazzi, A., Elovici, Y.: Network flow watermarking: a survey. IEEE Commun. Surv. Tutor. **19**(1), 512–530 (2016)
11. Iacovazzi, A., Sarda, S., Elovici, Y.: Inflow: inverse network flow watermarking for detecting hidden servers. In: IEEE Conference on Computer Communications, IEEE INFOCOM 2018, pp. 747–755. IEEE (2018)
12. Iacovazzi, A., Sarda, S., Frassinelli, D., Elovici, Y.: DropWat: an invisible network flow watermark for data exfiltration traceback. IEEE Trans. Inf. Forensics Secur. **13**(5), 1139–1154 (2017)
13. Understanding Node Capture Attacks in User Authentication Schemes for Wireless Sensor Networks. Understanding node capture attacks in user authentication schemes for wireless sensor networks (2020)
14. Lashkari, A.H., Draper-Gil, G., Mamun, M.S.I., Ghorbani, A.A.: Characterization of tor traffic using time based features. In: ICISSP, pp. 253–262 (2017)
15. Lin, Z., Tong, L., Zhijie, M., Zhen, L.: Research on cyber crime threats and countermeasures about tor anonymous network based on meek confusion plug-in. In: 2017 International Conference on Robots & Intelligent System (ICRIS), pp. 246–249. IEEE (2017)
16. Liu, W., Liu, G., Xia, Y., Ji, X., Zhai, J., Dai, Y.: Using insider swapping of time intervals to perform highly invisible network flow watermarking. Security and Communication Networks (2018)
17. Tianbo, L., Guo, R., Zhao, L., Li, Y.: A systematic review of network flow watermarking in anonymity systems. Int. J. Secur. Appl. **10**(3), 129–138 (2016)
18. Luo, X., Zhou, P., Zhang, J., Perdisci, R., Lee, W., Chang, R.K.C.: Exposing invisible timing-based traffic watermarks with backlit. In: Proceedings of the 27th Annual Computer Security Applications Conference, pp. 197–206 (2011)
19. Panchenko, A., Niessen, L., Zinnen, A., Engel, T.: Website fingerprinting in onion routing based anonymization networks. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, pp. 103–114 (2011)
20. Peng, P., Ning, P., Reeves, D.S.: On the secrecy of timing-based active watermarking trace-back techniques. In: 2006 IEEE Symposium on Security and Privacy (S&P 2006), pp. 15-pp. IEEE (2006)
21. Pham, D.V., Kesdogan, D.: Towards a causality based analysis of anonymity protection in indeterministic mix systems. Comput. Secur. **67**, 350–368 (2017)
22. Saputra, F.A., Nadhori, I.U., Barry, B.F.: Detecting and blocking onion router traffic using deep packet inspection. In: 2016 International Electronics Symposium (IES), pp. 283–288. IEEE (2016)
23. Verde, M.F., Macmillan, N.A., Rotello, C.M.: Measures of sensitivity based on a single hit rate and false alarm rate: the accuracy, precision, and robustness of', A z, andA'? Percept. Psychophys. **68**(4), 643–654 (2006)
24. Wang, D., Cheng, H., Wang, P., Huang, X., Jian, G.: Zipf's law in passwords. IEEE Trans. Inf. Forensics Secur. **12**(11), 2776–2791 (2017)

25. Wang, D., Zhang, X., Zhang, Z., Wang, P.: Understanding security failures of multi-factor authentication schemes for multi-server environments. Comput. Secur. **88**, 101619 (2020)
26. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In: 23rd USENIX Security Symposium (USENIX Security 2014), pp. 143–157 (2014)
27. Wang, X., Reeves, D.S.: Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 20–29 (2003)
28. Flows By Non-Blind Watermarking. Enhancing invisibility in network flows by non-blind watermarking (2014)