An Invisible Flow Watermarking for Traffic Tracking: A Hidden Markov Model Approach

Zhongjiang Yao[1,2], Lei Zhang[1,2], Jingguo Ge[1,2], Yulei Wu[3], Xiaodan Zhang[1]

[1]*Institute of Information Engineering, Chinese Academy of Science, Beijing, 100093, China*
[2]*School of Cyber Security, University of Chinese Academy of Science, Beijing, 100049, China*
[3]*College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK*
E-mail: {yaozhongjiang, zhanglei0511, gejingguo}@iie.ac.cn, y.l.wu@exeter.ac.uk, zhangxiaodan@iie.ac.cn

***Abstract*—Flow watermarking is a promising active traffic tracking technology. It helps to establish the correspondence between the sender and the receiver, by embedding watermarks into packets with certain features of active interference traffic. Existing watermarking technologies have several drawbacks, such as the vulnerability to multi-flow attacks, low robustness and invisibility. This paper proposes a new traffic tracking technique based on hidden Markov model, called Hidden Markov State-based Flow Watermarking (HMSFW). HMSFW divides the observation range, i.e., inter-packet time, as Markov states and uses the State Transition Probability Mean (STPM) as the watermark carrier. It then adjusts the STPM of a given time interval according to historical traffic characteristics. The proposed HMSFW not only improves the robustness of embedded watermarks, but also effectively enhances their invisibility. The feasibility and invisibility of HMSFW are validated via extensive experimental results.**

I. Introduction

With cyber fraud and privacy breaches proliferating, Internet users are increasingly aware of the importance of privacy protection. To this end, they use censorship-circumvention systems. However, this kind of system is a double-edged sword [1]. It can indeed help protect users' privacy, but criminals can also use it to circumvent the network supervision and hide their behavior engaged in illegal and criminal activities. The introduction of traffic obfuscation techniques make both edges of the sword sharper. Apart from protecting data privacy, packets are repackaged with *stepping stones* to hide their real sender/receiver when being transmitted over censorship-circumvention systems. Thus, perpetrators can perform malicious attacks without leaving any traffic traces due to multiple stepping stones [2].

Tracing the source of malicious activities or serious security threats is always a challenging research in the field of network security. Recently, active correlation technologies receive more attentions, among which the flow watermarking technology has become the research focus because of its low computational complexity and less traffic requirement.

Although the existing watermarking technology can resist offensive detection, e.g., Multi-Flow Attack (MFA) [3] and Mean-Square Autocorrelation (MSAC) attack [4], they still have not achieved good results in terms of invisibility. The time factor is the main feature that can be used to carry watermarks and is highly susceptible to noise interference. Any slight interference can cause significant changes in time characteristics. Thus, the most urgent problem to be solved in the current watermarking technologies is to improve the invisibility. The invisibility refers to the possibility that the watermark cannot be analyzed by a third-party detector to detect the embedded watermarks.

To this end, we introduce Hidden Markov State-based Flow Watermarking (HMSFW), an invisible watermarking technology based on Inter-Packet Time (IPT). The main contributions of this paper are summarized as follows:

- To achieve greater watermark invisibility, the proposed HMSFW uses the historical traffic characteristics as the gauge to interfere with the current traffic, so that the watermark before and after embedding features very similar traffic characteristics.
- To embed the watermark into current traffic based on the historical traffic characteristics, this paper divides the IPTs value range of a traffic into equal parts as hidden Markov states, and introduces the State Transition Probability Mean (STPM) as the watermark carrier.
- To handle the randomness of STPM as watermark carrier, in this paper a time interval is divided into a prediction part and a watermark part, the STPM distributions of which are mapped into a discrete uniform distribution respectively.
- Based on the public dataset, we design the experiment, and its results show that the statistical characteristics of the observation values after the watermark embedding are not significantly changed. In addition, the experiment shows good detection results on the watermark.

The rest of this paper is organized as follows. The related work on network flow watermarking is investigated in Section II. In Section III, the design of the HMSFW watermarking scheme is described. The proposed HMSFW is evaluated through extensive experiment results in Section IV. Finally, we conclude this paper in Section V.

II. Related Work

Active association is used as a traffic tracking technology to associate the source with destination nodes of a flow. As a promising active association technology, flow watermarking includes the following three categories.

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

A. Flow Rate-Based Watermarking

Yu et al. [5] introduced Direct Sequence Spread Spectrum (DSSS) which is used in CDMA into flow watermarking and proposed a new watermarking technology called DSSS-FW. However, the proposed Mean-Square Autocorrelation (MSAC) attack [4] reduces the practicality of DSSS-FW.

B. Packet Size-based Watermarking

Ling et al. [6] proposed a novel packet size-based watermark technique, in which they repackaged the virtual web object according to *Anonymizer*'s traffic and mapped the *k-ary* symbol to the packet size by *Monte Carlo* sampling. They then modulated the secret message bits to the size of the last packet of these virtual web objects. Although packet size-based watermarking technology can effectively resist the distortion caused by *Anonymizer*, it is not scalable enough.

C. Time Interval-based Watermarking

After careful investigation and analysis, we classify the time interval-based watermarking technology as follows.

a) Packets Count-based Technology: Pyun et al. [7] proposed to use the invariance of two connections, i.e., before and after a flow passes through the stepping stones, and divided the duration of each flow into short fixed length intervals for synchronizing the two connections being compared. Inter-packet time is adjusted to manipulate the packet count in specific intervals, for the purpose of embedding watermarks.

b) Inter-Packet Time-based Technology: Houmansadr et al. [8] presented the RAINBOW, which divides a watermark value into fragmentations as packet jitters within a given time interval, and changes IPTs accordingly. RAINBOW therefore makes a slight delay for each packet. Attacks such as *known/chosen flow* attack [9] and BACKLIT [10] have been tested and verified that watermarks embedded in accordance with RAINBOW can be effectively detected.

c) Interval Centroid-based Technology: Wang et al. [11] first proposed the centroid of IPT in a given time interval as a watermark carrier, and proposed an ICBW watermarking technique accordingly. Although ICBW has better invisibility and anti-jamming ability, MFA can still effectively detect the watermarks with multiple flows. Thus, *Houmansadr et al.* [12] proposed SWIRL using a double-layer slot centroid method. Although SWIRL can defend against MFA, it is vulnerable to attacks such as *known/chosen flow* attack and BACKLIT.

d) Packet Loss-based Technology: Iacovazzi et al. [2] proposed DropWat flow watermarking technology to simulate the packet loss behavior. DropWat converts the simulated binary sequence into packet loss sequence through the offline initialization stage and performs real-time packet loss operation during the online packet loss stage. Although DropWat has better invisibility in a specific network traffic, it is difficult to adapt to real-time changes in network status.

Invisibility and robustness have always been the challenges for the design of flow watermarking technologies. Although the existing flow watermarking technologies are claimed to have good invisibility, follow-up researches show that their invisibility solutions still face serious problems. The purpose of this paper is to propose a new flow watermarking method, which has both high invisibility and strong robustness.

III. The Proposed Network Flow Watermarking

In this section we propose the HMSFW watermarking technology. The basic idea of HMSFW is to depict a series of Inter-Packet Times (IPTs) of a flow f as a hidden Markov state sequence. HMFSW then adjusts the hidden Markov state sequence according to the pre-stored historical flows, which are the same with the flow f, and the state transition probability matrix $\mathbf{\Phi}$ learned from historical flows by machine learning techniques.

In the following subsections, a detailed HMSFW watermarking scheme is presented, including watermark encoding and detection methods.

A. The design of hidden Markov state and STPM

Before designing watermark encoding, it is necessary to propose a *variable* as the watermarks carrier for the packets of a flow transmission. This variable should be consistent during the packet forwarding process, even though significant transformations occur during the flow transmission, such as packet dropping, flow mixing and flow splitting/merging. The variable as a watermark carrier can be adjusted according to some specific rules. In what follows, the design of this variable will be presented.

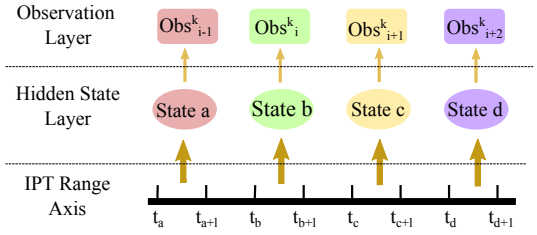

Fig. 1. Hidden Markov State Mapping.

As the packet IPT of a flow will not be changed by censorship-circumvention systems [7], we consider this feature in the design of this variable. The hidden states of Hidden Markov Model (HMM) will be used to characterize packet IPTs. Due to the nature of fault-tolerance of the hidden states in HMM, a certain degree of interference on packet's IPT of a flow by censorship-circumvention systems can still be acceptable. Let $\mathbf{\Sigma}$ represent a collection of all the possible packet IPTs of a network flow. The complete range of IPT values in $\mathbf{\Sigma}$ is divided into $\mathcal{V}$ equal length of sub-ranges, r_v^{sub}, where the subscript v represents the vth sub-range. Each r_v^{sub} represents a hidden Markov state S_v of the HMM, e.g., $S_v \Leftrightarrow [\tau_{start}^v, \tau_{end}^v), \tau_{end}^v - \tau_{start}^v = l$. Let us name IPT, IPT_i^k, by an observation, Obs_i^k, where k is the kth chosen interval and i is the ith observation. Each observation, e.g., $Obs_i^k \in [\tau_{start}^v, \tau_{end}^v)$, is corresponding with a hidden Markov state S_v, where $Obs_i \in S_v$. The observation sequence $\mathbf{obs}_k$ of the received flow is a hidden Markov state sequence $\mathbf{state}_k$,

e.g., $\{Obs_{i-1}^k, Obs_i^k, Obs_{i+1}^k, Obs_{i+2}^k\} \rightarrow \{S_a, S_b, S_c, S_d\}$ as shown in Fig. 1.

We are interested in the state transition probability $P_{(i-1,i)}^k$, which denotes the hidden Markov states transition probability from S_a to S_b, i.e., from Obs_{i-1}^k to Obs_i^k. Given any particular sequence of state transition probability $\{P_{(1,2)}, \cdots, P_{(N-1,N)}\}$ and the transition probability range $P > 0$, the $P_{(i-1,i)}$ is approximately uniformly distributed in the range [*0,P*) when *N* is large enough, which is basically the same with the problem on remainders of the modulo operations proposed in [11]. In other words, given any flow with sufficient packets, for any randomly chosen offset $o > 0$, the state transition probability of any two packets are uniformly distributed in theory. Therefore, the expected value of $P_{(i-1,i)}$ is

$$E(P_{(i-1,i)}) = \frac{P}{2}, (i = 1, \cdots, N) \tag{1}$$

Assuming the interval T_k ($k = 0, \cdots, K$) has *N*+2>0 packets, e.g., $\{Pac_0^k, \cdots, Pac_{N+1}^k\}$, whose corresponding state sequence $\mathbf{state}_k$ has state transition probabilities, e.g., $\{P_{(1,2)}^k, \cdots, P_{(N-1,N)}^k\}$. Let us define the STPM in the interval T_k as

$$E_k = AvTran(\mathbf{state}_k) = \frac{1}{N} \cdot \sum_{i=1}^{N} P_{(i-1,i)}^k \tag{2}$$

The STPM will be considered as a valuable parameter to the design of this variable. However, the experimental statistical analysis found that STPM did not appear equally probabilistic at any given time interval, which will undoubtedly bring loopholes to the watermark embedding. The STPM shows a mixture Gaussians distribution in reality. To make the STPM be equal and be able to be used as the watermark carrier, we introduce a distribution mapping, which is detailed in Section III.B.1). Basically, the STPM is mapped to a discrete uniform distribution, resulting in the case that the STPM of any packets in any given time interval has equal opportunity to fall into any sub-range.

In order to use the STPM discrete distribution mentioned above in our scheme for embedding watermark, the watermarker chooses $K \times T$ length part of a flow, which consists of intervals, i.e., $\{T_0, \cdots, T_k, \cdots, T_K\}$, and divides each time interval $T_k \Leftrightarrow [t_{start}^k, t_{end}^k)$ into two parts: a *prediction* part, which is a sub-interval of $[t_{start}^k, t_{splitter}^k) \in [t_{start}^k, t_{end}^k)$, and a *watermark* part, which is a sub-interval of $[t_{splitter}^k, t_{end}^k) \in [t_{start}^k, t_{end}^k)$. The prediction part is used for predicting the hidden state sequence of the following watermark part and for indicating the watermark to be carried by the watermark part. The watermark part of STPM is used for carrying watermark, which is determined by the STPM of the prediction part, by adjusting its state sequence order.

B. Watermark encoding

To perform state-based watermarking, we select *K* time intervals of length *T*. In each interval, e.g., the *k*th interval T_k, the observations of prediction part form its state sequence $pflow_k$, and the watermark part's observations form its state sequence $eflow_k$. In this paper, we use the *Baum-Welch* algorithm [13] to learn the state transition matrix $\mathbf{\Phi}$ of the specified flow. That is because *Baum-Welch* algorithm helps find the maximum likelihood estimate of the parameters of $\mathbf{\Phi}$. The state transition matrix $\mathbf{\Phi}$ is the basis for the prediction and adjustment of the watermark state sequence.

For embedding a watermark, we firstly obtain $pflow_k$ of prediction part according to the *Viterbi* algorithm [14], which is widely used for predicting hidden states of given observations in HMM, and $\mathbf{\Phi}$. Secondly, we predict the most possible hidden Markov state sequence of the following watermark part with the *Forward* algorithm [13] based on the state transition probability matrix $\mathbf{\Phi}$.

In the second step, two main aspects need to be addressed: (i) in order to improve the invisibility of the embedded watermark, we introduce several different watermark patterns, which will be detailed in section III.B.1); (ii) in section III.B.2), we elaborate the adjustment methods of both a single state and a state sequence for embedding watermark in the watermark part.

1) Watermark Patterns: To improve the invisibility of the embedded watermark, this paper maps the mixture Gaussian distribution obeyed by STPM to a discrete uniform distribution, which is able to make equal probability of embedded watermark in any state sequences within any interval.

We divide the mixture of Gaussians of $pflow_k$ into *H* parts with the same statistics, and number each part in the direction of the *x-axis* growth. Each part is a discrete value in a discrete uniform distribution, which can be expressed as $P_h^m \Leftrightarrow [E_h^{MIN_P}, E_h^{MAX_P})$, where *h* represents the *h*th part and *m* is the *m*th watermark pattern. Then, we divide the mixture of Gaussians of $eflow_k$ into *H* parts with the same statistics, and number each part in the direction of the *x-axis* growth as well. Each part is a discrete value in a discrete uniform distribution, which can be expressed as $W_h^m \Leftrightarrow [E_h^{MIN_W}, E_h^{MAX_W})$. The watermark pattern refers to the mapping rule from the parts of $pflow_k$ discrete uniform distribution to that of $eflow_k$ discrete uniform distribution, which is denoted by the symbol p_m. The mapping rule can be a sequential mapping, i.e., $P_h^m \rightarrow W_h^m$, a reverse order mapping, i.e., $P_h^m \rightarrow W_{H-h}^m$, a misalignment mapping, i.e., $P_h^m \rightarrow W_{h+o}^m$, or a random mapping, i.e., $P_h^m \rightarrow W_{Rand()\%H}^m$, where $0 \leq o \leq H$ is an offset and $Rand()\%H$ is a random number. Each mapping rule is numbered from 0 to *M*, and which mapping rule will be used is pre-negotiated by the watermarker and the detector. The watermark, which is expressed as sig_h^m, refers to the number of discrete uniform distribution part to which the STPM of $eflow_k$ belongs under p_m.

The watermark selection process can be described as follows. First, we determine the implicit state sequence of $pflow_k$ with the Viterbi algorithm. Second, we calculate the STPM of $pflow_k$, i.e., $E_i^P \in [E_h^{MIN_P}, E_h^{MAX_P})$, and get the watermark sig_h^m according to the given mapping rules p_m, i.e., $sig_h^m = Mapping(E_i^P, p_h)$. Third, we find the possible sub-

range of values for the STPM of $eflow_k$ according to , i.e., $sig_h^m \rightarrow W_h^m$. Finally, $eflow_k$ is adjusted accordingly with the Forward algorithm to make its STPM fall into the target range, i.e., $[E_h^{MIN_W}, E_h^{MAX_W}) \Leftrightarrow W_h^m$.

2) Hidden Markov State Order Adjustment: In this section, the adjustment of state sequence of the watermark part, i.e., $eflow_k$, consists of two aspects: a) adjust a state to another and b) adjust the order of states in $eflow_k$.

a) State Adjustment: Assume that there is a possible state sequence of $eflow_k$ predicted by the Forward algorithm, it is highly likely that the state corresponding to the coming *I*th observation in the sequence is $S_\alpha \Leftrightarrow [\tau_{start}^\alpha, \tau_{end}^\alpha)$. However, it finds that the *I*th state is $S_\beta \Leftrightarrow [\tau_{start}^\beta, \tau_{end}^\beta)$, when the *I*th observations Obs_I^k comes. Obs_I^k needs to be delayed a period of time Δt to make its corresponding state be S_α. We select the $Obs_I^{k\prime} \in [\tau_{start}^\alpha, \tau_{end}^\alpha)$ corresponding to S_α with history value directly or the value with the highest probability from *History*. The delay increment Δt of the Obs_I^k can be calculated by $\Delta t = \sum_{i=1}^{I} Obs_i^{k\prime} - \sum_{i=1}^{I} Obs_i^k$.

Packets are forwarded multiple times in the Internet, and noises are easily introduced because of the forwarding strategy, network congestion, etc. In order to improve the robustness of the watermark, we keep the target observations at a minimum distance ε from the state boundaries: when Obs_I^k is adjusted as mentioned above, the target value $Obs_I^{k\prime}$ should keep a distance ε to the upper and lower boundary values of state S_α respectively, which can be expressed as

$$\tau_{start}^\alpha + \varepsilon < Obs_I^{k\prime} < \tau_{end}^\alpha - \varepsilon \tag{3}$$

When the maximum probability observation is less than this distance, the second largest probability observation is selected instead. This process repeats until this condition is met.

b) States Order Adjustment: In this section, we design the adjustment algorithm for the state sequence of $eflow_k$ to embed a watermark, which is shown in Algorithm 1.

In Algorithm 1, we obtain the state transition probability matrix $\mathbf{\Phi}$ by using *Baum-Welch* algorithm with the history flow dataset $History$. $History$ is used to predict the state sequence of watermark part. In addition to the origin flow **iflow**, we also need the watermark pattern p_m negotiated by the watermarker and the detector to determine the mapping rule, and the history flow dataset $History$. $History$ helps adjust the state sequence of $eflow_k$ in two aspects: (i) appropriate historical observations sequences are used directly as adjustment targets; (ii) the observation with large probability of each state in the historical flows is used as the state sequence adjustment target.

Algorithm 1 consists of two phases: prediction phase and adjustment phase. In the prediction phase, the Viterbi algorithm is first used to find the state sequence $\mathbf{state}_k$ of $pflow_k$, and the STPM of $pflow_k$ is calculated according to $\mathbf{state}_k$; the $Mapping(.)$ function infers the watermark sig_h to be embedded. Then, we use the Forward algorithm to predict all the possible state sequence $\mathbf{V}_k$ of $eflow_k$, and then the optimal state sequence $\mathbf{exp}_k$ is selected according to the principle of *front-dense and rear-sparse* [15]. The adjustment phase first searches $History$ for the sequence collection $\mathbf{\Theta}_k$ of the observations that match the target state sequence $\mathbf{exp}_k$, and if there is, select the best one $\mathbf{obs}_k$ and adjust accordingly; otherwise, it needs to be adjusted according to the predicted state sequence $\mathbf{exp}_k$ and the state large probability observations $\mathbf{obs}_k$ of the historical flow IPT statistics. Finally, Algorithm 1 returns the marked flow **oflow**.

Algorithm 1 Algorithm for IPT Sequence Adjustment

Input: Transition Matrix: $\mathbf{\Phi}$, Origin Flow: **iflow**, Signal Pattern: p_m, History Flows: $History$
Output: Watermarked flow: $oflow$

```
flow = CapFlow(iflow, T_k)
History ← Store(iflow)
while pflow_k ← flow.Next() ∈ [t^k_start, t^k_splitter) do
    state_k ← Viterbi(pflow_k)
    E^P_k ← AvTran(state_k)
    sig_n ← Mapping(E^P_k)
end while
V_k ← Forward(p_k, Φ)
exp_k ← Select(V_k, sig_n, p_m)
while flow.Next() ∈ [t^k_splitter, t^k_end) do
    if Θ_k ← History.Find(exp_k) then
        obs_k ← Select(Θ_k)
        oflow ← Adjust(obs_k)
    else
        obs_k ← History.Statistics_IPT_Seq(exp_k)
        oflow ← Adjust(obs_k)
    end if
end while
return oflow
```

C. Watermark detection

The detectors are placed at one or more points in the network where we might expect to detect marked flows. To detect the watermark, the detector captures *K* intervals, length of *T*, of a flow, i.e., $\{T_0', \cdots, T_k', \cdots, T_K'\}$, and analyzes the state sequence $\mathbf{state}_k'$ of each interval. The detector analyzes the STPM $\tilde{E_k^P}$ of $\mathbf{state}_k'$ of prediction part interval $[t_{start}^k, t_{splitter}^k) \in T_k'$. If $\tilde{E_k^P} \in [E_x^{MIN_P}, E_x^{MAX_P})$, we can determine the watermark sig_h embedded, which indicates the watermark part STPM in $[E_y^{MIN_W}, E_y^{MAX_W})$ according to the watermark mapping rule p_m. Then, according to the watermark sequence, which is $[t_{splitter}^k, t_{end}^k)$, we calculate its STPM $\tilde{E_k^W}$. If $\tilde{E_k^W} \in [E_y^{MIN_W}, E_y^{MAX_W})$, the embedded watermark sig_h in the interval T_k is detected.

To eliminate missed or false detection, the traditional probability threshold method is used to further improve the watermark detection effect. Suppose we have a total of X_f watermarks embedded in a flow *f*, and the detector correctly detect Y_f, then $Pr_f = Y_f/X_f$ is called the watermark detection rate. We set a threshold ρ for determining whether a flow has been embedded watermarks. If $Pr_f \geq \rho$, we consider the flow is watermarked; otherwise it is an unmarked flow.

IV. Performance Evaluation and Analysis

We simulate the HMSFW watermarking system by Python with a 2-hour-long traces on September 7th-8th and October 1st-5th, 2018; the trace is randomly chosen from the WIDE MAWI archive [16]. In the WIDE dataset, the Secure Shell (SSH) flows are selected out of the traces for experiment, since SSH are frequently used with interactive stepping stones, which behaves similar to the censorship-circumvention system. According to the statistics, the dataset used for experiments contains 404 SSH flows. In the experiment, we selecte 64 of the 404 SSH flows to embed 829 watermarks.

A. Watermark Detection Analysis

We consider four watermark detection results at both time interval level and flow level: *True Positive* (TP), the number of intervals or flows considered as embedded watermark and they were; *False Negative* (FN), the number of intervals or flows considered as embedded watermark but they were not; *False Positive* (FP), the number of intervals or flows considered as not embedded watermark but identified as embedded; *True Negative* (TN), the number of intervals or flows considered not embedded watermark and they were not. Based on the above four results, four evaluation indicators are formed as $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, and $F_1 = \frac{2\times Precision\times Recall}{Precision+Recall}$.

a) Time Interval Level: Fig. 2 shows the watermark detection results with the four evaluation indicators as mentioned above, in which x-axis represents the relative variation of STPM sub-range before marked and y-axis represents the values of these four indicators. From Fig. 2, we can see that the proposed HMSFW can achieve the *Accuracy* 98% , *Recall* 89%, *Precision* 98% and F_1 93%. These four indicators show that the HMSFW watermarking technique is effective in traffic tracking, but their curves in Fig. 2 show that *Accuracy, Recall* and F_1 first increase and then stabilize as the STPM sub-range expands, while $Precision$ slowly drops.

The reason is analyzed as (i) when the sub-range of STPM is small and the flow is disturbed, the STPM is easily changed even with very slight noise. Some STPMs changes are too large, which results in the un-matched prediction part and watermark part in the specified watermark pattern p_m; (ii) when the sub-range of STPMs is expanded, the HMSFW's robustness is improved, which is the reason that TP_{ti} and TN_{ti} increase, i.e., *Accuracy, Recall* and F_1 increase, *ti* denotes time interval level. The increase in TP_{ti} and the decrease in FP_{ti} do not cause *Precision* a significant increase, because the base of TP_{ti} is inherently much larger than FP_{ti}; (iii) with the STPM sub-range continues expansion, the STPMs of prediction part and watermark part tend to be stable, which means the STPM total range becomes narrow. This situation is easy to cause un-marked interval T mistaken as the marked one, i.e., the slowly increase of FP_{ti}, but the TP_{ti} tends to be constant. This is why *Precision* starts to decrease, and *Accuracy, Recall*, and F_1 tend to be stable.

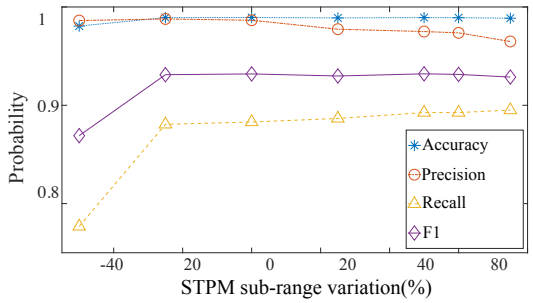

Fig. 2. Time Interval-based watermark detection results.

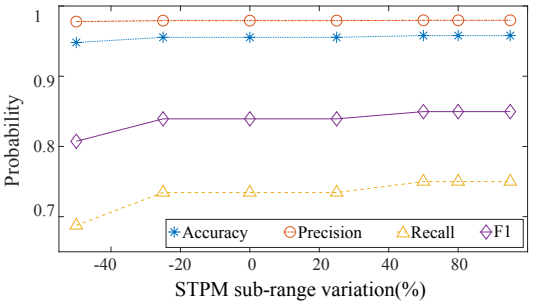

Fig. 3. Flow-based watermark detection results.

b) Flow Level: Fig. 3 shows the watermark detection results with the four indicators in the flow level under the condition that $\rho = 0.4$, in which x-axis represents the variation of STPM sub-range and y-axis shows the values of the four indicators. From Fig. 3, we can see that HMSFW can achieve *Accuracy* 96% , *Recall* 76%, *Precision* 98% and F_1 85%. These four indicators show that the proposed HMSFW watermarking technique is effective in traffic tracking, but their curves in Fig. 3 shows that the four indicators are first increase and then stabilize as the STPM sub-range expands.

The reason is analyzed as: (i) the same with time interval level analysis, when the sub-range of STPM is small, flows are easily disturbed by noises, and the STPM is easily changed. The prediction part and watermark part cannot match for each signal, which cause few correctly detected watermark and makes the $Pr_f < \rho$ common in a flow f; (ii) with the sub-range of STPMs expansion, the improvement of HMSFW's robustness makes the Y_f increase, i.e., the probability of $Pr_f \geq \rho$ increases, which is the reason for the increase in the four indicators; (iii) at the time interval level, the increase of FP_{ti} causes the stable of *Accuracy, Recall* and F_1, the slight decrease of *Precision*, which have weak influence on the indicator *Pr* of a flow. Therefore, the four indicators in flow level tend to be stable.

B. Invisibility

Recall invisibility refers to the possibility that the watermark cannot be analyzed by a third-party detector to detect the embedded watermark. The proposed HMSFW watermarking technology only adjusts IPTs of network traffic. Therefore,

the analysis of invisibility only needs to analyze the feature of IPTs, i.e., observation values. The analysis of network traffic mainly includes two points: (1) the distribution characteristics of the observations of each flow, and (2) the statistical characteristics of traffic observations embedded in watermarks.

The IPTs in each flow maintain the original state and original IPTs of the flow in the prediction part, while the watermark part has been adjusted. According to the Section III.B.2.b), the watermark part contains two adjustment methods, but both are adjusted based on the historical characteristics. Therefore, the watermark embedding trace cannot be detected from a single flow. Based on the analysis above, we mainly analyze the statistical characteristics of the flow observations of embedded signals. Figs. 4 and 5 are statistical distribution diagrams of flows observations before and after watermark embedded, respectively, in which x-axis represents the IPT and y-axis denotes the incidence probability of IPT. It can be seen that the

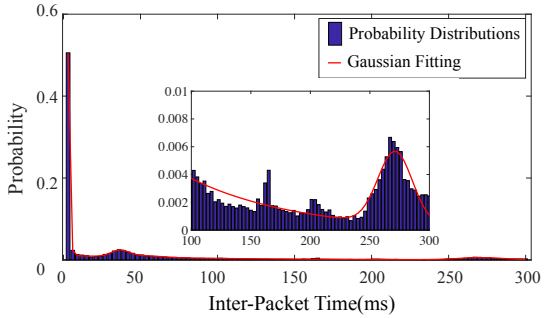

Fig. 4. Distribution of observations before watermarks are embedded.

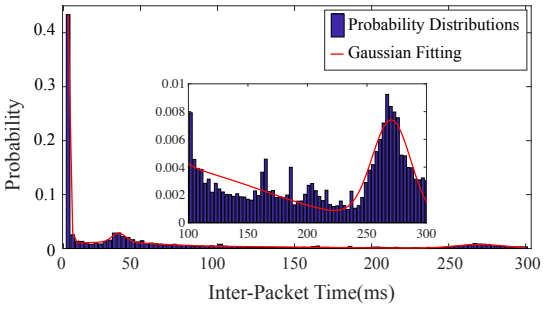

Fig. 5. Distribution of observations after watermarks are embedded.

distribution of flow IPTs is similar, with similar peaks and valleys. Both corresponding peaks and valleys have similar height and width. The small graphs in Figs. 4 and 5 are magnifications in the range of $[100, 300)$ on the x-axis. There is a difference in the IPTs distribution between the two small graphs. To quantify this difference in the observations distribution of the flow before and after embedding signals, we introduce cross entropy in this paper. Table 1 shows the entropy of IPTs distribution before and after embedding watermarks and the cross entropy of the two distributions. The difference between cross-entropy and pre-embedded is only 0.0221. According to the cross entropy calculation method, the cross entropy is small, which indicates that the watermarked IPT distribution probability is very similar to that before watermarked.

TABLE I
IPTs Statistical Distribution Entropy

Entropy Name	Entropy Values
Pre-watermark Distribution Entropy	3.9709
Suf-watermark Distribution Entropy	4.4357
Cross Entropy	3.993

V. Conclusion

This paper has proposed HMFSW using a hidden Markov model to embed watermarks, which can improve the watermark robustness and invisibility. HMFSW has used STPM as a new watermark carrier and maintains the equal probability of signal by using the STPM distribution mapping. The watermarks have been embedded according to the direct feature or statistical feature of the historical traffic information. The experiment results have shown that the proposed HMFSW has a high detection rate on watermarks with good invisibility.

References

[1] L. Wang, K.P. Dyer, et al, "Seeing through Network-Protocol Obfuscation,"*ACM Sigsac Conf.*, pp.57-69, 2015.

[2] A. Iacovazzi, S. Sarda, et al, "DROPWAT: An Invisible Network Flow Watermark for Data Exfiltration Traceback,"*IEEE Trans. on Infor. Forensics and Security*, vol.13, no.5, pp.1139-1154, 2018.

[3] N. Kiyavash, A. Houmansadr, et al, "Multi-flow attacks against network flow watermarking schemes,"*Conf. on Security Sym. USENIX Association*, pp.307-320, 2008.

[4] W. Jia, F.P. Tso, et al. "Blind Detection of Spread Spectrum Flow Watermarks,"*INFOCOM. IEEE*, pp.2195-2203, 2009.

[5] W. Yu, X.W. Fu, et al, "DSSS-Based Flow Marking Technique for Invisible Traceback,"*IEEE Sym. on Security and Privacy*, 2007.

[6] Z. Ling, X.W. F, et al, "Novel Packet Size-Based Convert Channel Attacks against Anonymizer,"*IEEE Trans. on Computes*, vol.62, no.12, pp.2411-2426, 2013.

[7] Y.J. Pyun, Y. Park, et al, "Interval-Based Flow Watermarking for Tracing Interactive Traffic,"*Comp. Networks*, vol. 56, no. 5, pp. 1646-1665, 2012.

[8] A. Houmansadr, N. Kiyavash, et al, "RAINBOW: A robust and invisible non-blind watermark for network flows,"*Network and Distributed System Security Sym.*, NDSS 2009.

[9] Z. Lin, N. Hopper. "New Attacks on Timing-Based Network Flow Watermarks,"*Security'12 Proc. of the 21st USENIX Conf. on Security Sym.*, pp. 381-396, 2012.

[10] X.P. Luo, et al. "Exposing Invisible Timing-Based Traffic Watermarks with BACKLIT,"*Proc. of the 27th Ann. Comp. Security App. Conf. On*, pp. 197-206, 2011.

[11] X. Wang X, S. Chen, et al, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems,"*IEEE Sym. on Security and Privacy, SP '07*, pp.116-130, 2007.

[12] A. Houmansadr, N. Borisov. "SWIRL: A Scalable Watermark to Detect Correlated Network Flows," *Network and Distributed System Security Sym.*, NDSS 2011.

[13] L. Baum. "An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process,"*Inequalities*, 1972, 3:1-8.

[14] A.J. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,"*IEEE Trans. on Infor. Theory*, 1967.April, 13 (2): 260-269.

[15] K. Cho, K. Mitsuya, et al, "Traffic Data Repository at the WIDE Project,"*Proc. of the Annual Conf. on USENIX Ann. Technical Conf. (ATEC '00)*, pp.51–51, 2000.

[16] J.W. Kang, et al, "Front-Dense and Rear-Sparse Double-Row Pile Foundation Pit Supporting and Protecting Structure," 2013.