2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems

Meek-based Tor Traffic Identification with Hidden Markov Model

Zhongjiang Yao[1,2], Jingguo Ge[1,2], Yulei Wu[3], Xiaodan Zhang*[,1], Qiang Li[1,2], Lei Zhang[4], Zhuang Zou[1,2]
[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
[4]School of Telecommunications Engineering, Xidian University, Xian, 710126, China
Email: {yaozhongjiang, gejingguo, zhangxiaodan}@iie.ac.cn, y.l.wu@exeter.ac.uk
*Correspondence author

Abstract—**Tor is one of the major technologies of censorship circumvention systems. To protect the privacy of users and the information of first hop to access Tor networks, Tor Browser introduces an obfuscation technology called Meek. Tor traffic is obfuscated by Meek to behave as ordinary cloud service traffic. In order to advance the capability of network monitoring systems, this paper proposes a Mixture of Gaussians based Hidden Markov Model (MGHMM), a new model for identifying Meek-based Tor traffic. The proposed MGHMM has two components: 1) Mixture of Gaussians (MOG) is used to characterize the Inter-Packet Time (IPT) distribution and the Packet Size (PS) density distribution; 2) HMM is used to compute the probability of a traffic observation sequence and identify Meek-based Tor traffic by using two-dimensional observations composed by IPT and PS. The effectiveness of the proposed model is evaluated with real-world traffic. Extensive experiments show that the proposed MGHMM is able to identify Meek-based Tor traffic effectively.**

I. Introduction

Tor network [1] is one of the most widely used anonymous communication systems, and also one of the main techniques of network censorship evasion systems. Tor network is mainly used to protect users' privacy. However, criminals use Tor's censorship to circumvent criminal activities. For example, the illegal website, Silk Road, uses Tor networks to engage in selling poison and illegal firearms [2]; the terrorist organization uses Tor networks to plan terrorist attacks [3]. To keep the network healthy, many researchers have come up with a variety of techniques for identifying simple Tor traffic (the Tor traffic that has not been obfuscated). In recent years, Tor networks introduced Pluggable Transport (PT), which can make Tor traffic hidden in the background traffic, to circumvent network censorship. Meek is a new PT, based on which the traffic is deemed to be the safest [4]. The introduction of Meek [5] has led to more rampant illegal activities and significant regulatory challenges to the network.

Meek is based on domain fronting technology, which essentially uses different domain names at different communication layers [5]. It consists of three steps: 1) the domain name of common cloud service providers (e.g., Google, Amazon, Azure, etc.) is placed in the Server Name Indication (SNI) of the Transport Layer Security (TLS) handshake protocol of client messages and is disclosed to the public; 2) the domain name of the server running Tor Bridge which is the access point to the Tor network, is placed in the Host header field, and the cloud service obtains the domain name after decrypting the TLS and forwards the packet to the Bridge; and 3) the real domain name of the visited website is encapsulated in the HTTP body in the Tor Cell data format [6]. Observers can only detect the traffic of common cloud services and the other ordinary traffic from the traffic datasets that contain Meek-based Tor traffic. If observers determine Meek-based Tor traffic based only on the SNI of the TLS handshake protocol's extension field, it is easily confused with the ordinary cloud service traffic. The relay support of cloud nodes to abnormal traffic may result in lower trustworthiness of server nodes [28].

Many existing studies [7-11] have been reported for the identification of simple Tor traffic. For example, He et al. [7] proposed a method for the extraction of features, e.g., packet size, by using the multiple relationships between packet size and Cell size. However, these approaches have been incapable of identifying the Tor traffic that is based on obfuscation techniques such as Meek, because obfuscation techniques help hide Tor traffic into ordinary traffic by random, mimicry or tunnel [4]. Researchers therefore proposed methods to identify Meek-based Tor traffic with TLS cipher suites [12] and polling message interval sequences [13]. However, the latest Tor Browser with versions 7.0.10 has repaired the problem caused by cipher suites difference. Meek-based Tor traffic is now able to carry the same 15 cipher suites as the ordinary TLS Client does. The polling interval may also be adjusted, which means that it cannot be used as a universal recognition feature.

To this end, this paper presents a Mixture of Gaussians based Hidden Markov Model (short for MGHMM), a new and generic Markov model for Meek-based Tor traffic identification. MGHMM uses state features formed by two-dimensional observation vectors, obtained from Inter-Packet Time (IPT) and Packet Size (PS) which are the most common and basic characteristics of a flow. It avoids Tor Browser version restrictions. Experiment results show that the proposed MGHMM can effectively identify the Meek-based Tor traffic.

The main contributes of this paper can be summarized as follows:

- This paper proposes a novel traffic identification model named MGHMM. For the first time, it merges Mixture of Gaussians (MOG) with Hidden Markov Model

978-1-5386-6614-2/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2018.00075

(HMM) for traffic identification. The model uses MOG to depict two-dimensional observation probability density of each state, and then constructs the HMM model by considering the state changes of traffic during the communication. MGHMM gets rid of PT version restrictions, and thus it is a more generic traffic identification model.

- MOG is introduced for the first time to describe the probability density distribution of two-dimensional observation vectors. In the MOG, IPT and PS are described jointly with covariance matrix, which means IPT and PS are no longer treated as separate observations. The closer the MOG probability density distribution is depicted to real density distribution, the more reliable the observation is as a fingerprint feature. From the experiment results, an excellent Tor traffic identification can be achieved when MOG fits the states of HMM.
- The paper uses the real Meek-based Tor traffic collected from Internet to verify and evaluate the effectiveness and accuracy of the proposed MGHMM model. Experiment results show that MGHMM can obtain an identification rate of 99.4%.

The rest of the paper is organized as follows. Section II describes the motivation and related work. Section III presents the proposed MGHMM model, including the model inputs and the necessary derivation processes. Section IV carries out experiment design and performance analysis. Section V gives the conclusion of this paper.

II. Related Work

A. Motivation

Meek encapsulates Tor traffic in HTTPS and confuses it with ordinary cloud service HTTPS traffic. Meek-based Tor traffic can therefore trick the forced filtering in the network. On the other hand, Meek-based Tor traffic is TLS-encrypted, and Deep Packet Inspection (DPI) would not be possible to analyze the TLS-encrypted part [4]. For these reasons, Meek plays an increasingly important role in the development of Tor anonymous communication. However, there are few researches on the identification of Meek-based Tor traffic in the current literature. The most widely used methods relied primarily on the difference of TLS cipher suites between Meek-based Tor traffic and ordinary traffic [14], but this has been fixed in the new Tor Browser releases [15]. In addition, some studies [4, 14, 16] relied on dozens of features, e.g., the packet size distribution and the entropy, bringing heavy workload on identification but poor recognition accuracy. In order to improve the accuracy of identifying Meek-based Tor traffic with fewer features, this paper presents a novel traffic identification model based on HMM.

B. Related studies

IPT and PS have been widely used for traffic classification and identification. An early study on application flow identification classifier which relied on the first 4~10 packets size of an observation sequence, was proposed in [18]. The authors proposed two separate classifiers with IPT and PS in [18] and [19], respectively. An extension of [18] was presented in [20], in which they proposed a complex state structure for each packet and a method trying to join IPT and PS through vector quantization with a codebook. A packet-level traffic classifier trying to join IPT with PS was proposed in [21], but they are still computed separately within the same function.

HMM has been proposed for traffic classification and identification. Wright, Monrose and Masson [18] proposed two HMM analysis models with IPT and PS separately. A traffic classifier based on multiple Packet-Level Hidden Markov Models (PL-HMMs) using Gamma function to characterize IPT and PS probability density was proposed in [21]. Although PL-HMMs can be used to identify different application traffic and obtain high traffic recognition rate, it would be computationally expensive as multiple PL-HMMs are required. Although [22] constructed a single HMM model based on Tor network virtual circuit construction process and Tor log information, it can only effectively identify Tor flows which contain Tor circuit initial messages and have not been obfuscated by obfuscation technologies. Similarly, Korczynski and Duda [23] used the random fingerprint of SSL/TLS-based application traffic in different applications to construct HMM to identify different application traffic, but it needs a large amount of observation and analysis work in the early stage. The HMM was adopted to predict future traffic by Chen, Wen and Geng in [24], which can avoid direct measurement of the traffic volume, but instead we estimate and predict the hidden traffic volume based on those simple flow statistics. The most recent studies on traffic identification based on HMM were conducted is [25] and [26]. Weighted ENsemble Classifier (WENC) based on HMM, which studies the HTTPS handshake process and the following data transmission period, was proposed for classification of HTTPS encrypted traffic in [25], and the traffic classification accuracy of WENC is 90%. The authors in [26] proposed a Long-Term Evolution (LTE) signal detection scheme based on Hidden semi-Markov model to capture the spatial-temporal characteristics of normal wakeup packet generation behavior, but the experiment was based on simulation rather than real network environment.

In this paper, we propose MGHMM, an HMM based on MOG to describe the traffic fingerprint by using two-dimensional probability density distribution of the observation formed by IPT and PS. MOG introduces a mixing factor that more accurately characterizes the density distribution of IPT and PS, including the relation between IPT and PS. An observation sequence then maps a hidden state transition sequence, whose occurrence probability can be used as a discriminant index. MGHMM uses a single HMM and needs fewer identification features and computational overhead, which can be used as one of the technologies of smart network management [29].

III. The Proposed Model

For the sake of the clarity of illustration, the main symbols used in the description of the proposed MGHMM are summarized in Table I, where notation variables are shown in normal italic symbols, column vectors are denoted with lowercase bold letters, and parameter matrix is represented by uppercase bold letters. The probability density function is denoted with $f(\cdot)$. $\widehat{X}$ is the final likelihood of MGHMM parameters (e.g., the prior probability or the parameters of MOG in MGHMM).

TABLE I. THE MAIN SYMBOLS USED IN THE PROPOSED MODEL

Symbol	Description	Symbol	Description
$\mathbf{m}$	mean vector of MOG	$\mathbf{A}$	state transition matrix
$\mathbf{\Omega}$	prior probability matrix	a_{ij}	transition probability from s_i to s_j
R	observation sequence	$\mathbf{w}$	weight vector of MOG
$\mathbf{r}_i$	the i th observation variable in R	ω_{ij}	weight of a Gaussian in MOG
$\boldsymbol{\mu}_{ij}$	mean vector of the j th Gaussian in the i th state	$\mathbf{r}_i'$	the i th observation vector in sub-sequence of R
ps_i	packet size in $\mathbf{r}_i$	ipt_i	inter-packet time in $\mathbf{r}_i$
$\boldsymbol{\sigma}_{\mathrm{ij}}{}^2$	variance matrix of the j th Gaussian in the i th state	$\mathbf{\Phi}$	variance matrix of MOG, whose element is a covariance matrix of sub-Gaussian
λ	parameters set of MGHMM	s_i	the i th state in S
η_{lib}	probability of the l th observation generated by the b th Gaussian of state s_i	γ_{ilb}	the l th observation probability generated by the b th Gaussian in s_i
α_i^l	forward probability of the i th state in the l th position of the observation sequence	β_i^l	backward probability of the i th state in the l th position of the observation sequence
S	observation state set	N	the number of states
L	the number of observations	B	the number of Gaussians
ξ_{lij}	probability of transition from s_i to s_j at the l th position	ς	value to determine a traffic if it is a Meek-based Tor traffic

A. The proposed MGHMM

In order to get rid of the limitation of the Tor Browser version and identify Meek-based Tor traffic which is obfuscated in ordinary HTTPS traffic, this paper presents the MGHMM model. MGHMM takes the two-dimensional observation vectors composed of IPT and PS as inputs. The observations are clustered as hidden states. Each hidden state cluster is characterized by MOG, and the hidden state transition rule is characterized by the Markov chain. The occurrence probability of the observation sequence then outputs.

Specifically, MGHMM is composed of a series of hidden states and observation sequence. The hidden state variable s_n is contained in the state set $S=\{s_0,\cdots,s_n,\cdots,s_N\}$, where N is the sum of the states. The observable variable $\mathbf{r}_l$ is contained in the observation sequence $R=\{\mathbf{r}_0,\cdots,\mathbf{r}_l,\cdots,\mathbf{r}_L\}$, and each observation in R is $\mathbf{r}_l=[ipt_l,ps_l]^T$. ipt_l is the IPT between the l th packet and $(l+1)$ th packet, and ps_l is the PS of the l th packet. The two variables of ipt_l and ps_l can be correlated with a mixing coefficient. As MGHMM is based on MOG and HMM, of which are characterized with their own parameters. Therefore, the MGHMM can be characterized by a set of parameters as:

$$\lambda=\{\mathbf{\Omega},\mathbf{A},\mathbf{w},\mathbf{m},\mathbf{\Phi}\}$$

where $\mathbf{\Omega}$ is a prior probability matrix for hidden states. $\mathbf{A}$ is the state transition matrix, whose element a_{ij} ($1\le i\le N$, $1\le j\le N$) is the probability of transition from state s_i to state s_j . $\mathbf{w}$ is the vector of mixture coefficient, also called weight, whose element ω_{ib} is the b th Gaussian distribution in the state s_i . $\mathbf{m}$ is the mean vector, whose element is the mean $\boldsymbol{\mu}_{\mathrm{ij}}$ of the b th Gaussian. $\mathbf{\Phi}$ is the full covariance matrix of Gaussian distribution, whose element is $\boldsymbol{\sigma}_{ib}$.

Each observation sub-sequence corresponds to a hidden state sequence of HMM. Each state is characterized by a bank of B Gaussian as follows:

$$f_i(R)=\sum_{b=1}^{B}\omega_{ib}N(R;\mathbf{m},\mathbf{\Phi})$$

where $N(R;\mathbf{m},\mathbf{\Phi})$ is the b th Gaussian distribution of a hidden state. In what follows, the MOG and HMM in the construction of the proposed MGHMM are elaborated.

- Mixture of Gaussians (MOG)

To characterize each hidden state, we introduce MOG, which combined several different Gaussians with different weights. Each Gaussian computes a two-dimensional variable with a covariance matrix, which joints IPT and PS. The two-dimensional probability density of each state estimated by MOG is the basis of the HMM's assessment of state transition probabilities.

The Expectation-Maximization algorithm [27] is a well-founded statistical algorithm to get around this problem by an iterative process. In order to obtain the optimal parameter value of MOG, this paper adopts the Expectation-Maximization algorithm, where the expectation step (**E-step**) computes the expected probability of an observation coming from a specific Gaussian component with the current estimate for the parameters, and the maximization step (**M-step**) computes the parameters maximizing the expected log-likelihood found on the **E-step**.

E-step: we first choose a state s_i from the clustered observations. Then, we estimate the probability of the l th observation in the training observation sequence with prior probability, which is generated by the b th Gaussian component as follows:

$$\eta_{lib}=\frac{\omega_{ib}N(R;\mathbf{m},\mathbf{\Phi})}{\sum_{b=1}^{B}\omega_{ib}N(R;\mathbf{m},\mathbf{\Phi})}$$

M-step: we re-determine the parameters of each Gaussian component to maximize the Gaussian parameters to obtain the optimal values. The mixed Gaussians parameters mainly contain mean vector **m** and covariance matrix **Φ** and its weight in state s_i .

Repeat the **E-step** and **M-step** to improve the parameters of MOG until the parameters go convergence.

- Hidden Markov Model (HMM)

After the estimation of the parameters of MOG, this paper computes the HMM parameters of MGHMM based on the Expectation-Maximization algorithm, where **E-step** computes the transition probability between any two HMM states, and **M-step** re-computes the MGHMM parameters.

E-step: After determining each hidden state of HMM, we calculate the probability of occurrence of each state with Forward algorithm and Backward algorithm separately. Then, we calculate the state transition probabilities with the Forward-Backward algorithm[30]. The transition probability from state s_i to state s_j is:

$$\xi_{lij} = \frac{\alpha_j^l \beta_j^l}{\sum_{i=1}^{N} \alpha_i^l \beta_i^l}$$

M-step: After obtaining the transition probabilities between states, the component parameters and prior probabilities for each state are re-evaluated, which are also the parameters of MGHMM.

The probability of the observation generated by the b th Gaussian of state s_i can be expressed as:

$$\gamma_{ilb} = \xi_{lij} \cdot \eta_{lib} = \frac{\alpha_j^l \beta_j^l}{\sum_{i=1}^{N} \alpha_i^l \beta_i^l} \cdot \frac{\omega_{ib} N(R;\mathbf{m},\mathbf{\Phi})}{\sum_{b=1}^{B} \omega_{ib} N(R;\mathbf{m},\mathbf{\Phi})}$$

The prior probability is:

$$\widehat{\varepsilon_i} = \frac{\sum_{l=1}^{L} \gamma_{ilb}}{L}$$

The weight of the b th Gaussian distribution is:

$$\widehat{\omega_{ilb}} = \frac{\sum_{l=1}^{L} \gamma_{ilb}}{\sum_{l=1}^{L} \sum_{b=1}^{B} \gamma_{ilb}}$$

The mean of the b th Gaussian distribution can be given by:

$$\widehat{\boldsymbol{\mu}_{ilb}} = \frac{\sum_{l=1}^{L} \gamma_{ilb} \cdot \mathbf{r}_l^n}{\sum_{l=1}^{L} \gamma_{ilb}}$$

The covariance of the b th Gaussian distribution is:

$$\widehat{\boldsymbol{\sigma}_{ilb}^{\;2}} = \frac{\sum_{l=1}^{L} \gamma_{ilb} (\mathbf{r}_l - \widehat{\boldsymbol{\mu}_{ib}})(\mathbf{r}_l - \widehat{\boldsymbol{\mu}_{ib}})^T}{\sum_{l=1}^{L} \gamma_{ilb}}$$

Based on the derivation above, the state-based fingerprint of Meek-based Tor traffic can be achieved. We can identify a Meek-based Tor flow by calculating the occurrence probability of an observation sequence as follows:

$$\zeta = \wp(R|\lambda) = \sum_{l=1}^{L} \sum_{i=1}^{N} \alpha_i^l \beta_i^l$$

where $\wp(R \mid \lambda)$ donates the likelihood of the observation R under the condition λ , the training results of which allow us to choose a threshold θ , and then a flow from the testing traffic dataset will be determined as Meek-based Tor traffic when ζ is greater than θ .

B. *Training*

In order to obtain better classification results, we adopt the well-known K-means algorithm at the initial state classification and re-order the observations of each state as $\{\mathbf{r}_0', \cdots, \mathbf{r}_l'\}$. In order to obtain the best fitting effect, we choose different state clustering and Gaussian mixture components. The number of state is selected in the real range [3, 10]. For ease of understanding, the selected number of states in all the figures in this section is 3. For each observation state, we select the appropriate number of mixed Gaussians in the range of [7, 15].

Based on cluster results obtained by the K-means algorithm, MGHMM is trained to obtain the maximum likelihood of MOG parameters, i.e., weight, mean vector and covariance matrix of each cluster. The MOG fitting effect of the two-dimensional vector observations composed of IPT and PS, are evaluated in Fig. 1.

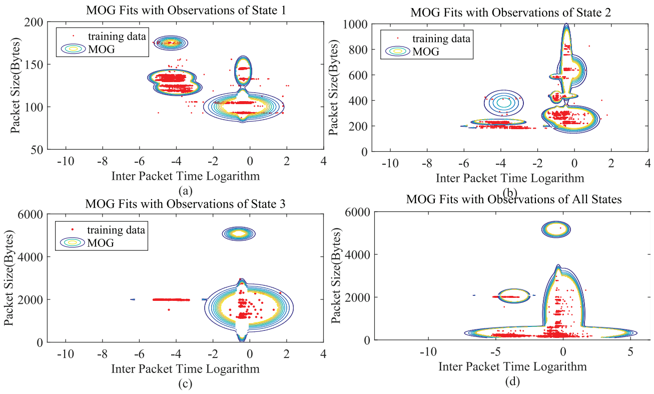

Fig. 1. The fitting of two-dimensional vectors with MOG: (a) the fitting result of state cluster 1, (b) the fitting result of state cluster 2, (c) the fitting result of state cluster 3, and (d) the fitting result of all training data sets.

Fig. 1, whose x-axis denotes the logarithm of inter-packet time and y-axis denotes packet size, fits the overall variable density distribution formed by the two-dimensional vectors of IPT and PS. The results show that a Gaussian mixture can fit well for each type of observation. The place where the observation points are concentrated becomes the Gaussian center of the MOG, and each MOG can cover most of the observations. Those observations that cannot be covered need larger training data sets. That is to say the training set used in this paper is not large enough, which is the reason why there are observations that are not covered by the MOG in the Fig. 1(a).

After the above training, the proposed MGHMM becomes the maximum likelihood of Meek-based Tor traffic with different states and Gaussian components. The Meek-based Tor traffic identification is obtained based on the observation ranges characterized by those MGHMM parameters.

IV. Experiment Results and Analysis

To increase the credibility, we leverage real network traffic to evaluate the effectiveness and accuracy of the proposed MGHMM on the identification of Meek-based Tor traffic. The traffic dataset is 15GB in size, including web pages, emails, images, videos, audios, and files (e.g., executables, source code, and PDF files), generated by browsers such as Chrome, Firefox, Tor Browser, Arora, etc. All the traffic is captured with a browser driver script on the gateway of a local area network. We only consider uploading HTTPS traffic extracted based on TCP SYN and FIN flags. The training traffic dataset and test traffic dataset are given in TABLE II.

TABLE II. THE RESULTS OF TRAFFIC IDENTIFICATION

Measures	Training		Testing	
	Flows	Packets	Flows	Packets
Meek-based Tor	525	6662046	215	576541
Ordinary	37293	8030003	6524	4057828

A. Evaluation metrics

All traffic identification can be classified into the following four results: True Positive (TP) represents the identification of the true Meek-based Tor traffic as Meek-based Tor traffic, and False Positive (FP) is the variable representing the false identification of the ordinary traffic as Meek-based Tor traffic. False Negative (FN) is the one falsely identifying Meek-based Tor traffic as ordinary traffic, and True Negative (TN) is the one correctly identifying the true ordinary traffic as ordinary traffic. The evaluation indicators used in this paper to assess the performance of the proposed MGHMM are based on these four metrics and can be expressed as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN},\ Precision = \frac{TP}{TP+FP}$$
$$Recall = \frac{TP}{TP+FN},\ F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

The *Accuracy* is identified as the proportion of correctly identified traffic, namely, the proportion of identified true Meek-based Tor traffic and ordinary traffic with all tested traffic. The *Precision* is identified as the proportion of correctly identified Meek-based Tor traffic, namely, the proportion of identified true Meek-based Tor traffic with all the identified Meek-based Tor traffic. *Recall* is the proportion of correctly identified target traffic with all the target traffic, namely, the proportion of the identified true Meek-based Tor traffic with all the true Meek-based Tor traffic. F_1 is an important indicator to evaluate the effectiveness of the identification method.

B. Experiment results and Analysis

Based on the training results in Section III.B, we make the corresponding experiment of Meek-based Tor traffic identification. When the test traffic observations are clustered into different states, and each state is characterized into different Gaussians, their evaluation metrics are different. The evaluation metrics are compared when all the test traffic is divided into 3 states, 6 states and 10 states, and each state consists of 7-15 Gaussian components. The comparison results are shown in Fig. 2.

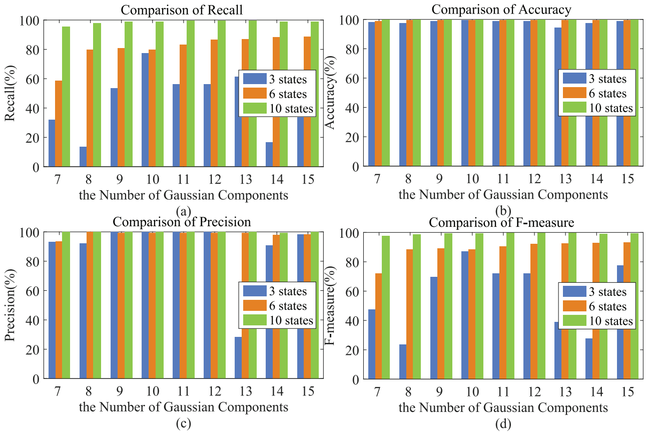

Fig. 2. The comparison of traffic identification evaluation metrics: (a) the comparison of recall, (b) the comparison of accuracy, (c) the comparison of precision, and (d) the comparison of F_1

In Fig. 2, the x-axis is the number of Gaussian components of each state, and the y-axis is the traffic identification accuracy. The results are under the conditions that the traffic observations are divided into 3 states, 6 states and 10 states, and each state consists of 7 to 15 Gaussian components. From Fig. 2, we can see that the more states are divided, the higher identification accuracy is, if each state is characterized with the same numbers of Gaussians. Similarly, the larger the number of Gaussian components is, the higher identification accuracy is when the traffic has the same states volume. It is worth noting that the evaluation metrics tend to be less steady when fewer states are divided. The main reason is that some observations far away from a state cluster center are forcedly clustered into the state when there are only a few clusters. In addition, it is not a good practice to use many Gaussian components. For example, the accuracy with 14 and 15 Gaussians is not better than that with only 11 Gaussians; in addition, that with 14 and 15 Gaussians takes longer calculation time.

In the identification process, too few state divisions and Gaussian components can cause the accuracy to decrease. However, too many state divisions and Gaussian components mixed will only bring bigger computational overhead when the accuracy reaches a certain value, and the accuracy will not be improved further. With appropriate state divisions and Gaussian components, not only can the computational overhead be reduced, but also an optimal identification rate can be obtained.

TABLE III. THE RESULTS OF TRAFFIC IDENTIFICATION

Measures	Value
Accuracy	99.98%
Precision	100%
Recall	99.43%
F_1	99.72%

Table III gives the identification results when the test traffic is divided into 10 states, and each state has 11 Gaussian components. As can be seen from this table, the proposed MGHMM can get 99.98% on accuracy, 100% on precision, 99.43% on recall, and 99.72% on F_1 . The reason why such good results can be obtained is that, on one hand, the adoption of K-means algorithm provides a good initial classification result; on the other hand, MOG is used to characterize the observed density distribution of each state which can cover almost each peak and trough. The experiment results indicate that MGHMM can effectively identify the Meek-based Tor traffic.

V. Conclusions

In this paper, we have proposed a new model called MGHMM which uses the two-dimensional features of IPT and PS to identify Meek-based Tor traffic. According to the given training method, the [IPT, PS] two-dimensional surface have been shown. The hidden states of observations can be characterized accurately with MOG, and the Meek-based Tor traffic can be identified based on the hidden state transition of an observation sequence. Experimental results have shown that the proposed MGHMM can effectively identify Meek-based Tor traffic and has a high identification performance. Since MGHMM performs the identification by using common features only, it can be used as a potential analysis for any types of traffic.

Acknowledgment

This work is supported by the National Key R&D Program (Grant No.: 2017YFB0801801), the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No.: 2017ZX03001019-003), and the Science and Technology Service Network Initiative (STS) Project of Chinese Academy of Science (Grant No.: Y7X0071105).

References

[1] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Commun. of the ACM*, vol.42, no.2, pp 39-41, 1999.

[2] https://en.wikipedia.org/wiki/Silk_Road_(marketplace).

[3] https://archive.org/details/ISIL-tor-guide.

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

[5] D. Fifield, C. Lan, and R. Hynes, et al, "Blocking-resistant communication through domain fronting," *Proc. on Privacy Enhancing Technol.*, vol. 2015, no.2, pp. 46-64, 2015.

[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," *J. of the Franklin Institute*, vol. 239, no.2, pp.135-139, 2004.

[7] G.F. He, M. Yang, J.Z. Luo, et al, "Online identification of Tor anonymous communication traffic,". *J. of Software*, vol. 24, no.3, pp. 540-556, 2013.

[8] X. Bai, Y. Zhang, and X. Niu, "Traffic Identification of Tor and Web-Mix," *IEEE Eighth Int. Conf. on Intelligent Systems Design and Appl.*, pp.548-551, 2008.

[9] [11] F. Mercaldo and F. Martinelli, "Tor traffic analysis and identification," *Aeit Int. Conf.*, pp.1-6, 2017.

[10] Z. Rao, W. Niu, and X.S Zhang, et al, "Tor anonymous traffic identification based on gravitational clustering," *Springer Peer-to-Peer Networking and Appl.*, vol. 3, pp. 1-10, 2017.

[11] A.H. Lashkari, G.D. Gil, and M.S.I. Mamun, et al, "Characterization of Tor Traffic using Time based Features," *Int. Conf. on Inf. Systems Security and Privacy*, pp. 253-262, 2017.

[12] M. Husák, Čermák, Milan, and T. Jirsík, et al, "HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting," *Eurasip J. on Inf. Security*, vol. 2016, no.1, pp. 30, 2016.

[13] T. Janczak, "Polling in wireless networks," *U.S. US 7706399 B2*, 27 Apil 2010.

[14] Y.Z. He, X. Li, and M.L. Chen,et al, "Identification of Tor Anonymous Communication with Cloud Traffic Obfuscation," *Advanced Engineering Sciences*, vol. 49, no.2, pp. 121-132, 2017.

[15] Tor: https://www.torproject.org/.

[16] Q. Tan, J. Shi, and B. Fang, et al, "Towards measuring unobservability in anonymous communication systems," *Journal of Computer Research and Development*, vol. 52, no. 10, pp. 2373-2381, 2015.

[17] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," *Proc. of the 2006 ACM CoNEXT Conf.* Article No. 6, 2006.

[18] C. Wright, F. Monrose, and G.M. Masson, "HMM profiles for network traffic classification," *Proc. of the 2004 ACM workshop on Visual. and data mining for comput. security*., pp. 9-15, 2004.

[19] T. Auld, A.W. Moore, and S.F. Gull, "Bayesian neural networks for internet traffic classification,". *IEEE Trans. on neural networks*, vol. 18, no.1, pp. 223-239, 2007.

[20] C.V. Wright, F. Monrose, and G.M. Masson, "Towards better protocol identification using profile HMMs," *JHU Tech. Rep.*, vol. JHU-SPAR051201, 2005.

[21] A. Dainotti, W.D. Donato, and A. Pescape, et al, "Classification of Network Traffic via Packet-Level Hidden Markov Models," *IEEE Global Telecommun. Conf. (GLOBECOM)*, pp. 1-5, 2008.

[22] S. Zhioua, "Tor traffic analysis using Hidden Markov Models," *Security & Commun. Networks*, vol. 6, no.9, pp. 1075-1086, 2013.

[23] M. Korczynski and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," *IEEE INFOCOM, 2014 Proc.*, pp. 781-789, 2013.

[24] Z. Chen, J. Wen, and Y. Geng, "Predicting future traffic using Hidden Markov Models," *IEEE Int. Conf. on Network Protocols (ICNP)*, pp. 1-6, 2016.

[25] W. Pan, G. Cheng, and Y. Tang, "WENC: HTTPS Encrypted Traffic Classification Using Weighted Ensemble Learning and Markov Chain," *IEEE TrustCom/BigDataSE/ICESS*, pp. 50-57, 2017.

[26] J.H Bang, Y.J Cho, and K. Kang, "Anomaly detection of network-initiated LTE signaling traffic in wireless sensor and actuator networks based on a Hidden semi-Markov Model," *Comput. & Security*, vol. 65, pp.108-120, 2017.

[27] A.P.Dempster, N.M.Laird, and D.B.Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. of the royal statistical society Series B*, vol. 39, no. 1, pp.1-38, 1977.

[28] C. Huang, G. Min, Y. Wu, et al. "Time Series Anomaly Detection for Trustworthy Services in Cloud Computing Systems," *IEEE Transactions on Big Data*, vol. 99, pp. 1-1, 2017.

[29] Y. Wu, F. Hu, G. Min, and A. Zomaya (eds.), "Big Data and Computational Intelligence in Networking," *Taylor & Francis/CRC*, ISBN: 9781498784863, 2017.

[30] J.L. Fan, "Forward-Backward Algorithm," *Springer International*, vol.627, pp. 97-116, 2001.