IEICE TRANS. INF. & SYST., VOL.E95–D, NO.12 DECEMBER 2012

PAPER Special Section on Parallel and Distributed Computing and Networking

Towards Cost-Effective P2P Traffic Classification in Cloud Environment

Tao BAN[†a)], **Shanqing GUO**[††], ***Nonmembers***, **Masashi ETO**[†], **Daisuke INOUE**[†], ***and*** **Koji NAKAO**[†], ***Members***

SUMMARY Characterization of peer-to-peer (P2P) traffic is an essential step to develop workload models towards capacity planning and cyber-threat countermeasure over P2P networks. In this paper, we present a classification scheme for characterizing P2P file-sharing hosts based on transport layer statistical features. The proposed scheme is accessed on a virtualized environment that simulates a P2P-friendly cloud system. The system shows high accuracy in differentiating P2P file-sharing hosts from ordinary hosts. Its tunability regarding monitoring cost, system response time, and prediction accuracy is demonstrated by a series of experiments. Further study on feature selection is pursued to identify the most essential discriminators that contribute most to the classification. Experimental results show that an equally accurate system could be obtained using only 3 out of the 18 defined discriminators, which further reduces the monitoring cost and enhances the adaptability of the system.

key words: *P2P, network monitoring, traffic classification, QoS*

1. Introduction

Provisioning and improving the Quality of Service (QoS) – the ability to provide different priority to different applications, users, or data flows in the transmission process – has been one of the key concerns in realizing dependable and high-quality network services. Appropriate QoS policies could only be enforced when network traffic is categorized with high accuracy. However, accurate classification of network traffic according to their application types is made difficult by the popularity of obfuscation techniques such as packet encryption, random/changing ports, and proprietary protocols in data transmission.

In this paper, we present a study on network traffic analysis to support QoS operation within a cloud-like environment. Recent advances in cloud computing indicate that the most critical bottleneck in advancement of cloud services is the bandwidth limitation between users and cloud providers. This indicates that the analysis on the most bandwidth intensive application will contribute most to QoS management in a cloud environment, similar as in a conventional network environment.

Among the wide range of network applications that contribute most to the Internet traffic, peer-to-peer (P2P) file-sharing is regarded as the application type that most significantly affects QoS management in real network environments because of the following facts reported in a recent Internet study [1]:

- P2P generates the most traffic in the Internet – ranging from 43% to 70% across the surveyed continents;
- most P2P protocols are bandwidth-intensive, which results in significant deterioration of QoS;
- P2P file-sharing causes much controversy because of copyright infringement; and
- many P2P clients are vulnerable to cyber attacks and when compromised will lead to information leakage and other catastrophic problems.

Due to its popularity and efficiency, P2P is likely to continue being the dominating content-distribution protocol in the foreseeable future. Identifying and regulating this most bandwidth intensive part of the Internet traffic contributed by P2P file-sharing could not only support QoS operations such as resource reallocation and route planning, but help to protect the cloud infrastructure from risks such as malware contagion and privacy leakage as well. Moreover, P2P file-sharing protocols tend to intentionally employ obfuscation techniques to prevent detection or filtering and thus are regarded the most challenging protocol type. Therefore, in this paper, we focus our study on how to differentiate P2P file-sharing hosts from other hosts. With accurate identification and categorization of P2P traffic, a network operator may throttle P2P applications to ensure good performance of business critical applications. Network engineering problems such as workload characterization and modelling, capacity planning, and route provisioning could also benefit from accuracy identification of P2P traffic.

1.1 Related Research

Early research on P2P traffic analysis focuses on well-known port numbers assigned by the Internet Assigned Numbers Authority (IANA). However, this method is no longer viable due to the popularity of port fluctuations among contemporary P2P protocols [3]. It is estimated that 90% of P2P traffic are transfered on random ports [4]. Further advances lead to the introduction of signature based protocol classification known as *deep packet inspection* (DPI) [5], [6]. DPI searches for string patterns of the applications and perform classification on this basis. It is by far the most reliable way to classify teletraffic and is widely used in commercial products. However, inspection of application layer data usually runs into legal and privacy concerns. Moreover, DPI is also known as resource-expensive

Manuscript received January 10, 2012.
Manuscript revised May 22, 2012.

[†]The authors are with the National Institute of Information and Communications Technology, Koganei-shi, 184–8795 Japan.
[††]The author is with the Shandong University, China.
a) E-mail: bantao@nict.go.jp

DOI: 10.1587/transinf.E95.D.2888

and incapable to work on encrypted transmission.

The limitations of port-based and payload-based analysis motivate the use of transport layer statistics, which is independent from payload, for traffic classification. Bernaille et al. [7], [8] introduce a method which classifies a bidirectional flow using the packet-length sequence of its first four or five packets. This scheme could treat with encrypted packets and thus provides further improvement on the classification accuracy of a DPI method. However, the downfall is that it could be invalidated by simple techniques such as packet length padding or randomization.

More recent studies on P2P protocol classification based on sophisticated flow-level properties such as duration, packet size, and inter-arrival times are reported in [9], [10]. The method introduced by Erman et al. [9] classifies a flow based on the parameters extracted from its first 8 packets and refines the prediction when more packets become available. Huang et al. [10] applies a similar idea to analyze the so-called *talk blocks*, where each talk block contains the group of sequential packets sent in one direction. While deployment of early statistical parameters is proved to be successful in numerical studies, a classifier based on this scheme can be confused by manipulation and padding of early packet length.

In addition to flow-level features, host-level statistics are proved to be very helpful in identifying P2P protocols. Mostly exploited host levels include the social level, the functional level, and the application level [11], [12]. The social level measures the host behavior in terms of the number of interconnecting hosts; the functional level states the role of a host on the network (client, server, or both); and the application level identifies transport layer features such as the ratio of the number of interconnecting hosts to the number of active communication ports. Further study on transport layer analysis are reported in [13], [14], where additional heuristics, e.g., filtering flows by port numbers, are incorporated to improve the accuracy of transport layer analysis.

Some other approaches to host analysis are also reported. In [15], Hu et al. suggest that voting – summarization of the prediction results on multiple flows of a host – could help to improve the accuracy of protocol identification. In [16], Collins and Reiter report that host-level features such as the bandwidths, failed connections, packet length profiles, and packet volumes of individual flows could help to determine different applications acting at a host. And in [17], Hurley et al. investigate the application of a host-based classifier in real-time where host-level features are extracted from a limited number of early packets.

1.2 Proposed Method

In this paper, we present a new host-analysis scheme, which was first introduced in [18] and later improved in [19], [20], and report the recent progress of the study. The main idea of this study follows that of the payload-independent approach [12], i.e., two conditions are kept true during the analysis: (A) no access to user payload to respect user privacy, and (B) no assumption on reliable relationship between well-known port numbers and application protocols to treat with increasing complexity in today's P2P protocols. The contributions of this study are summarized in the following points.

First of all, a cost-effective host-level P2P traffic classification problem is formulated and solved by supervised machine learning methods. In addition to traditional host-level features such as statistics on payload volume and packet length that describe the bandwidth-intensive nature of P2P hosts, entropy based host-level features are introduced to grasp the one-to-many social-level characteristics of P2P transmission. Utilizing the 18 features extracted from IP packet headers, one can get a classification system with up to a 100% accuracy in classifying known P2P hosts from ordinary hosts, and a 98% accuracy in identifying previously unknown P2P protocols.

Second, tunability of the system, i.e., easy controllability of the trade-off between sampling rate and system response time is proven to be very efficient. Compared with traditional flow-level analysis presented in [11], [12], the assumption of 100% packet sampling rate to handle the initialization of TCP sessions is exempted, resulting in great ease in applying the system at different network access points to the monitored network. Compare with traditional host-analysis that usually takes days to gather sufficient behavioral information of the hosts [14], our method only requires a reasonable monitoring time, say, one or two minute, to perform reliable classification. Such prompt classification is designed to meet the requirement of an Internet Service Provider (ISP) or Cloud Service Provider for QoS policy implementation. To further alleviate the monitoring, analysis, and storage cost, feature selection is engaged to identify the most essential features that help differentiate P2P hosts from ordinary hosts, with very promising result reported, i.e., equally accurate classification could be achieved with only 3 out of 18 features.

Finally, a system framework using virtualization technology to generate labelled training set for evaluating P2P-host classification systems is introduced. The system could be regarded as a simulated cloud environment, which could lead to further insights of QoS management schemes in the cloud era ahead.

The rest of this paper is organized as follows. Section 2 presents the proposed scheme on P2P traffic monitoring. Section 3 describes the methodology for host-level analysis. Section 4 reports experimental results based on the proposed scheme. Section 5 draws the conclusion.

2. Tracing in Virtualized Environment

Virtualization has played and will continue to play a key role in cloud computing. Virtualization's ability to separate the Operating System (OS) and application from the hardware gives it ideal properties to best deliver on-demand services that are essential to cloud service provision. A typical scenario to consolidate enterprise servers using virtualization

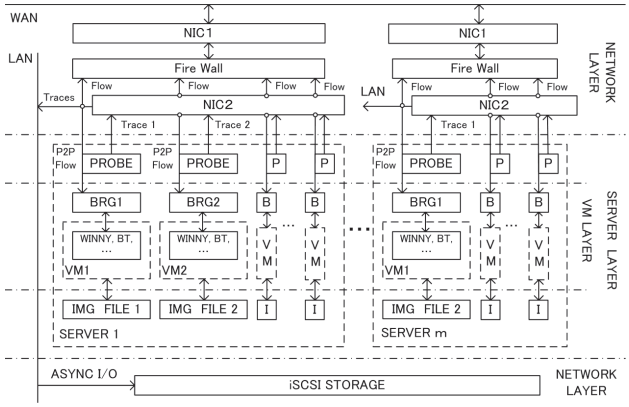

Fig. 1 Overall framework of the traffic monitoring and analysis system.

is to run multiple guest OSes on each server instead of just one to increase the utilization rate of every server. Generally, uncontrolled P2P file-sharing applications are considered harmful to the cloud infrastructure and thus are put into a list of prohibited applications. Without the luxury of a P2P-friendly cloud system, in [18], we built a cloud-like environment to collect background and P2P network traffic and evaluated the proposed scheme on collected traces. As shown in Fig. 1, the proposed system for P2P network trace collection is made up of three layers: a network layer, a server layer, and a virtual-machine layer. Here, an NIC is a network interface controller; a PROBE is a network monitoring tool to collect the traffic traces; a VM is a virtual machine; an IMG FILE is a file in the hypervisor system that simulates the file system for the guest OS; and a BRG is a bridge interface that combines an Ethernet interface with one virtual TAP (virtualize netork tap) interfaces.

2.1 Network Layer

The network layer has two functions: accessibility to the outside network and high-performance storage service for the upper layers. The Wide Area Network (WAN) interface connects to a broadband Internet connection with direct access to the Internet. With carefully tuned firewall rules, the specific P2P network is accessible from clients installed in the guest OSes. At the other side, the Local Area Network (LAN) interface connects to a reliable high-performance Ethernet, so that the local machines can forward the trace files to the storage server.

2.2 Server Layer

The server layer offers a virtualization environment to the guest OS's, captures the individual traces for each of them, and send the traces to the storage server on the network layer. A P2P network built with the Virtual Machine (VM) technology, has the following advantages. (1) It makes more efficient usage of system resource, which is one of the main topics of Green Computing. (2) Because the P2P network under study might bare some vulnerabilities that could be exploited by certain cyber attacks, VM can help to sandbox the P2P client so that enforces the hypervisor system free from risk. (3) The last but not least important, thanks to the fast system recovery and reboot capability of the VM technology, it is much easier to redo the experiment or adapt the system to analyze other P2P protocols than maintaining the same number of physical machines.

2.3 Virtual Machine Layer

The virtual machine layer is characterized by guest OSes where specific P2P clients are installed with Internet connection enabled. At each time we install only a single P2P client upon each guest OS and let it connect to the outside P2P network. To make a more versatile network that is able to simulate different network conditions, traffic control software can be installed in each guest OS. The good news is that most P2P applications offer an option to control the bandwidth assigned for file sharing.

The proposed scheme tries to strike a balance between traditional monitoring schemes such as transport layer analysis, flow-level analysis, and application level tracing. The proposed single-protocol exclusive network has the following merits. First, characteristics of a specific (P2P) application can be easily abstracted from the collected traces. Second, the traces are automatically labelled with good accuracy but little labor cost, suitable for supervised analysis. Finally, since the traces are collected at the network-level, the system built for one P2P network is reusable for any other (P2P) protocols.

3. Data Analysis

In the virtualization-based tracing system described above, by recording the name of the P2P client installed and run in the guest OSes, the traces are automatically labelled with the P2P protocol name. Because of its single-protocol-exclusive property, a collected trace carries essential information on the behavior of the protocol. Using these labelled P2P traces together with some background traces captured in the same network environment with P2P protocols restrained, we can define a classification task that differentiates P2P file sharing hosts (positive class) from ordinary hosts (negative class).

3.1 Host-Level Analysis

Our analysis makes use of statistical features, referred as discriminators hereinafter, extracted from the collected network traces and detects packet flows associated with P2P hosts. By using packet header information alone, we mitigate the collection-computation cost of an alternative signature based approach and prevent privacy concerns that could arise otherwise.

Traffic metrics such as the traffic volume, packet size, number of preserved connections are often good indicators of P2P protocols. Previously, these discriminators are often

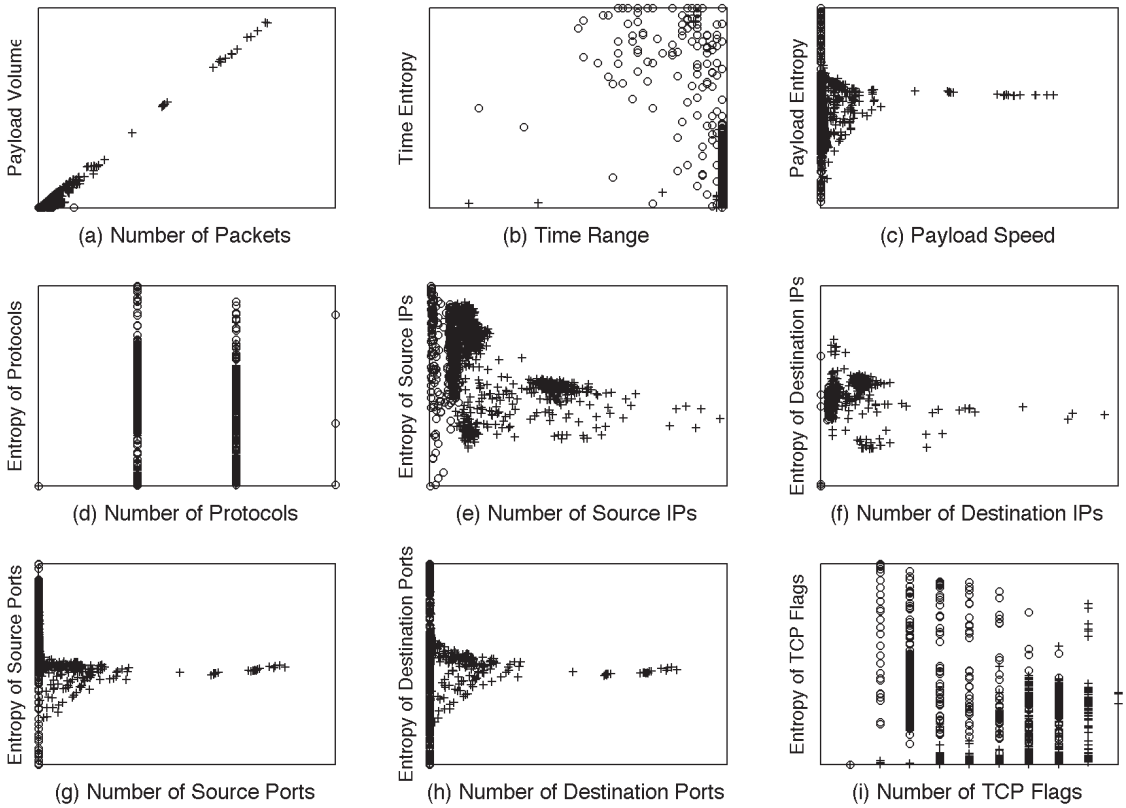

Fig. 2 Scatter plots in the multi-dimensional space defined by discriminators.

derived on *network flows* – traffic channels between communicating peers defined by the 5-tuple, i.e., {source address, destination address, source port, destination port, protocol}. The definition of network flow well suits traditional network activities such as email exchange, web access, and database access, under the Client-Sever (CS) model, and is effective in analyzing protocols such as HTTP, SMTP, Telnet, and DNS. However, recent P2P applications tend to use obfuscation to bypass conventional detectors. For example, most P2P clients use dynamic port numbers or reuse well-known port numbers of other protocols to prevent detection by port-number based detectors and use packet-length padding to circumvent packet-length based detectors. Therefore, at the network-flow level, there could be little difference between a P2P flow, a web browsing flow, or an FTP flow – all these protocols tend to maximize the bandwidth usage for better user experience. Consequently, flow-level statistics cannot offer sufficient information for differentiating P2P flows.

In this study we are interested in the status of a host, i.e., whether it is doing P2P file sharing, rather than the characteristics of its specific communication session/channel. For better capturing the decentralized nature of a P2P network, we proposed to go beyond the flow level to the host level for an overall view of the communication. To do so, we treat all the communications bounded to a target host (associated with a specific IP address) as a single stream and define discriminators upon these host-level streams. Discriminators are extracted from host-level streams within a fixed time window, and analysis is performed on basis of the discriminators.

3.2 Transport Layer Discriminators

For better classification accuracy on P2P hosts, it is expected that a discriminator vector extracted from a P2P host will show statistical difference from vectors associated with ordinary hosts. One of the most intuitive characteristics that separate P2P protocols from ordinary protocols is bandwidth-intensity. In Fig. 2, the host statistics are plotted in the 2-dimensional subspaces defined by randomly paired discriminators. It is easy to observe the difference of the two classes (P2P hosts shown as crosses and non-P2P shown as circles) in these bandwidth related discriminators. For example, in Fig. 2 (a), most P2P hosts are distributed at the upper part which indicates that payload volumes of P2P hosts are much higher than those of ordinary hosts.

In addition to the discriminators that well represent the bandwidth-intensive nature of P2P streams, it is worthwhile to note the connection pattern of P2P file-sharing protocols could also manifest their presence. To treat with instability and variable connectivity, most P2P file sharing applications tend to keep a large amount of connections between peering hosts. Thus the one-to-many connection pattern from a host upon the peering hosts will also be a key characteristic characterizing P2P streams. Accordingly, the distribution of the observed packets among peering hosts for a P2P host will be different from that of a web browsing client: the packet distribution over peering IP for a P2P client that

downloads content simultaneously from multiple host shall be more dispersed, while the packet distribution for the web browsing client shall be more concentrated on a few web servers. Figure 3 illustrates an example of the packet number distribution of a host that is downloading stream media using PPStream (Fig. 3 (a)) and another host that is doing web browsing using Firefox (Fig. 3 (b)). Each plot shows a distribution of packet numbers observed in a 1-minute period, where no normalization is made on the packet number for better perception. It is worthwhile to note two facts about the two distributions. First, the distribution in Fig. 3 (a) is much flatter than that in Fig. 3 (b), indicating the difference in the nature of the communication. Second, the number of peering hosts are comparable in the two graphs, which is different from our intuition. These observations suggest that the key factor to differentiate these flows shall not be the number of peering IPs but the shape of the distribution.

An ideal metric that captures the degree of dispersal or concentration of a distribution is the *entropy* [21]. In information theory, the entropy is a measure of the uncertainty associated with a random variable. In our case, it is used to measure the randomness of related attributes of the packets. These attributes could be the source IP address, the source port, protocol type or any other properties of interest that could contribute to separating the two classes. Take the source IP addresses appeared in a fixed time window as an example. The entropy, H_{SIP}, along the source-IP-address

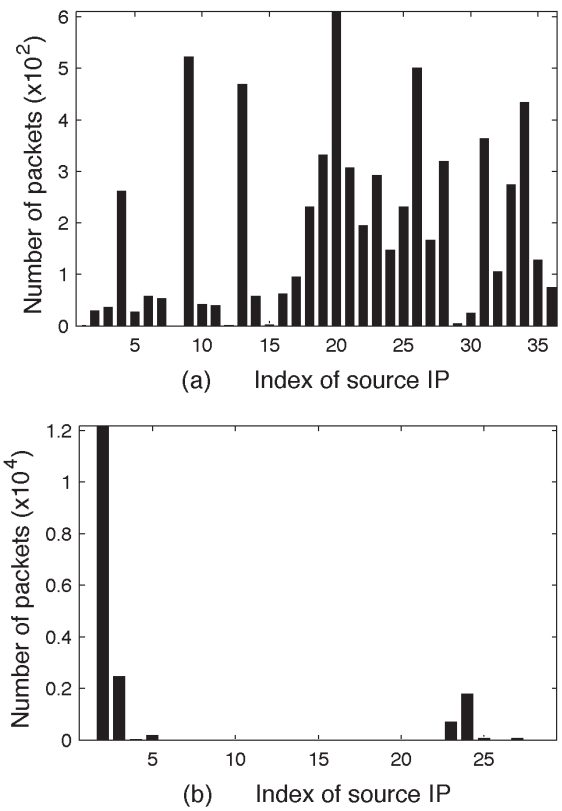

Fig. 3 Packet distribution over source IP. (a) PPStream. (b) Web browsing.

is calculated as

$$H_{\mathrm{SIP}} = -\frac{1}{\log_2 n} \sum_{i=1}^{n} p_i \log_2 p_i, \tag{1}$$

where n is the number of unique IPs and p_i is the probability that the ith IP shows up as source IP in the time window. In the above example, the entropy of the distribution in Fig. 3 (a) is $H_1 = 0.87$, that of the distribution in Fig. 3 (b) is $H_2 = 0.32$. Basically, a dispersed distribution tends to give an entropy value close to one, and a concentrated distribution tends to give a value close to zero. Therefore, the degree of dispersal or concentration for the distribution is captured by a scalar parameter.

Figure 2 shows the scatter plots in the multi-dimensional space defined by all discriminators we have defined. It is easy to percept the separability of the two classes in the 2-dimensional subspaces defined by paired discriminators: data in the same class are located close to each other but far from those in the opposite class.

3.3 Learning Methodology

Despite that many analytical models, e.g., clustering, function regression, association rule mining, etc., could help to characterize the host behavior based on the collected network traces, we find the classification model best fits the objective of this study – to identify whether a host is performing P2P file sharing operations at the moment. First, evaluation of unsupervised learning methods is known to be difficult and ad hoc without (or even with) a labelled data set. Second, when a benchmark data set is available, unsupervised methods could hardly beat a supervised methods on prediction accuracy because of the lack of class information in the training process. In the literature, there are some scenarios that unsupervised learning could generalize better on previously unknown classes. In our case, as a one-against-the-rest classification (i.e., P2P v.s. non-P2P) problem is formulated, there will be conceptually no unknown class, and thus it leaves little space for such a fightback for clustering. Moreover, classification is a better studied field than unsupervised learning with respect to the generalization ability of the models and there are a large pool of available tools to deal with extremely large data sets. Finally, as the above system provides us an cost-effective way to create high quality labelled data sets for training and evaluation of the learning methods, there is no special need to cling to unsupervised learning.

As our objective is to separate P2P hosts from ordinary nodes, the task is formulated as a *binary classification* problem. That is, we are given an empirical data set of ℓ samples,

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_\ell, y_\ell) \in \mathcal{X} \times \{\pm 1\}, \tag{2}$$

where $\mathcal{X}$ is the nonempty set of all possible *observations*, $\boldsymbol{x}_i$, and $y_i \in \{\pm 1\}$ are class labels. A positive sample ($y_i = +1$) belongs to the P2P class and a negative sample ($y_j = -1$) belongs to the non-P2P class.

We adopt the Support Vector Machine (SVM) [22] because of its generality and high prediction performance. SVM realizes the following idea. Given two classes of samples as in (2), the input vectors $\boldsymbol{x}_i$ are first mapped into a high (possibly infinite) dimensional feature space, $\mathcal{F}$, through a nonlinear mapping function Φ; then the optimal hyperplane that realizes the maximal margin in $\mathcal{F}$ is constructed. With the so called *kernel trick*, Φ is implicitly implemented by a kernel function $K(\cdot,\cdot)$, which equals to an inner product in the feature space. The decision function yielded by an SVM classifier turns to be a linear combination of the feature vectors $\Phi(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b \tag{3}$$

$$= \sum_{i=1}^{\ell} \alpha_i \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}) \rangle + b, \tag{4}$$

where $\boldsymbol{w}$ is the normal vector of the decision hyperplane in $\mathcal{F}$ and b the bias. $\boldsymbol{w}$ can be written as a linear combination of the feature vectors in $\mathcal{F}$, weighted by α_i. A feature vector with non-zero α_i is known as a support vector. A novel sample $\boldsymbol{x}$ with $f(\boldsymbol{x}) > 0$ is assigned to the positive class, otherwise it is assigned to the negative class.

3.4 Feature Selection

In Fig. 2, there are some redundant features can be observed in the scatter plots. For example, the time range (the y-axis in Fig. 2 (b)) offers little discriminative information in the classification – there is severe overlapping between different classes when all the data are projected onto it. Considering that redundant discriminators could not only introduce much complexity in the learning but also impose unnecessary costs on data collection, storage, and processing, it is considered helpful to evaluated the significance of all these discriminators and eliminate the irrelevant ones. Below we introduce the SVM Recursive Feature Elimination (SVM-RFE) [23] method that is used to identify the essential discriminators for P2P host classification. One of the reason for adopting this method is its capability to discover linear as well as nonlinear correlation among discriminators.

SVM-RFE implements the main idea of the backward selection method. Starting with a pool of all the discriminators, one builds a classifier and removes one or a subset of the discriminators from the pool that are least important in the classification. The elimination process is repeated until a predefined number of discriminators are left or all the discriminators are ranked.

In SVM-RFE, the eliminated discriminators are determined based on the ranking criterion defined in the following. In (3), suppose $\boldsymbol{\alpha}^*$ and $\boldsymbol{w}^*$ associate with the decision function that realizes the maximal margin, it is easy to check that,

$$\|\boldsymbol{w}^*\|^2 = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{5}$$

For a linear SVM, i.e., $K(\cdot,\cdot)$ is the inner product function, (5) can be simplified to

$$\boldsymbol{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* \boldsymbol{x}_i. \tag{6}$$

It is obvious that if some elements of $\boldsymbol{w}^*$ are zero, the elimination of the associated input discriminators will not lead to any variation in the decision function. Furthermore, a feature associated with an w_i^* close to zero may be considered insignificant and deleted without degeneration in generalization ability of the decision function. Thus, the significance of the kth feature could be measured by a ranking criterion

$$R_k = \sqrt{\|\boldsymbol{w}^*\|^2 - \|\boldsymbol{w}^{*(k)}\|^2} = \left| \sum_{i=1}^{\ell} y_i \alpha_i^* x_{ik} \right|, \tag{7}$$

where x_{ik} is the kth element of $\boldsymbol{x}_i$, and $\boldsymbol{w}^{*(k)}$ is obtained from $\boldsymbol{w}$ by setting all components x_{ik} to 0 for $i = 1, \ldots, \ell$.

The discussion can be extended to the nonlinear case where elimination of an input feature corresponds to deletion of multiple discriminators in the feature space. The contribution of the kth feature to $\|\boldsymbol{w}^*\|$ can be evaluated as

$$R_k = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* (K(\boldsymbol{x}_i, \boldsymbol{x}_j) - K(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(k)}))^{1/2}, \tag{8}$$

where $\boldsymbol{x}^{(k)}$ is the vector with the kth feature of $\boldsymbol{x}$ set to 0. Note that for the sake of simplicity and speedup of computation, $\boldsymbol{\alpha}^{*(k)}$, the solution of the optimization problem with the kth feature deleted, is supposed to be equal to $\boldsymbol{\alpha}^*$.

4. Experiments

In this section, we apply analysis on the traces collected from the cloud-like environment introduced in Sect. 2. The traces are collected using the following settings. For each of the eight-core servers, six Windows 2000 are installed as guest OSes. Related software clients are installed and run upon the guest OSes. A P2P trace is collected and labelled based on the client running on the guest OS. The background traffic is captured in a similar network environment where common network activities such as web browsing, FTP/HTTP downloading, and online gaming are permitted. In SVM training, we use the Gaussian kernel which is a universally adaptable kernel function with reliable performance in a wide range of applications.

Parameters of the classifier, i.e., width parameter of the kernel, γ, and error penalty parameter, C, are determined by 10-fold cross validation. All the reported results are averaged on 100 runs on the randomly shuffled versions of the same data set. Note that the network infrastructure such as the bandwidth and routing management policies may vary from one network to another. The optimal parameters for performing the classification will generally differ subtly from one case to another. Therefore, we provide here the parameter selection strategy adopted in the experiments, i.e., 10-fold cross validation, rather than listing up all the used parameters. Our experience shows that $C = 100$ and $\gamma = 1$ are good initial values for parameter tuning.

4.1 Experiment Settings

To evaluate the classification performance of the proposed scheme, we apply it to the data sets created using the following two settings.

In the first setting, we access the classification performance of the proposed scheme on known P2P protocols. To do so, we perform classification between the background traffic and two most popular P2P protocols, i.e., BitTorrent – the most popular P2P file-sharing protocol world wide and PPLive – a typical protocol of the new generation P2P applications known as P2PTV. Training and test are performed on trace data containing the same protocols. Hereinafter we refer the first test set as *test set 1* (TS-1 in figures).

In the second setting, the system's generalization ability to previously unknown protocols is taken into consideration. Here, training is performed using the same trace data as that is used in the first setting, however, in the test data, we add traces generated from previously unknown P2P protocols. Namely, we add traces of eMule and PPStream to the test data, where eMule is a popular P2P file-sharing protocol and PPStream is a mainstream P2PTV protocol. We refer the second test set as *test set 2* (TS-2 in figures).

4.2 Analysis on Time Window Size

In the first experiment, we explore the influence of the time-window size, w, on prediction accuracy. To justify whether a host is using P2P protocols, bounded traffic needs to be monitored for at least w seconds so that discriminators could be extracted from the captured trace during this period. In this sense, the window size is closely related to the response performance of the system. To access its influence, w is selected from $\{1, 2, 4, 8, 16, 32, 64\}$ (seconds), and as w changes its value, the variation of prediction accuracy on the two test sets are recorded and shown in Fig. 4 (a) and 4 (b), respectively.

In Fig. 4 (a), it is easy to spot that the discriminant information in the discriminators gradually increases as w increases from $1s$ to $64s$. When the sampling rate is 1, i.e., all captured data are used for feature extraction, the prediction rate increases from 95.57% ($w = 1s$) to 99.67% ($w = 64s$). Lowering the sampling rate, which is noted as a parameter r hereinafter, leads to a degeneration of the accuracy to some extent. Still, in all cases, the accuracy increases as the window size grows.

As for the results on test set 2 shown in Fig. 4 (b), we can say that including previously unknown protocols in the test set will render the data distribution different from that of the training data and thus results in a degeneration of the prediction accuracy. Figure 4 (b) shows that the conclusion that increment of w leads to higher prediction accuracy also applies to previously unknown P2P protocols, despite of more obvious fluctuations. When $w = 64s$, all the settings with different sampling rates achieve an accuracy rate above 88.75%. For $r = \frac{1}{64}$, the increment of the accuracy

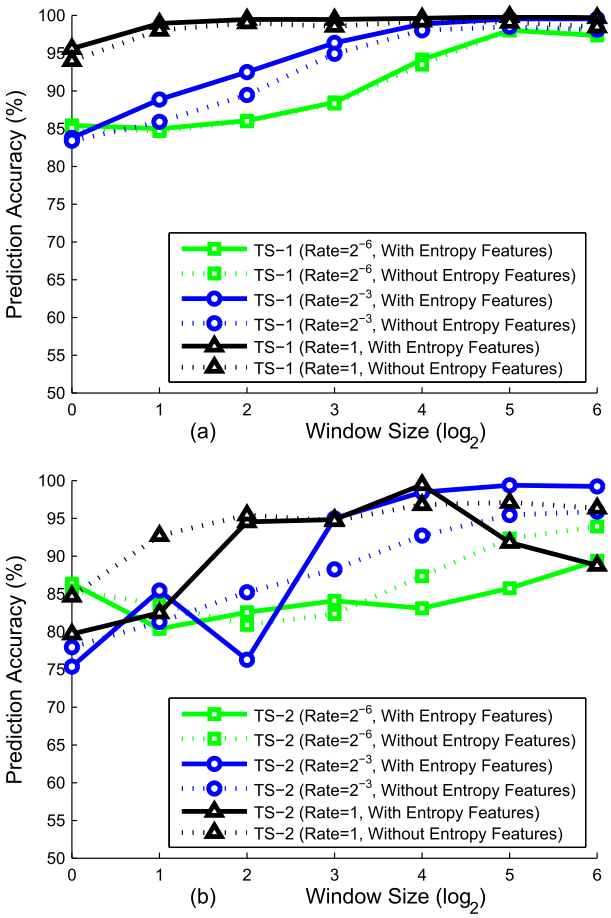

Fig. 4 Prediction accuracy v.s. window size.

appears to be very stable, in spite of a drop at the starting point.

As a reference, the dotted lines in Fig. 4 show the prediction accuracy without the entropy discriminators. In Fig. 4 (a), it is easy to spot that for test set 1, including the entropy discriminators always increases the separability of the two classes, resulting in increased prediction accuracies. In Fig. 4 (b), entropy discriminators work very well in some of the cases, e.g., when $r = \frac{1}{8}$. In some other cases, they lead to subtle degeneration of the prediction accuracy because of the increased complexity they have introduced. Further justification of the entropy features has motivated the study on feature selection, see Sect. 4.4. for more discussion.

4.3 Analysis on Sampling Rate

When monitoring high speed switched networks, the sampling rate r on the network interface is an important parameter that determines the scalability of system. Generally, we want to reduce the sampling rate for cost-effective traffic data collection, storage, and analysis. The second group of experiments are designed to verify the influence of sampling rate on the prediction accuracy. To do so, the traces are first captured using full sampling, i.e., $r = 1$, followed by a sub-sampling procedure using different r parameters

selected from $\{\frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$. As r changes its value, the variations of classification accuracy on the two test sets are recorded and shown in Fig. 5 (a) and 5 (b), respectively.

Figure 5 (a) looks similar to Fig. 3 (a), although at this time, the increment in sampling rate contributes to the increment in prediction accuracy. For $w = 1s$, the accuracy starts from 85.41% at $r = \frac{1}{64}$, slightly dropping to a minimum of 83.78% at $r = \frac{1}{8}$, and then increases gradually to 95.57% at $r = 1$. For $w = 8s$, the accuracy constantly increases from 88.44% at $r = \frac{1}{64}$ to 99.41% at $r = 1$. For $w = 64s$, except the beginning point at $r = \frac{1}{64}$, all other r values all give accuracy above 99.14%. We can also easily observe the analogy between Fig. 5 (b) and Fig. 4 (b). Similarly, discordance between the training set and test set has led to larger variations in the prediction accuracy. Still, we can find a trend that increment in the sampling rate generally leads to increment in prediction accuracy, except for the case of $w = 1s$.

As can be seen in Fig. 5 (a), including entropy discriminators could stably improve the prediction accuracy on test set 1. For test set 2, the best accuracy is obtained with entropy discriminators at $(w = 64s, r = \frac{1}{8})$, nevertheless, entropy discriminators also lead to remarkable fluctuations in prediction accuracy as other parameters change.

To summarize, the above experiments on windows size and sampling rate show that, more computing resources, i.e., longer observation time and/or higher sampling rate will provide us more knowledge about the behavior of the monitored host, and thus results in higher prediction accuracy. Another important discovery is that, the positive effect of increasing the window size of observation excesses the negative effect of decrement in sampling rate. This suggests that for better recognition accuracy and less system performance downgrade, we can keep a rather small sampling rate for network performance purpose while increase the window size until satisfactory generalization performance is achieved.

Entropy discriminators are remarkable in the following two aspects. First, for prediction on known protocols (test set 1), stable gain in prediction accuracy is guaranteed by incorporating entropy discriminators. Second, when used for predicting previously unknown protocols (test set 2), the optimal result are always obtained using these discriminators although they might also introduce some fluctuations when other parameters change. Above all, when the entropy discriminators are also taken into consideration, $w = 16$ and $r = \frac{1}{8}$ seems to be a very good parameter combination that supports good accuracy on prediction of P2P hosts (99.14% on test set 1 and 98.38% test set 2), without imposing a heavy impact on the monitored network.

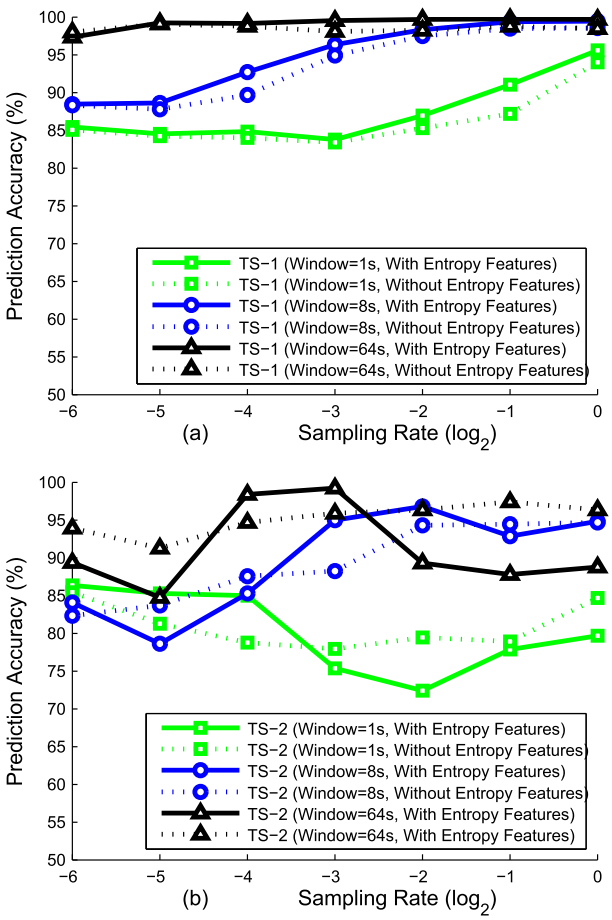

Fig. 5 Prediction accuracy v.s. sampling rate.

4.4 Feature Selection Results

Feature selection is performed on the traces to answer the questions that which are the most essential discriminators for classifying P2P hosts from ordinary hosts and what is the necessary number of discriminators for the classification. To simplify the discussion, we set $w = 16$ and $r = \frac{1}{8}$ in the following experiment. The obtained order of discriminators is shown in Fig. 5, with the features ranked in descending order of their significance. Then we train the classifiers on the data with the discriminator at the end of the sorted list removed at each step and record the prediction accuracy. According to Fig. 6, the most important discriminator is the *entropy over source ports*: we can get an accuracy of 92% using this single feature. This indicates that the scattering

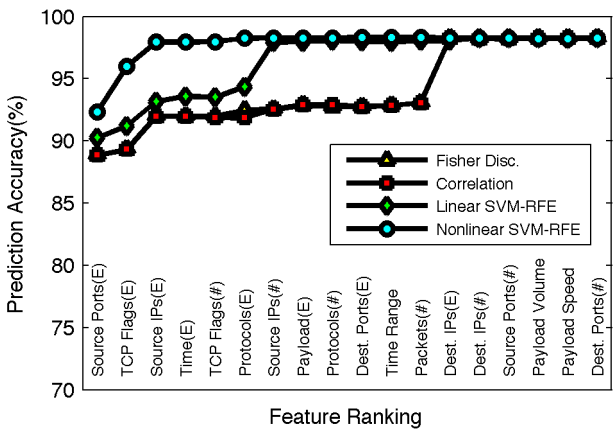

Fig. 6 Prediction accuracy against ranked discriminators.

of the traffic into different ports is one of the key features that characterizing P2P hosts. The most essential subset of discriminators is apparently *entropy over source ports*, *entropy over TCP flags*, and *entropy over source IPs*. Using only these three discriminators we could obtain a classifier as good as using all the discriminators. Adding other features to this discriminator set gives little improvement on prediction accuracy.

We have to note that some other entropy based features such as *entropy over time bins*, *entropy over protocol types*, and *entropy over payload volume* provide significant discriminant information as they are all ranked high in the list, despite that the first three discriminators seems to be perfectly matching this classification task. On the other hand, discriminators at the end of this list e.g., *number of destination ports*, *payload volume*, *payload speed*, and *number of source ports* which are presumed to be important are in fact redundant to the task and could be safely discarded without significant loss in prediction accuracy.

As a reference, the results of Linear SVM-RFE and two feature selection based on popular filter metrics, i.e., Fisher Discriminant (FD) and correlation coefficient (CC) are also reported in Fig. 4. The ranking orders of these methods are omitted for clarity. We can see that the FD based and CC based methods give almost of the same performance on the prediction accuracy. These linear method ranks traditional features such as *payload speed*, *payload volume*, and *number of source IPs* higher than entropy based features. This could be explained by the facts that entropy based features tend to nonlinearly correlate with the class label and are under estimated by the linear methods. The linear SVM-RFE method shows a better performance than the other two linear methods because of the employment of an advanced classifier for evaluating the significance of the features.

It is important to note that variation of the network infrastructure may affect the selected parameters for classifier training and further lead to subtle variation in the feature ranking. The good news is that for a network whose infrastructure is not changing from time to time, the obtained classification system is rather stable, as suggested by our numerical study over multiple runs. When in a dynamic environment, an effective strategy to keep the system at its optimal status is introducing adaptive learning strategies to the learning and keep updating the classification model with most recent information. Related topics are usually covered in online learning [24], which is a rapidly growing subfield of statistics and machine learning.

5. Conclusion

In this paper, we have presented a study on applying machine learning techniques for characterizing P2P file sharing hosts in the network for network engineering purpose. For better lightweightness and adaptability, we define informative discriminators based on the headers of the packets instead of deep payload inspection. For better accuracy, we propose to perform the analysis at host level so that entropy based discriminators that capture the one-to-many topological nature of a P2P file-sharing transmission could be defined. Numerical study showed that with appropriate parameter tuning the proposed scheme could realize lightweight (with $\frac{1}{16}$ sampling rate) and accurate prediction with high accuracy (above 98%) even for previously unknown P2P protocols. We also apply feature selection methods to identify the most significant features that help to differentiate P2P hosts from other hosts. The numerical experiments show that P2P hosts are best characterized by a group of three features: *entropy over source ports*, *entropy over TCP flags*, and *entropy over source IPs*. All the studies sums up to an effective monitoring and analysis system for P2P host classification with very low collection and storage cost.

References

[1] http://www.ipoque.com/en/resources/internet-studies
[2] M. Zhang, W. John, K. Claffy, and N. Brownlee, "State of the art in traffic classification: A research review," PAM Student Workshop, 2009.
[3] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," MASCOTS, 2006.
[4] N. Basher, A. Mahanti, C. Williamson, and M. Arlitt, "A comparative analysis of web and peer-to-peer traffic," Proc. 17th International Conference on World Wide Web, pp.287–296, 2008.
[5] L7-filter, Application layer packet classifier for linux, http://l7-filter.sourceforge.net/, accessed 2012-05.
[6] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," Proc. 13th International Conference on World Wide Web, pp.512–521, 2004.
[7] L. Bernaille, R. Teixeira, and K. Salmatian, "Early application identification," Proc. ACM International Conference on Emerging Networking Experiments and Technologies, 2006.
[8] L. Bernaille, R. Teixeira, I. Akodjenou, A. Soule, and K. Salmatian, "Traffic classification on the fly," ACM SIGCOMM Computer Communications Review, vol.36, no.2, pp.23–26, 2006.
[9] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," SIGCOMM, 2006.
[10] G. Huang, G. Jai, and H. Chao, "Early identifying application traffic with application characteristics," Proc. IEEE Conference on Communications (ICC08), pp.5788–5792, 2008.
[11] T. Karagiannis, A. Broido, M. Faloutsos, and K. Klaffy, "Transport layer identification of P2P traffic," Proc. 4th ACM SIGCOMM Conference on Internet Measurement, pp.121–134, 2004.
[12] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: Multilevel traffic classification in the dark," SIGCOMM, 2005.
[13] M. Perenyi, T. Dang, A. Gefferth, and S. Molar, "Identification and analysis of peer-to-peer traffic," J. Comm., vol.1, no.7, pp.36–46, 2006.
[14] W. John and S. Tafvelin, "A Heuristics to classify internet backbone traffic based on connection patterns," ICOIN, 2008.
[15] Y. Hu, D. Chiu, and J. Lui, "Application identification based on network behavioural profiles," Proc. 16th International Workshop on Quality of Service (IWQoS), pp.219–228, 2008.
[16] M. Collins and M. Reiter, Finding peer-to-peer file-sharing using coarse network behaviors, Lect. Notes Comput. Sci., Comput. Secur., ESORICS, vol.4189, pp.1–17, 2006.
[17] J. Hurley, E. Garcia-Palacios, and S. Sakir, "Host-based P2P flow identification and use in real-time," ACM Trans. Web, vol.5, no.2, p.7, 2011.
[18] T. Ban, R. Ando, and Y. Kadobayashi, "Monitoring and analysis of network traffic in P2P environment," NICT J., vol.54, nos.2/3, pp.31–39, 2008.

[19] T. Ban, S. Guo, Z. Zhang, R. Ando, and Y. Kadobayashi, "Practical network traffic analysis in P2P environment," Proc. 2nd International Workshop on TRaffic Analysis and Classification (TRAC2011), 2011.
[20] T. Ban, S. Guo, M. Eto, D. Inoue, and K. Nakao, "Entropy based discriminators for P2P teletraffic characterization," Proc. 2011 International Conference on Neural Information Processing (ICONIP 2011), 2011.
[21] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," SIGCOMM, 2005.
[22] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, 1995.
[23] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Mach. Learn., vol.46, no.1-3, pp.389–422, 2002.
[24] V. Vovk, A. Gammerman, and G. Shafer, Algorithmic Learning in a Random World, Springer Science Business Media, 2005.

Tao Ban received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1999, the M.S. degree in engineering from Tsinghua University, Beijing, China, in 2003, and the Ph.D. degree in information science from Kobe University, Kobe, Japan, in 2006. He is currently an expert researcher with the Network Security Research Center (NSRC), National Institute of Information and Communications Technology, (NICT) Tokyo, Japan.

Shanqing Guo is an associate professor with the School of Computer Science and Technology, Shandong University, China. He obtained his Ph.D. from Department of Computer Science, Najing University. His research interests include software and network security, with an emphasis on program analysis and reverse engineering techniques at network level and virtual machine level.

Masashi Eto received LL.B degree from Keio University in 1999, received the M.E. and Ph.D. degrees from Nara Institute of Science and Technology in 2003 and 2005, respectively. From 1999 to 2003, he was a system engineer at Nihon Unisys, Ltd., Japan. He is currently a senior researcher at NICT, Japan. His research interests include network monitoring, intrusion detection, malware analysis and autoconfiguration of the Internetworking.

Daisuke Inoue received his B.E. and M.E. degrees in electrical and computer engineering and Ph.D. degree in engineering from Yokohama National University in 1998, 2000, and 2003, respectively. He joined the Communications Research Laboratory (CRL), Japan, in 2003. The CRL was relaunched as the NICT in 2004, where he is the director of Cybersecurity Laboratory in NSRC.

Koji Nakao is the Information Security Fellow in KDDI, Japan. Since joining KDDI in 1979, he has been engaged in the research on multimedia communications, communication protocol, secure communicating system, and information security technology for the telecommunications network. He is also an active member of Japan ISMS user group, which was established in the 1st Quarter of 2004. He is the board member of Japan Information Security Audit Association (JASA) and that of Telecom-ISAC Japan, and concurrently, a Technical Group Chairs (ICSS: information communication system security) of the Institute of Electronics, Information and Communication Engineers. He received the B.E. degree of Mathematics from Waseda University, in Japan, in 1979. He received the IPSJ Research Award in 1992, METI Ministry Award and KPMG Security Award in 2006, Contribution Award (Japan ITU), NICT Research Award, Best Paper Award (JWIS) and MIC Bureau Award in 2007.