5 Applied Machine Learning Theory

5-1 Monitoring and Analysis of Network Traffic in P2P Environment

BAN Tao, ANDO Ruo, and KADOBAYASHI Youki

Recent statistical studies on telecommunication networks outline that peer-to-peer (P2P) file-sharing is keeping increasing and it now contributes about 50-80 % of the overall Internet traffic[1]. Moreover, more and more network applications such as streaming media, internet telephony, and instant messaging are taking a form of P2P telecommunication. The bandwidth intensive nature of P2P applications suggests that P2P traffic can have significant impact on the underlying network. Therefore, analyzing and characterizing this kind of traffic is an essential step to develop workload models and possible amelioration in network traffic engineering and capacity planning. In this paper, we introduce an adaptive system for handy P2P trace capture and analysis. The system can efficiently organize limited resources to build a reliable and tractable network. Traces captured by the system are analyzed for characterization of Winny behavior with very interesting results reported.

Keywords
Traffic monitoring and analysis, Virtual machine, Peer to peer network, File sharing.

1 Introduction

To promote the reliability, availability, and stability of the Internet, P2P network analysis is becoming a research area attracting more and more attention in research as well as network administrative respect because of the following reasons: (1) The proportion of P2P traffics is now prevalent over the Internet and keeps increasing. (2) Many P2P applications are bandwidth-intensive which leads to excessive network congestion and possible dissatisfied customers and customer churn. Appropriate capacity planning can only be available when P2P traffic is identified with high accuracy. (3) P2P file sharing always causes quite a bit of controversy because of legal issues especially copyright law violations. (4) Most P2P clients are vulnerable to malware attacks and when compromised will lead to serious information leak and other catastrophic problems.

However, traditional monitoring and analysis techniques do not support analysis of evolving P2P network applications which show more sophisticated characteristics. First, current P2P networks bear more advanced infrastructure and demonstrate more complicated traffic patterns than traditional applications. Second, presently most P2P networks make use of customized or dynamically allocated port numbers: a P2P client can even easily operate on HTTP port 80. This renders traditional analysis methods based on well-known port numbers assigned by ICANN[2] not applicable to analyze P2P traffic. Then, to circumvent both filtering firewalls as well as legal issues, recent P2P applications not only operate on top of nonstandard, custom-designed proprietary protocols, but also are intentionally disguising their traffic as normal traffic. Finally, there is an increasing tendency in the P2P protocols to support payload encryption.

Towards a tractable solution for P2P network monitoring and analysis, we address two problems in this paper. First, we propose a system structure for efficiently employing a limited amount of network and computer resources to implement a comparatively large P2P network. The introduced system has multiple appealing points. (1), Traffic traces collected from the system share the same characteristics of the real network traces. (2), Easy adaptability to multiple P2P networks is guaranteed. (3), Access to a large network is not required. The second problem we will discuss in the paper is the characterization of Winny network behavior. Winny is a typical anonymous P2P network originated in Japan. Characterization of Winny network is considered very important regarding both network engineering and security. We will mainly use flow-level information for the analysis.

The rest of this paper is organized as follows. Section **2** gives a general review of related research on P2P network tracing and characterization. In section **3** we describe the proposed system structure for network application trace capture. Section **4** reports some preliminary experimental results on P2P traffic analysis based on the proposed system. Conclusion is drawn in section **5**.

2 Related work on traffic monitoring and analysis for P2P networks

Internet traffic monitoring and analysis has always been a handy tool to fight against threats that impact the network, to prevent abusive or illegal usage of critical internet resources, and to minimize harm and distress caused by malicious people or software. However, as mentioned, P2P networks usually show more sophisticated characteristics than traditional Internet applications, many work needs to be done to adapt traditional traffic monitoring and analysis systems to analyze P2P traffic. In this section we review some related researches on P2P traffic monitoring and analysis.

2.1 Network-level tracing

Network-level tracing usually refers to IP-level packet monitoring at suitable points in the network infrastructure. Because network-level tracing is transparent to the P2P network, and is capable to analyze and compare multiple P2P systems simultaneously with other domain applications, it has been the main research stream. One of the drawbacks of Network-level tracing is that depending on the access point to the network as well as identification accuracy, large local bias can be introduced. Thus in order to gather a sufficient sample of traffic, monitoring program must be deployed at a key point in the network infrastructure, e.g. the gateway to an academic network.

2.1.1 Transport layer based analysis

Network traffic generated by traditional applications, e.g. Web, Ftp, Telnet, can be identified based on the well-known ports registered to the ICANN port list[2]. In the pre-P2P era, using port numbers less than 1024 or that appears in the ICANN port list was sufficient to identify most Internet traffic. Currently, this technique is not applicable for determining the traffic of P2P, streaming, and other new applications because the occurrence of any of the following situations will render the port-number based method inapplicable. First, many applications use dynamic port numbers, e.g. MS Windows Media Server/Player. Second, different applications may use the same port numbers simultaneously. Finally, proprietary protocols may use unregistered port numbers. Still, transport layer information are very important for traffic analysis. In particular, flow based approaches define a sequence of packets based on the transport layer information and use its statistical information for further analysis. According to the study in[3], although not perfect, transport layer information can still help to identify a considerable part of P2P traffic.

2.1.2 Payload-examination-based method

Payload examination is probably the most powerful but most resource-consuming technique to detect the dynamic ports of streaming media applications and P2P applications. In the case of media streaming, a control session as well as a data session are established between the client and server. The port number of the data session is determined dynamically by the negotiation between the client and server via the control session. Carefully examining the control session can help to find the port number of the data session. However, not only packet payload capture and analysis usually arrive at legal, privacy, and financial obstacles, but it also subjects to some technical defects. On the one hand, reverse engineering a growing number of poorly documented P2P protocols is usually considered a tedious work. On the other hand, for a P2P protocol with payload encryption, decode of user payload can be technically impractical. The tools mmdump[4] and SM-MON[5], apply payload examination to differentiate streaming media traffic from other Internet traffic.

2.1.3 Signature-mapping-based method

Researches show that identification of different Internet traffic using signature-based method is promising in some situations[6]. To extract the signature of a type of application, a portion of payload with distinguishable information from other packets is examined for all related applications. Usually, for privacy reasons, these payloads can only include the IP header and a limited number of bytes adjacent to the header. The limited payload information sometimes may not be enough for the increasing number of applications, considering the situation that P2P applications are trying to disguise their existence. Another drawback of signature based method is the large amount of offline work to discover the signatures of individual applications. Thus automated signature generation is a promising technique to relief the workload of analysts[7].

2.1.4 Flow-level characterization

Some of the P2P traffic characterization researches are based on the statistics or patterns of the packet flows[9][10]. In a Packet switching network, a packet flow or traffic flow is defined as a sequence of packets from one particular source to a single destination. In the IP network, traffic can be divided into flows by the 5-tuple source IP, destination IP, protocol, source port and destination port. The commonly accepted flow timeout is 64-second as proposed in[8], i.e., if no packet arrives in a specific flow for 64 seconds, the flow expires. Features such as host distribution and traffic volume are used to characterize the traffic. While flow-based analysis contributes to valuable insights into P2P traffic characteristics, it has some limitations regarding the inability to obtain application-level details.

2.1.5 Hybrid systems

In[4], a payload based and a non-payload-based methodologies are proposed to identify P2P traffic. The payload-based approach uses heuristics such as well known port number, frequently observed signatures in a 16-byte payload, and source and destination IP addresses. On the other hand, the non-payload based method uses no knowledge on the user payload information, but rather statistics on TCP/UDP pairs and port pairs to identify users who maintain a large number of distinct connections at the same time. Their method was reported to be able to identify 95 % P2P flows with a false positive rate of 10 %. In[11], both packet-level and flow-level information is applied to analyze P2P traffic. The experiments show that teletraffic characteristics in different P2P networks can significantly differ so that detailed studies of teletraffic in various P2P environments are suggested for better understanding.

2.2 Application-level tracing

Depending on the characteristics to be discovered, one can also apply application-level tracing tools to trace and analysis the traffic of a specific application. According to the mode the monitoring program works in, application-level tracing methods are classified into two categories.

2.2.1 Passive application-level tracing

Passive application-level tracing is performed by monitoring the resource discovery and network maintenance messages that a P2P node sends and receives while communicates with other peers at the application level. Usually, a modified client is used which passively logs the message it is asked to route but does not participate in other interactions. Passive application-level tracing can be performed easily without the necessity to access to a key point in the network infrastructure. However, it is only transparent to the specific P2P network and is not expected to trace a significant subset of a P2P network.

2.2.2 Active application-level tracing

Active application-level tracing addresses the problem of discovering global network information when access to the network infrastructure is impossible. It employs an aggressive querying and connection policy so that the monitoring peer attempts to connect to and interrogate as much of the P2P network as possible. Crawling peers on the P2P network can be intentionally guided to maximize the size and typicality of trace data. Note that this sometime casts shade on the transparency of the collected traces because of the peer behavior is markedly different from an ordinary peer with more reconnections and resource-discovery messages invoked.

In this paper, we want to make a balance between all the approaches mentioned above. On the one hand, to offer an insight into a specific network, we set up a one-protocol-exclusive network and collect the traces in this network for analysis. Data collected from this network can be assumed to only belong to this application. Such an exclusive network has two merits. First, application specific characteristics can be abstracted from the traces data. Second, the data are automatically labeled with good reliability but little laboring, and can be further analyzed by supervised learning methods. On the other hand, because the traces are collected at the network-level, the methodology applied to one P2P network can be easily generalized to another. We can accumulate as many P2P applications as needed by redoing the same experiment with different P2P applications installed at each time. Thus with only a tractable network, traces with good variation can be made without any access to the network infrastructure of a large-scale network.

3 Design and implementation of P2P traffic tracing system

In this section, we describe the design and implementation of a P2P traffic tracing system. Figure 1 illustrates the overall design of the system, which consists of three layers: network layer, server layer, and virtual machine layer.

3.1 Network layer

The network layer has two functions: accessibility to the outside network and high-performance storage service. The 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 local machines. On the other hand, the LAN interface connects to a reliable high-performance Ethernet, so that the local machines can forward the trace files to the storage server. For performance consideration, the iSCSI protocol is adopted in our system to connect local machines to the remote storage server. The purpose of using a separated storage server is to relief the workload of local machines. Note that a machine may have multiple guest operating systems and traffic capturing tools running simultaneously. This may render traffic analysis impossible at the same time. Moreover, collected data from multiple machines may offer statistically reliable information of the network.

3.2 Server layer

At the server layer, a system Virtual

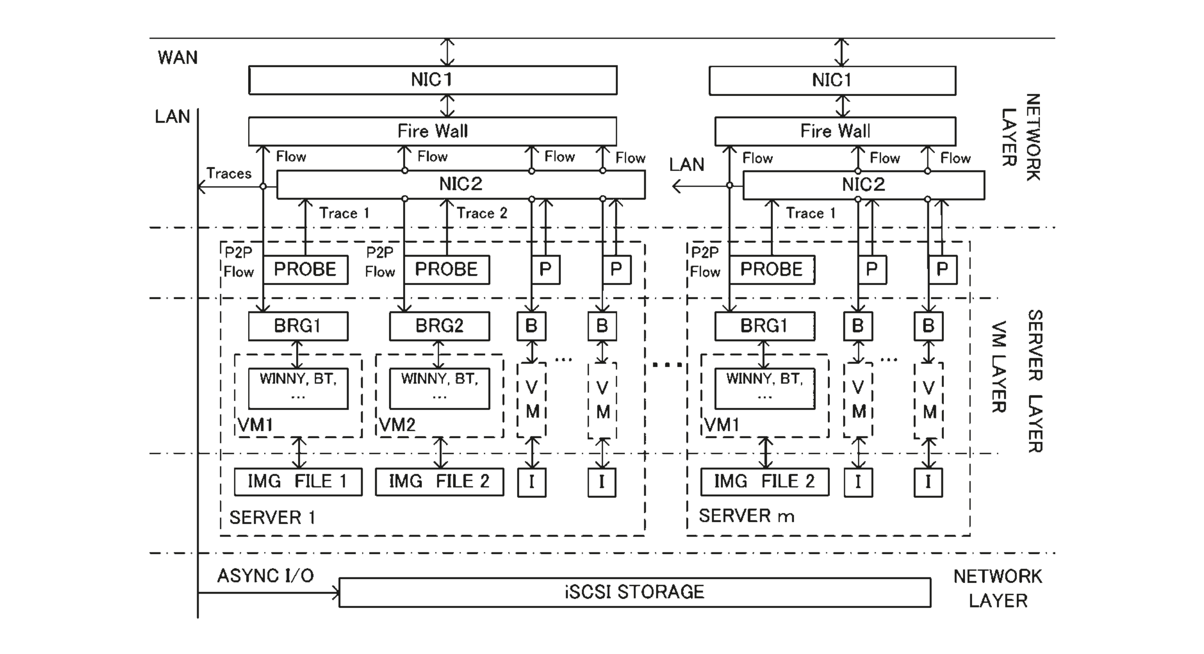

Fig.1 *Architecture of the capturing and analysis system*

Machine (VM) monitor (also known as hypervisor)[12] is installed to allow the multiplex of the underlying physical machine between different virtual machines. Upon the hypervisor which simulates the system services of a physical machine, multiple guest operating system are installed, serving as isolated machines. Via the VM technology, we can have the following advantages over a network composed of physical machines: (1) we can have multiple OS environments co-existing on the same computer, in strong isolation from each other. This helps us to build a comparably large-scale network environment without the necessity to handle with so many machines. (2) It makes more efficient usage of the system resources, which is one of the main topic of Green Computing. To collect related traffic to a specific P2P application, only one P2P client is installed and run upon an OS. This will generally leads to great waste of the processing ability of a machine. However, with the VM technology, we can assign suitable number of tasks, i.e. the number of VMs, for each server according to its system specification. Apparently, with the same resource, we can achieve a much larger network environment. (3) A third reason is that, since the P2P network is not secure, there is a possibility that the system might be compromised by some attacks. VM can help to sandbox the OS so that render the hypervisor system safe from possible risks. (4) 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.

Another function of the server layer is capturing the individual traces for each of the guest OSes and forwarding the data to the storage server via an Ethernet connection.

3.3 Virtual machine layer

At the virtual machine layer, the guest OSes are connected to the server layer by the virtual NIC interface simulated by the hypervisor. The traffic is then forwarded to the Internet. At each time we install only one P2P client upon each guest OS and let it connect to the outside P2P network. Note that current implementations of the hypervisor as the KVM, VMware, Xen, does not support traffic controlling, thus the bandwidth is equally shared by the guest OSes. To make a more versatile system that is able to simulate different network conditions, traffic control software can be installed in each guest OS. A good news is that most P2P applications offer an option to control the bandwidth assigned for file sharing.

In our experiments, we managed to run more than twenty VMs with each of our Dell PE 2950 servers. And with the 6 servers at hand, we were able to build a P2P network with more than 100 nodes. Remember that our goal is not to set up an isolated simulation of P2P network. Thus 100 nodes with individual settings on bandwidth and other options can provide us P2P traces with reliable characteristics of the network.

4 Experiments

To testify the feasibility and performance of the proposed system structure, we apply it for traffic analysis of three kinds of applications, namely, Winny, BitTorent, and miscellaneous network applications. For each of the six PowerEdge servers, we run 16 virtual machines with Windows 2000 installed. Qemu is chosen as the hypervisor software for performance reason.

4.1 Experiment Ⅰ

Like most of the second generation P2P networks, Winny also uses super nodes to improve the network scalability. When the conversation between two peers is established, Winny compares the uploading-bandwidth setting of the two and regards the peer with the higher value as a genteel node. Then searches are mainly sent in the genteel direction. This mechanism results in three groups of nodes. The super nodes which have higher bandwidth and computation power mainly serve as proxy and indexing servers for other peers. At the same time, super nodes also have more chance to download the required file. Middle level nodes which have ordinary network and computation resource get file from super nodes and offer service for lower nodes. Lower nodes only offer limited services to other nodes.

Apparently, uploading-bandwidth is the most important parameter to determine the role of a node in the Winny network. The other fact influences a node's behavior is Winny's working mode. Winny has two working modes: flood query mode and downloading (uploading) mode. Flood query refers to the process that Winny sends out searches to neighboring nodes and receives replies. During flood query, mainly control messages are exchanged among peers. While during the download process, mainly content messages are transferred. So there could possibly be some difference between traffic traces captured in different working modes. In the first experiment, we try to discover how the uploading-bandwidth and the working mode influence Winny's behavior. We set the uploading-bandwidth of Winny clients to 8 ranks: 819.2, 409.6, 204.8,102.4, 51.2, 25.6, 12.8, and 6.4 Kbps. We also set half of the Winny clients in flood query mode and the rest in downloading mode. Reported results are averaged over the collected traces over an one-hour period.

Figure 2 shows the number of conversations initialized between a Winny client and its neighbors. We can see that a Winny in downloading mode generally creates more conversations than one in flood query mode. More detailed analysis shows that conversations created by a low-bandwidth Winny are usually weaker with respect to connection speed and lasting time. A high-band Winny in flood query mode also actively participates in transferring queries between other peers and thus initializes a comparatively large number of conversations.

The result in Fig. 3 is apparent. For a Winny in downloading mode, the larger the bandwidth, the more packets transferred during a unit time frame. For a Winny in flood query mode, the situation is similar, however, there are not as many packets transferred as in the downloading mode. Figure 4 validates our prediction that the control messages and content messages are different in size. Hence difference in averaged packet size can be a basic

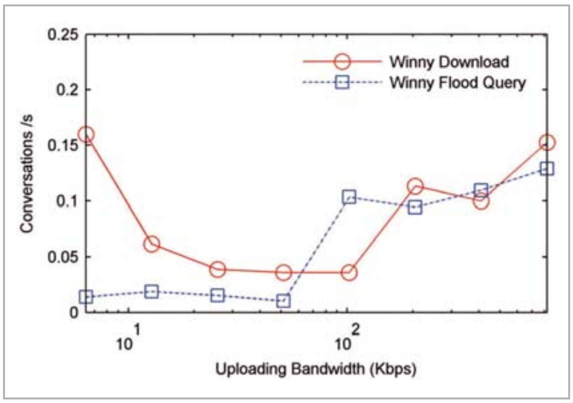

Fig.2 Conversations per second

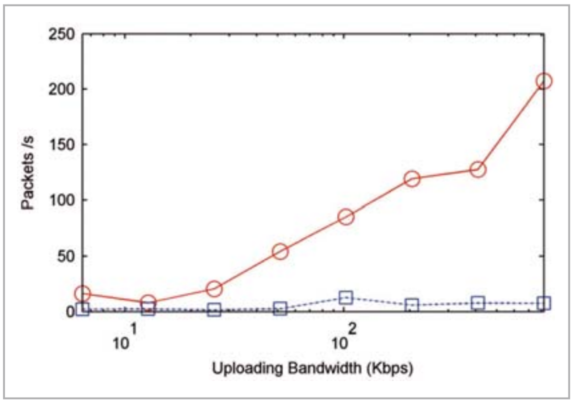

Fig.3 Packets per second

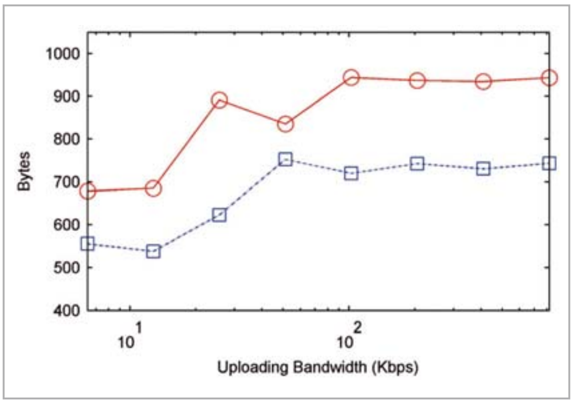

Fig.4 Average traffic size

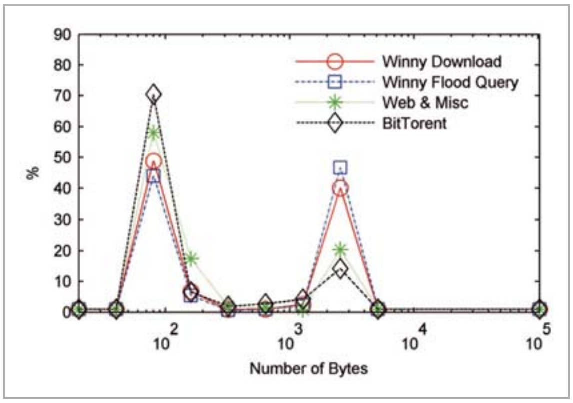

Fig.5 Packet size distribution

heuristic to identify Winny clients in different working mode.

4.2 Experiment Ⅱ

In the second experiment, we try to explore the discriminant ability of the packet size distribution of different network applications. The same VM network is built for Winny, BitTorrent, and miscellaneous web applications including web browsers, ftp clients, and ssh clients.

In Fig. 5, we show the packet size distributions of the mentioned applications. Winny traffics under different working modes appear to have very similar distributions. This means that packet size distributions can be a trustable feature to characterize the Winny traffic. BitTorrent and miscellaneous applications show noticeable difference between Winny's traffic. However, all of the four traces shows prevalent traffic in packet length ranges 40-79 and 1280-2559. This suggests that packet length distribution does not have significant discriminant ability for these applications. To make difference between them, we should either consider detailed distribution information defined in much finer bins or incorporate other knowledge or heuristics.

5 Conclusion

In this paper, we have proposed a Virtual Machine based traffic monitoring system framework. Thanks to the VM technology, the system can efficiently use available network and computation resources to build a comparatively large network environment. Even without access to a large network, clean and reliable network-level traces can be collected by setting up an exclusive network for a specific P2P protocol. Moreover, the system can be adapted to various P2P networks and network applications with little effort.

In the experiment section, traces collected by the network have shown statistical properties. By incorporating more information at the packet-level, flow-level, and transport-level, the system is considered promising to offer insight into the behavior of evolving P2P networks together with various emerging network applications.

References

1 http://www.ipoque.com/news_&_events/internet_studies/internet_study_2007.

2 Internet Corporation for Assigned Names and Numbers, "http://www.icann.org/"

3 T. Karagiannis, A. Broido, M. Faloutsos, and K. Klaffy, "Transport Layer Identification of P2P Traffic," Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, pp. 121-134, 2004.

4 J. van der Merwe, R. Caceres, Y. Chu, and C. Sreenan, "mmdump-A Tool for Monitoring Internet Multimedia Traffic," ACM Computer Communication Review, 30 (5):48-59, 2000.

5 H. Kang, H. Ju, M. Kim, and J. W. Hong, "Towards Streaming Media Traffic Monitoring and Analysis," Proceedings of 2002 Asia-Pacific Network Operations and Management Symposium, pp. 97-108, 2002.

6 S. Sen, O. Spatscheck, and D. Wang, "Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures," Proceedings of the 13th International Conference on World Wide Web, pp. 512-521, 2004.

7 P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: Automated Construction of Application Signatures," Proceedings of the ACM SIGCOMM Workshop on Mining Network Data, pp. 107-202, 2005.

8 K. Claffy, H.W. Braun, and G. Polyzos, "A Parametrizable Methodology for Internet Traffic Flow Profiling," IEEE JSAC, 13 (8):1481-1494, 1995.

9 S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic across Large Networks," Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, pp. 137-150, 2002.

10 M. Kim, H. Kang, and J. W. Hong, "Towards Peer-to-Peer Traffic Analysis Using Flows," Proceedings of the 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, pp. 55-67, 2003.

11 R. Bolla, R. Rapuzzi, and M. Sciuto, "Monitoring and Classification of Teletraffic in P2P Environment," Proceedings of the 2006 Australian Telecommunication Networks and Application Conference, pp. 313-318, 2006.

12 J. E. Smith and R. Nair, "The Architecture of Virtual Machines," Computer 38 (5): 32-38. 2005.

Ban Tao, *Ph.D.*

Expert Researcher, Traceable Secure Network Group, Information Security Research Center

Network Security, Machine Leaning

ANDO Ruo, *Ph.D.*

Researcher, Traceable Secure Network Group, Information Security Research Center

Network Security, Software Security

KADOBAYASHI Youki, *Ph.D.*

Guest Expert Researcher, Traceable Secure Network Group, Information Security Research Center

Network Security