2-5 Efficient Traceback Method for Detecting Illegal Access

KAI Toshifumi, NAKATANI Hiroshige, SHIMIZU Hiroshi, SUZUKI Ayako, and TSUKAMOTO Katsuji

The amount of damage by illegal access is increasing with the spread of the Internet. Especially the DoS (Denial of Service) and DDoS (Distributed DoS) attacks cause system down and often have serious impacts on the society. Various attacker detection techniques have been proposed until now, of which characteristics in performance and easiness of implementation are discussed in this paper. Based on the discussion, we propose hybrid traceback method that solves the drawbacks of the exiting techniques. Advantages of this proposed scheme to the exiting ones are clarified by some numerical models and experiment.

Keywords
Traceback, DDoS attacks, Illegal access

1 Introduction

Information-communications networks have grown to become an integral part of the social infrastructure, and a variety of services are now provided over these networks. On the other hand, we have seen an increasing number of incidents of attacks aimed at disrupting these services. Typical examples include DoS (denial of service) and DDoS (distributed DoS) attacks. These attacks often use packets with forged source IP addresses. Therefore, it is difficult for victims to track down the real source of such an attack and to implement countermeasures.

One method of locating the source of attacks involves tracing the path of illegal packets back to their source (Fig.1).

However, the Internet consists of autonomous systems (ASs) such as ISP network, government networks, and other systems managed under individual policies, and it is virtually impossible to perform traceback across several of these ASs under a single method. It is thus considered necessary to execute traceback between ASs (inter-AS) and within each AS (intra-AS) in a hierarchical manner[1][2].

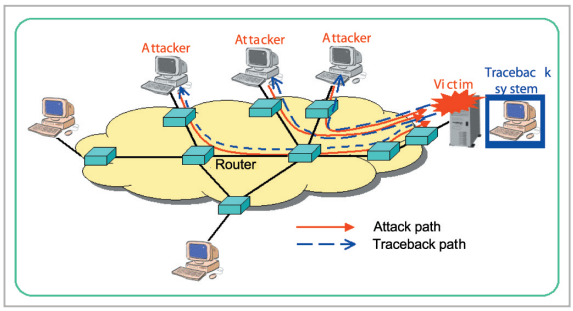

Fig.1 *Traceback*

Various intra-AS traceback methods have been devised to date[3]-[7], but each method has both advantages and disadvantages. Some techniques have been proposed in which a number of these methods are used in combination[8][9], but the effectiveness of this approach has not been verified.

We researched mechanisms and combinations of existing intra-AS traceback methods designed to capitalize on their relative strengths and make up for their respective shortcomings[9][10], and devised a method

that is superior to existing methods both in terms of performance and practicality. Based on this method, we then devised a new intra-AS traceback method that overcomes the disadvantages of existing methods[11]. We also devised and proposed a new inter-AS traceback method and verified its performance when used in conjunction with the intra-AS traceback methods[12].

However, this paper reports only on the newly devised intra-AS traceback methods, and does not deal with inter-AS traceback. First, we will evaluate the features of three existing traceback methods (the ICMP, marking, and Hash approaches) currently considered most effective. We then describe the methods (interlock and uTrace) we have devised (our original methods) for the rapid traceback of DDoS attacks that cannot be addressed adequately with existing methods. Further, we propose new methods (Hybrid I and II) consisting of our original methods and the Hash approach. We then describe the superiority of our original methods based on comparisons with existing methods in terms of success rates per traceback time and ease of implementation on a real network. We will also discuss the traceback performance of the new methods under high-traffic conditions.

2 Existing methods

Table 1 shows three typical IP traceback technologies and examples of implementation. Each of these methods consists of traceback data-acquisition modules mounted on routers (or external devices) and a traceback terminal that collects traceback information from these modules to identify packet paths.

Table 1 *Existing traceback methods*

Name	Implementation example
ICMP method	iTrace **iTrace-II ***
Marking Scheme	FMS AMS **AMS-II ***
Hash method	Paffi **SPIE ***

2.1 ICMP method

(1) Overview

The ICMP method samples packets with a certain probability and generates special ICMP packets called "iTrace packets" to send traceback information on these packets to the traceback terminal.

As a specific example of implementation of the ICMP method, we will describe the iTrace-II approach we developed through modifications to iTrace[3]. To increase the sampling rate while keeping network load low, iTrace-II generates an iTrace packet that can send several sampled packets at once. Table 2 shows the operational procedure of iTrace-II. When the traceback terminal collects a specified number of iTrace packets from all routers on the attack path, the traceback operation is successful.

Table 2 *Operation of iTrace-II*

Operation of a module on a router
Operation of a module on a router (i) Samples forwarded packets randomly with probability "P" (ii) Writes the following sampled packet information to an iTrace packet: • Number of bytes of IP header and payload • IP addresses of preceding, present, and next routers • Time of sampling (iii) When the number of sampled packets reaches limit L, this iTrace packet is sent to the traceback terminal. (iv) Repeats (i) through (iii) above (takes no special action during traceback)

(2) Features

iTrace-II can send a fairly large amount of traceback information at once through the generation of special packets.

2.2 Marking scheme

(1) Overview

In a marking scheme, each module writes traceback information directly to sampled packets, instead of generating special packets for traceback. This represents a major difference from the ICMP method.

As a typical example of implementation, we will describe AMS-II[5]. Table 3 shows the operation of this method. When the trace-

back terminal (featuring a router map) collects hash values (restored from fragments) from all routers on the attack path, the traceback operation is successful. Note that a path confirmation threshold must be reached at all routers.

Table 3 Operation of AMS-II

Operation of module on a router
(i) Samples forwarded packets randomly with probability "P" (ii) Calculates a 64-bit Hash value from its own IP address, divides it into eight 8-bit fragments, and writes one to the ID field (16 bits) of the randomly sampled packets (iii) Writes fragment numbers (0 to 7) and the hop count from the victim terminal on the remaining eight bits (sampled packets then sent) (iv) Repeats (i) through (iii) above (takes no special action during traceback)

(2) Features

The marking scheme does not place load on the network due to the generation of special packets for traceback. Therefore, it is possible to perform traceback fairly rapidly even if the volume of packet flow per attack terminal is low, as in the case of DDoS attacks.

2.3 Hash method

(1) Overview

Unlike the two methods above, the Hash method uses the traceback terminal to query routers actively. In the case of SPIE[6], a typical implementation of this method, routers save hash values on all in-transit packets, and the traceback terminal queries these routers on a packet basis.

(2) Features

Since routers save hash values for all packets, it is possible to trace back single-packet attacks with high accuracy.

2.4 Disadvantages of existing methods

Table 4 lists the range of traceable attack packet flow volumes and disadvantages of the existing methods.

Table 4 Comparison of existing methods

Name		ICMP (iTrace-Ⅱ)	Marking (AMS-Ⅱ)	Hash (SPIE)
Volume of packet flow per attack terminal	Large amount (DoS)	○	○	—
	Small amount (DDoS)	×	△	—
	Single packet	—	—	○
Problem at the time of introduction		No problem	Need to modify router maps/headers	Cost (size of required memory is a bit large)

○: High performance △: Medium performance
×: Low performance —: Not available

In consideration of network load, the ICMP method is recommended for use at a low sampling rate (1/20,000). This method can perform traceback only when tens of thousands of attack packets are sent.

The marking scheme requires that the traceback terminal always features an accurate router map. Since the module at each router alters the ID field of IP headers, this scheme cannot adapt to an environment in which fragments are generated. These factors render it considerably difficult to adopt this scheme in actual networks.

In the Hash method, it is necessary to exchange dozens of query packets per traced packet. Therefore, this method is not suitable for traceback of attacks involving a high volume of packets. In addition, each router requires a fairly large amount of memory to save the hash values.

3 Newly devised methods

Combining the ICMP and Hash methods, we devised an interlock method with a performance comparable to that of a marking scheme against multiple-packet attacks. In a previous paper[10], we proposed a Hybrid I method consisting of this interlock method and ordinary Hash methods. Later, we devised a UDP (uTrace) method that alone provides higher performance than the marking scheme. We then combined this method with the Hash method to arrive at the Hybrid II method. In this section, we describe these new methods.

3.1 Hybrid I method

3.1.1 Configuration of Hybrid I

Upon occurrence of a multiple-packet attack, Hybrid I will perform traceback by the interlock method, which consists of the ICMP (iTrace-II) and Hash (SPIE) methods; when a single-packet attack occurs, Hybrid I will perform traceback using the Hash method alone. These two modes, selected according to the type of attack, operate independently. Since we have already described the Hash method, the following sections will introduce an overview and features of the interlock method.

3.1.2 Overview of interlock method

iTrace-II and SPIE modules are mounted on each router, but these two modules do not interfere with each other within the router. No functional changes have been made to existing modules. The traceback terminal receives packets from iTrace-II and queries SPIE.

Upon reception of a traceback request from the IDS (intrusion detection system) or the network administrator when an attack occurs, the traceback terminal will check sampled iTrace packets sent from the routers for attack packets. If attack packets exist, the terminal will use SPIE to immediately trace these packets back to their source.

3.1.3 Features of interlock method

In the interlock method, iTrace packets trigger SPIE to execute traceback. It is possible to locate the attack path if the attack packets are sampled by at least one router on the path. Therefore, this method is faster than iTrace-II, which cannot identify the attack path until the attack packets are sampled by all routers on the path.

3.2 Hybrid II method

3.2.1 Configuration of Hybrid II

When a multiple-packet attack occurs, Hybrid II will use the uTrace (UDP) method; when a single-packet attack occurs, Hybrid II will use the Hash method alone, as in the case of Hybrid I. The following sections describe the uTrace method.

3.2.2 Configuration of uTrace

A uTrace module is mounted on each router. This module checks in-transit packets to select those that match the characteristics of the intended packets, writes traceback information to a UDP packet (called a "uTrace" packet) as in the case of iTrace-II, and sends the packet to the traceback terminal. The terminal, on the other hand, sends a traceback request message that includes the characteristics of the intended packets and receives uTrace packets from the routers. Based on the traceback information written to the uTrace packet, the terminal selects the next router to which it will send the traceback request message.

3.2.3 Operation of uTrace

Figure 2 shows an overview of the uTrace operation. To start traceback, the traceback terminal needs to have information on the characteristics of the intended packets and the IP address of the victim terminal's immediate router. The characteristics data includes protocol numbers and port numbers of higher-level protocols. The traceback terminal receives these data items from the IDS or network administrator and follows the steps below to execute traceback:

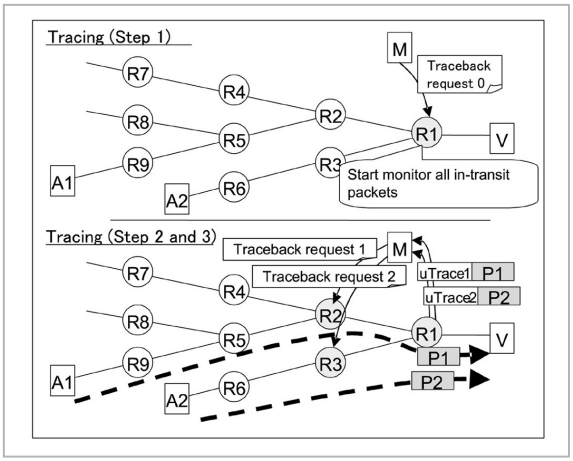

Fig.2 Overview of uTrace operation

(Step 1) Transmission of traceback request messages

The traceback terminal sends a traceback request message (Traceback Request 0) to a specified router. This message includes information on the characteristics of the intended packets, the terminal's IP address, a request

ID, the requested number of uTrace packets, and an effective period for the request.
(Step 2) Transmission of uTrace packets

After receiving the traceback request message from the terminal, the router will monitor all in-transit packets until the end of the effective period of the request, and will select packets (P1, P2) that match the characteristics of the intended packets. The router will then write traceback information to a uTrace packet and send this packet to the terminal.
(Step 3) Selection of next router

When receiving the uTrace packet, the traceback terminal will add the IP address of a router (R2, R3) (located immediately before the current router) to the traceback list. If this address has not been added to the traceback list, the terminal will send traceback request messages (Traceback Requests 1, 2) to that address.

Each time attack packets are detected, the traceback terminal can collect traceback information from the routers sequentially (from the victim terminal's immediate router to the attacker terminal's immediate router) through repetition of the above steps.

4 Performance evaluation

We established mathematical models to determine performance of individual traceback methods and compared the new and existing methods.

4.1 Evaluation models

To compare performance of individual traceback methods quantitatively, we used the evaluation models described in Sections ***4.1.1*** through ***4.1.5*** below.

4.1.1 Traceback

In computer forensics, the goal is to locate the path of attack packets in addition to identifying the attacker (terminal). In this paper, "traceback (attack detection)" refers to the location of both the attacker terminal and the path of attack packets.

4.1.2 Environmental conditions

The evaluation models are based on the assumption that evaluation is performed in an environment in which ideal IDSs and networks are used. An "ideal IDS" would be able to isolate attack packets accurately from non-attack packets, and would issue an alert upon occurrence of an attack. An "ideal network" would not be subject to any communications delays, congestion, or packet losses.

The volume of normal traffic flow is sufficiently high so that it is possible to ignore a delay in sending sampled packets to the traceback terminal when the iTrace-II or interlock method is used.

4.1.3 Attacks

The traceback methods mentioned in this paper are not affected by the actual data contained in the packets. The volume of attack packet flow (packets/sec) is therefore the only variable parameter on the attacker side.

4.1.4 Network

To render evaluation as simple as possible, we selected an S-branch tree as the attack path. The necessary parameters included the number of tree branches (S) and the hop count between the attacker and victim terminals. All end nodes of this tree are attacker terminals.

4.1.5 Path confirmation threshold

The traceback terminal only trusts information that has been collected in excess of this threshold value. The larger the value, the higher the accuracy; however, with larger values more time is required for traceback.

4.2 Mathematical models

Table 5 lists common symbols and values employed in our mathematical models. We used "fb" (number of successes, number of trials, and success rate) as a binomial distribution function.

Table 5 *List of symbols used in mathematical models*

Parameter	Symbol	Unit
Sampling rate	P	—
Path confirmation threshold	B*	Piece
Volume of attack packet flow per attack terminal	A	packets/sec
Traceback time	T	Sec
Hop count	H	Hops
Number of tree branches	S	—
Traceback success rate	Q	—

* Definitions differ depending on method

4.2.1 Mathematical models of existing methods

Equations 1 and 2 below show the relationships between traceback time and accuracy in the ICMP method (iTrace-II) and marking scheme (AMS-II) described in Section **2**, respectively. Note that Equation 1 does not allow for a delay in combining and sending sampled packets.

$$Q = \prod_{d=0}^{H-1}\left(1 - \sum_{k=0}^{B-1} fb(k, ATS^d, P)\right) \quad \text{Eq. 1}$$

$$Q = \prod_{d=0}^{H-1}\left(1 - \sum_{k=0}^{B-1} fb(k, ATS^d, P(1-P)^{H-d-1}/8)\right)^8 \quad \text{Eq. 2}$$

4.2.2 Mathematical models of interlock method

In the interlock method, if at least one of the routers on the attack path sends iTrace packets, it becomes possible for SPIE to identify the entire attack path. Therefore, the traceback success rate can be expressed by Equation 3, which does not depend on the number of attacker terminals or the network topology.

$$Q = 1 - \sum_{k=0}^{B-1} fb(k, ATH, P) \quad \text{Eq. 3}$$

4.2.3 Mathematical model of uTrace

The uTrace method does not depend on probability. The trace operation succeeds after a sufficient time has elapsed to enable acquisition of the required traceback information. This period lasts from the start of traceback until the passage of attack packets in an amount equal to hop count H + (path confirmation threshold - 1). Therefore, the relationship between traceback time and accuracy can be expressed by Equation 4.

where

$$Q = \left\{ \begin{array}{l} 0 \quad \text{where} \quad T < \dfrac{H + (B-1)}{A} \\ 1 \quad \text{where} \quad T \geq \dfrac{H + (B-1)}{A} \end{array} \right\} \quad \text{Eq. 4}$$

4.2.4 Comparison based on mathematical models

To compare performance of traceback methods against DDoS attacks, we determined the minimum volume of attack-packet flow per attacker terminal (packets/sec) that can be traced back by each method with a traceback time of 10 seconds. We considered a certain volume to be "traceable" if the success rate was 95% or higher. Table 6 lists conditions other than the volume of attack packet flow. This set of conditions is equivalent to a DDoS attack in which the number of attacker terminals (SH) is 1,024 and the total volume of packet flow (ASH) is 10,240 packets/sec. Table 7 shows the performance of traceback methods determined based on the respective mathematical models.

We compared the performance of these methods under identical conditions. The interlock and AMS-II methods can trace back one-tenth the volume of attack-packet flow of iTrace-II. On the other hand, uTrace can perform traceback of less than one-hundredth of the iTrace-II volume. These methods do not have an upper limit to the traceable volume of packet flow per attacker terminal; in other words, they can trace back attack packet flow no matter how large the flow is. Therefore, uTrace can trace back a wider range of attacks than other methods.

Table 6 Parameters of evaluation models

Parameter	Value
Hop count (H)	10
Number of branches (S)	2
Volume of attack packet flow per attack terminal (A)	10[packet/sec]
Path confirmation threshold (B)	2
Sampling rate	(iTrace-II, interlock) 1/4000 (AMS-II) 1/20
Number of sampled packets to be combined (L)	5

Table 7 Traceable volume of attack packet flow

Name	Minimum attack packet flow traceable for ten seconds [packet/sec]
ICMP method (iTrace-Ⅱ)	1904.5
Marking scheme (AMS-Ⅱ)	181.3
Interlock method (iTrace-Ⅱ+SPIE)	189.8
UDP method (uTrace)	1.1

4.3 Test-bed experiments

4.3.1 Experiment procedure

To verify uTrace performance, we conducted experiments on a test bed using the configuration shown in Table 8. We determined a fixed attack packet flow volume per attacker terminal (50 packets/sec) and varied the hop count from the attacker terminal to the victim terminal: three, five, 10, 15, and 20. The SYN flood attack type was used, with a path confirmation threshold of two. We measured the time to complete traceback. We tested traceback 60 times with each hop count, and compared the average traceback time with theoretical figures.

Table 8 Test-bed configuration

Equipment or environment	Item	Description
Network	Number of PC routers	40
	Number of end nodes	160 (30 actual PCs)
	Type of network	Ethernet (100base)
PC router	OS	Reahat Linux7.3
	CPU	Celeron2.0GHz
	Memory	512Mbytes

4.3.2 Experimental results

Table 9 shows the results of experiments with different hop counts. We confirmed that actual measurement and theoretical values were nearly the same on an attacker-terminal basis. The actual detection time was shorter (quicker) than the theoretical time. This is because the theoretical calculations are based on the assumption that an ideal IDS is used, and therefore, the first packet goes through the router not immediately but rather 1/50 (0.02) sec. after the start of traceback. Incidentally, differences were noted between the mathematical model and actual experiments on a test bed in terms of network delay time, processing time at the traceback terminal and routers, and TCP ACK response from the victim terminal subject to SYN flood attack. The experimental results show that these factors have little impact on traceback time.

Table 9 Results of experiments with different hop counts

Hop count	Theoretical value (sec)	Measurement value	Standard deviation
3	0.080	0.077	0.0072
5	0.120	0.118	0.0077
10	0.220	0.217	0.0077
15	0.320	0.320	0.0107
20	0.420	0.419	0.0156

5 Discussion

5.1 Performance of proposed methods

As described in Section ***4.2.4***, uTrace features higher performance than the AMS-II and interlock methods. Like the ICMP method, uTrace poses no difficulties in particular in installation. Table 10 summarizes these points.

As seen in this table, the Hybrid II method consists of the UDP (uTrace) and Hash methods; in combination, these methods provide high traceback performance against various types of attacks.

Table 10 Comparison of new and existing methods

Name		ICMP (iTrace-II)	Marking (AMS-II)	Hash (SPIE)	Interlock (iTrace-II + SPIE)	UDP (uTrace)
Volume of packet flow per attack terminal	Large amount (DoS)	○	○	—	○	○
	Small amount (DDoS)	×	△	—	△	○
	Single packet	—	—	○	—	—
Problem at the time of introduction		No problem	Need to modify router maps/ headers	Cost (size of required memory is a bit large)	Cost (size of required memory is a bit large)	No problem

○: High performance △: Medium performance
×: Low performance —: Not available

5.2 Operation under high-traffic conditions

With a high volume of network traffic flow, uTrace modules on routers may miss some attack packets and traceback speed may decrease due to the delay in message transmission between the traceback terminal and routers. We thus conducted two sets of experiments to test these eventualities.

Firstly, we studied the probability of gen-

eration of uTrace packets with a high volume of router traffic. When we forwarded one packet/sec of attack traffic and 25,000 packets/sec of background traffic between the routers (RedHat Linux 7.3, Pentium 3.0 GHz, memory: 1.0 GB), a uTrace module missed 51.33% of the attack packets, and this percentage of uTrace packets failed to be generated. However, in addition to the uTrace module software that runs on routers, we have already developed a dedicated uTrace system that can be installed in parallel with the network using a network processor. Conducting the same experiment using this system, we were able to confirm that the uTrace module misses no attack packets even with 50,000 packets/sec of background traffic.

Secondly, we placed a bottleneck on the network to constrict bandwidth, sent uTrace packets and background traffic exceeding the bottleneck bandwidth, and studied the rate of packet loss. We discovered that when traffic flow is twice the bottleneck bandwidth, the rate of uTrace packet loss becomes nearly 50%.

These results suggest that without adequate measures, traceback speed will decrease under high-traffic conditions. However, by attaching the dedicated uTrace system and exchanging uTrace packets and traceback request messages through the dedicated line, it is possible to avoid the effects of traffic and to perform traceback as predicted under the mathematical models.

6 Conclusions

Based on the evaluation of typical existing traceback methods, we devised hybrid methods and demonstrated their superiority over existing methods. We then tested these new methods using actual PC routers on a test bed to verify performance.

Although this paper dealt only with intra-AS traceback methods, we have also devised a new inter-AS traceback method, and have already verified its performance in test-bed evaluations when used in conjunction with the hybrid methods.

Acknowledgements

This research was carried out based on the Research and Development on Security of Large-Scale Networks project commissioned by the National Institute of Information and Communications Technology. We would like to express our gratitude to NICT for its guidance and support.

References

1 K. tsukamoto et al., “A consideration on Traceback over inter-ASes”, IEICE General Conference, Mar. 2004 (in Japanese)

2 M. Oe et al. “An Inplementation and Verification of a Hierarchical Architecture for IP Traceback”, IEICE Transaction on. Communications. Vol. J86-B, No. 1 pp. 1486-1493 Aug. 2003. (in Japanese)

3 S. Bellovin, “ICMP Traceback Message”, InternetDraft:draft-bellovin-itrace-00. txt, submitted Mar. 2000.

4 S. Savege, D.Wtherall, A. Karlin, and T. Anderson, “Practical Network Support for IP Traceback”, Proceedings of Sigcomm 2000, Aug. 2000, Stockholm, Sweden

5 D. Song and A. Perrig, “Advanced and Authenticated Marking Schemes for IP Traceback”, Proc. IEEE INFO-COM, April 2001.

6 Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Kent ST, and Strayer T, “Hash-Based IP Traceback”, Proc. of the ACM SIGCOMM 2001 Conf. San Diego, Aug. 2001.

7 A. Hayakawa et al. “FUSEI AKUSESU HASSHINGEN TUISEKI SISUTEMU NI OKERU TUISEKI JIKAN NO HYOUKA”, IPSJ 64th General Conference, Mar. 2002. (in Japanese)

8 R. Yamada et al. “The Design and Implementation of The Hybrid Traceback Scheme for Finding True

Source of Packets", IPSJ DSM2003, Jan. 2003. (in Japanese)
9 N. Fukuda et al. "Research and Development for Traceback System in Large-Scale Networks", IEICE General Conference, Mar. 2004. (in Japanese)
10 T.Kai et al. "Traceback Method for Searching Attackers Transmitting Packets with False Source IP Addresses", IPSJ DPS-WS 12th, Dec. 2004. (in Japanese)
11 T. Kai et al. "Efficient Traceback Method for Detecting DDoS Attacks", IPSJ CSEC 27th, Dec. 2004. (in Japanese)
12 T. Kai et al. "Inter AS Traceback Method for Detecting DDoS Attacks", IPSJ CSEC 28th, Mar. 2005. (in Japanese)

KAI Toshifumi

Advanced Technologies Development Laboratory, Matsusita Electric Works, Ltd.

Networks Security

NAKATANI Hiroshige

Senior Engineer, Advanced Technologies Development Laboratory, Matsusita Electric Works, Ltd.

Networks Security

***SHIMIZU Hiroshi**, Dr. Agr.*

General Manager, New Business Planning Office, Matsusita Electric Works, Ltd.

Networks Security

SUZUKI Ayako

Core Network Business Headquarters, NTT Advanced Technology Corp.

Networks Security

***TSUKAMOTO Katsuji**, Ph.D.*

Professor, Kogakuin University Dept. of Computer Engineering

Multimedia Communication,Network Security