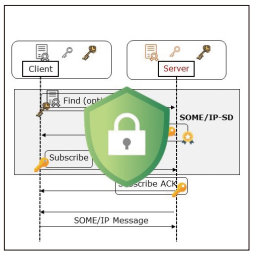


A Proposal of the Security Extension Protocol for an Automotive Ethernet

Hiroshi UEDA*, Ryo KURACHI*, Hiroaki TAKADA, Masayuki INOUE, Shogo KAMIGUCHI, and Kenya WADA

Ethernet and SOME/IP (Scalable service-Oriented MiddlewarE over IP) have been increasingly applied to modern vehicles. Although SOME/IP serves as an automotive middleware solution for control message transmission, it lacks security measures, making it vulnerable to various attacks. To address this issue, we present a security extension protocol for automotive Ethernet. Furthermore, we evaluate the protocol and demonstrate its effectiveness.

Keywords: automotive security, In-vehicle network, automotive Ethernet, SOME/IP

1. Introduction

As the number of functions in automobiles has increased in recent years, the electronic control systems in automobiles have become more complex, and many control computers, known as "electronic control units" (ECUs), have been installed in automobiles. In addition, as the sensors in automobiles have become more sophisticated, a high-capacity communication protocol has become required, and automotive Ethernet has been attracting attention. In order to ensure real-time performance, various protocols extending the physical and media access control (MAC) layers have been proposed for automotive Ethernet communications. Furthermore, the SOME/IP (Scalable service-Oriented MiddlewarE over IP) protocol has been proposed as a protocol for its upper layers in order to create a more flexible control system.(1) In conventional electronic control systems, statically arranged functions are achieved by receiving data broadcast from each node. On the other hand, the SOME/IP protocol assumes that services are received from dynamically arranged functions in each node in order to allow flexibility in the arrangement of functions on the electronic control system. Therefore, before initiating SOME/IP communication, SOME/IP-SD (Service Discovery) is executed.(2) However, there are security challenges in establishing dynamic connections via SOME/IP-SD. This paper proposes an extended protocol to securely establish communication between two nodes. Furthermore, the proposed protocol is compared with various security protocols, and the necessity of the proposed protocol is discussed.

2. SOME/IP Protocol and Related Research

2-1 SOME/IP protocol

The SOME/IP protocol is a communication protocol whose specifications were established by the AUTomotive Open System Architecture (AUTOSAR), an industry organization that standardizes automotive software platforms. This communication protocol is characterized by the fact that it is designed for server-client communication, whereby the client establishes a communication channel by dynamically discovering and connecting to a server that executes a service. This protocol for service discovery is called the "SOME/IP-SD protocol." After the communication channel between the server and the client is established, the service is initiated by the server sending a SOME/IP message to the client, as shown in Fig. 1.

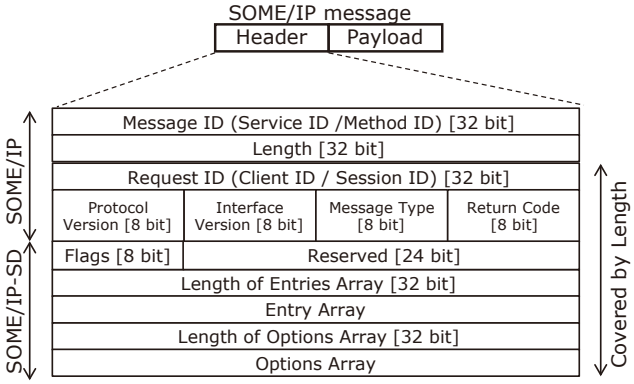


Fig. 1. SOME/IP message format

2-2 SOME/IP-SD protocol

The SOME/IP-SD protocol defines a sequence that both the server and client nodes execute at startup. After starting up, the client sends a Find message specifying the Service ID and waits for a response from the server. The server responds to the client through the Offer Service attaching connection information, when the Find Service from the client specifies the Service ID executed by itself. The client that receives the Offer Service requests a connection by sending a Subscribe message using the connection information provided by the Offer Service. The server responds using "Subscribe Eventgroup Ack" or "Subscribe EventGroup Nack" to indicate whether it permits or denies the connection. When Subscribe Eventgroup Ack is returned, the connection in SOME/IP-SD is treated as completed, and the client can receive the service with

SOME/IP messages thereafter. This sequence is shown in Fig. 2. The SOME/IP-SD protocol is used until the Subscribe Eventgroup Ack is returned, and then data is sent using SOME/IP messages.

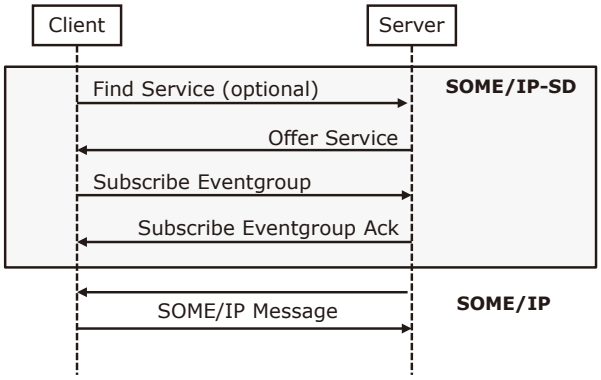


Fig. 2. Sequence of SOME/IP-SD

2-3 Threats to SOME/IP-SD protocol

Since the SOME/IP-SD protocol is a communication protocol that requires dynamic connection establishment as described above, the following attack methods are assumed.

(Attack Method 1) Eavesdropping by an unauthorized client

If an attacker is connected to the network, the attacker can detect which node is executing each service by eavesdropping the information communicated by the SOME/IP-SD protocol. For this reason, it is considered necessary to have a means of ensuring confidentiality for the SOME/IP-SD protocol.

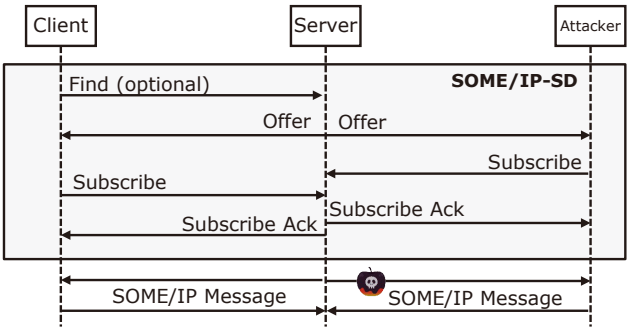


Fig. 3. Eavesdropping in SOME/IP-SD

(Attack method 2) Man-in-the-middle attack

Since in the SOME/IP-SD protocol, arbitrary messages can be sent without prior authentication, an attacker connecting to the network may inject data or commands and exploit them. An example of a specific attack method that takes advantage of this vulnerability is a man-in-the-middle attack, as shown in Fig. 4. By rewriting the Offer Service sent from a legitimate server to the information of the attacker's node and sending it, it is possible to connect the client to the server controlled by the attacker and send abnormal data.

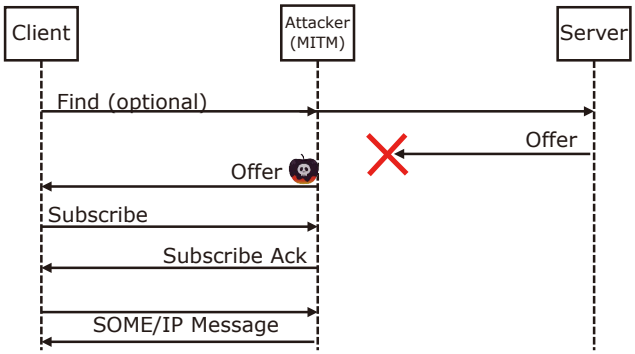


Fig. 4. Man-in-the-middle attacks in SOME/IP-SD

(Attack method 3) Copycat attack

In the paper (3), an attack method called a "copycat attack" is mentioned. In a copycat attack, an attacker waits for a legitimate server to send an Offer Service, and immediately after it is sent, the attacker sends their own Offer Service containing their own endpoint information. This attack method takes advantage of the client's characteristics to execute the Offer Service sent later and directs the client to the attacker's fake server. This attack is achieved by a combination of the aforementioned (Attack method 1) and (Attack method 2). The attacker can send abnormal data to the client by eavesdropping on the Offer Service from a legitimate server and impersonating the fake server.

2-4 Countermeasures against the threats and related studies

The following countermeasures are being discussed to eliminate threats to the SOME/IP protocols.

(Countermeasure 1) Protection by Message Authentication Code

AUTOSAR has prescribed specifications called "Secure Onboard Communication" (SecOC),[4] which has already been used in conventional in-vehicle control networks. Regarding the in-vehicle Ethernet, too, a method of protecting each message by assigning a Message Authentication Code (MAC) to it, which is made possible by adapting the in-vehicle Ethernet to SecOC, is being considered. However, this method is based on the assumption that a pre-written symmetric key is kept in a safe place, and thus it is not effective if this symmetric key is leaked. In other words, although it has the advantage of having the lowest communication overhead, the problem is that it is ineffective if the symmetric key is leaked.

(Countermeasure 2) Protection by TLS

To establish secure communication between nodes, consumer devices often use TLS*1 for TCP and DTLS*2 for UDP. However, since TLS imposes commensurate computational processing on both the server and the client, the computational complexity may be too large for embedded devices with limited computing resources. Furthermore, in applications used in automobiles, where

high real-time performance is required, the need to establish a connection with TLS in addition to executing SOME/IP-SD poses a problem, because it takes too long for each application to start.

In addition, the following countermeasures have been discussed as existing studies.

Fukuda et al. proposed a method in which the session management server authenticates messages on behalf of an in-vehicle control system by installing a session management server.[5] This method centralizes session management and makes it easy to manage these logs. However, if the session management server does not operate, all communication may not be established, and it takes time to establish communication, which is a problem.

Iorio et al. proposed an authentication protocol to protect SOME/IP messages.[6] This method is based on the assumption that handshaking takes place through unicast between all clients and servers using client and server certificates, and Find messages are not broadcast (or multicast). For this reason, the problem is that the number of messages increases as the number of services and nodes increases.

Zelle et al. conducted a security analysis of the SOME/IP protocol and proposed an improved protocol.[3] This method is achieved by sharing the session key (symmetric key) that is generated by the Offer Service notifier to the Offer Service after startup and before the session starts. This method makes it possible to protect the Offer Service by using a group key. However, because communication cannot be started until the session key is shared at each startup, and because the Find Service from the client is not supported, unless the frequency at which the server periodically sends the Offer Service is increased, connection initiation is delayed, posing a problem.

In this way, although various methods have been proposed, no method that is fully compatible with the SOME/IP-SD protocol and enhances security strength has been proposed. Therefore, in this paper, we propose a communication method that is compatible with the SOME/IP-SD protocol and enhances security while reducing the communication overhead.

3. Proposed Method

This chapter presents a protocol for establishing a secure communication channel between nodes for SOME/IP. Our proposed method focuses on SOME/IP-SD, which is executed when the communication between the server and the client is established, to protect communication between the server and the client. The following sections describe the requirements and outline of the designed protocol.

3-1 Requirements

There are several requirements unique to in-vehicle control systems.

(Requirement 1) Lower communication overhead

Since an in-vehicle control system is a distributed control system, communication at startup is crowded. For this reason, it is important to reduce the communication overhead at startup. There is concern that it would require a long time to establish communication if a large number of communications were carried out to ensure security. Therefore, a method that reduces the communication overhead as much as possible is required.

(Requirement 2) Protection of Find Service

After starting up, the client waits for a response from the server by executing the Find Service through multicast. The time required to establish communication can likely be shortened by using the Find Service rather than waiting to receive the Offer Service from the server. For this reason, a protection method that assumes the execution of the Find Service through multicast is required.

(Requirement 3) Ability to set security level

Conventional in-vehicle control systems do not encrypt all communication between nodes. For this reason, it is considered that communication can be classified into communication that needs to be encrypted and communication that does not need to be encrypted. Therefore, it is considered necessary to have a mechanism that allows each service to define its own security level and to change whether or not encryption is used and which encryption method is used.

3-2 Designing secure communication channels

There are several challenges when applying TLS or DTLS to SOME/IP. First, in SOME/IP handshaking, the client must discover through multicast where the server that executes the service from the client is located. During this process, the client distributes service group information, including the service ID, without encryption.

For this reason, there is concern that configuration information on the system may be leaked by eavesdropping as described in Attack method 1. In addition, it has become possible for a fake message to be sent to the client using the eavesdropping service group information by a man-in-the-middle attack as described in Attack method 2, which may lead to a copycat attack as described in Attack method 3.

Another method is to use MAC to secure SOME/IP messages by having each node have a static secret key (symmetric key). When using this method, although it is possible to protect against tampering with SOME/IP messages, the contents of the service may be leaked. In addition, as mentioned above, the method of storing the symmetric key statically cannot ensure security after leakage. For this reason, it is also necessary to authenticate and authorize the sender by using a server certificate or a client certificate.

Furthermore, multicast messages must be protected in addition to unicast messages, in order to achieve full compatibility with SOME/IP protocol. To protect messages belonging to a specific service instance, a pre-shared symmetric key is used. This single symmetric key is assumed to be shared in a secure location on all nodes before the session is established. Using this pre-shared symmetric key to protect the Find Service increases security and reduces the time required to establish the service compared to existing methods.

Based on the above, we propose a method that uses a pre-shared symmetric key, a server certificate, and a client certificate. The feature of this proposed protocol is that it is fully compatible with the conventional SOME/IP protocol and can protect the Find and Offer Services. In the following sections, after the key management in the proposed method is explained, the proposed protocol is explained.

3-3 Key management

The proposed method is based on the assumption that a symmetric key, and a server certificate or client certificate, are distributed to each node in advance. Therefore, this initial key needs to be written in a secure location, such as a factory. Then, when the SOME/IP-SD protocol is executed, the symmetric key is used to protect the Find Service. Also, mutual authentication using certificates is aimed at protecting the Find Service against server and client spoofing. Therefore, as shown in Fig. 5, each node should be assigned either a client or a server role, and each node should hold the private key and the corresponding certificate (server certificate or client certificate).

In addition, it is assumed that the symmetric key is updated at each node after the connection is established. For this reason, the symmetric key should be updated every time a connection is established in SOME/IP-SD and stored in nonvolatile memory.

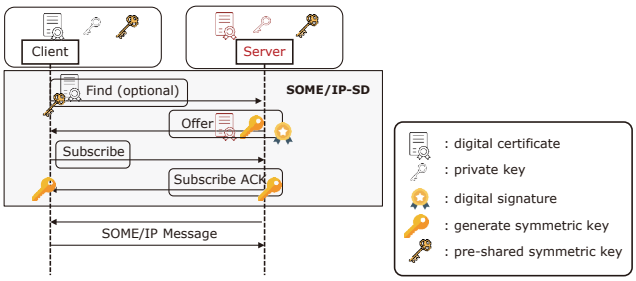


Fig. 5. Outline of key management and sequence in the proposed method

3-4 Proposed protocol

This proposed protocol basically adopts hybrid cryptography and is compatible with the SOME/IP-SD protocol. However, it differs significantly from other protocols in that it uses a pre-shared key to protect the Find Service. More specifically, the proposed protocol has the following features.

As shown in Fig. 5, when sending a Find Service, the client should first attach the client certificate to the Find Service, encrypt it with the pre-shared key, and then assign a MAC to protect it from eavesdropping or tampering. The server receiving the Find Service should verify and decrypt the MAC using the pre-shared key, and confirm that it is a request from an authorized client by verifying whether the client certificate was issued by a trusted root.

Next, before notifying the requesting client of an Offer Service, the server should generate a session key (symmetric key) after confirming that the service can provide the service requested from the client according to the list given in advance. After encrypting this session key using the public key attached to the client certificate, the server should respond to the client through the Offer Service. This procedure ensures that only the client that owns the corresponding private key can decrypt the service, thus ensuring the confidentiality of the session key.

Furthermore, in the Offer Service, a digital signature should be attached to the entire response in order to ensure its authenticity and integrity. Next, when receiving the Offer Service, the client should verify the server certificate sent to it to confirm its authenticity. Finally, the client should check the validity of the digital signature using the public key extracted from the server certificate. After confirming that the digital signature has not been tampered with, the client should decrypt the session key using its own private key and use it in subsequent communications. The Subscribe Eventgroup and Subscribe Eventgroup Ack should be sent encrypted with the session key to ensure confidentiality.

3-5 Service protection levels

When a normal SOME/IP message is sent after a session is established through the SOME/IP-SD protocol, it should be possible to set a protection level on a per-service basis. The protection level should basically be achieved by describing the applicable protection level in the client certificate; however, it should also be assumed that it is controlled by a pre-established list maintained by the server. For this reason, the server should be allowed to change the protection level when communication is unstable due to an attack, etc. An example of service protection levels is shown in Table 1.

Table 1. Example of service protection levels

Protection level	Service protection measure
0	None
1	Tamper-proof by message authentication codes
2	Encryption and tamper-proof
3	Encryption and digital signature protection

4. Evaluation

In this paper, we evaluated from two perspectives. First, we describe the measurement results of the communication overhead using an actual machine. Next, we describe the results of comparison with the countermeasures mentioned above and related studies.

4-1 Evaluation environment

The evaluation environment was made up with two Raspberry Pi devices and one switching hub, as shown in Fig. 6. The protocol was implemented using vsomeip.[7]

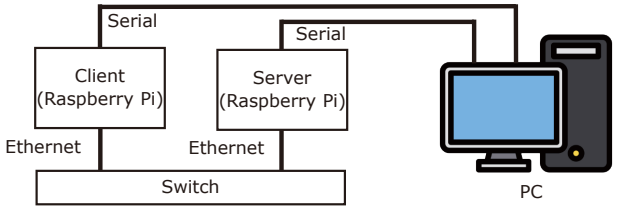


Fig. 6. Configuration of evaluation environment

4-2 Evaluation results 1: Evaluation of communication overhead

As an evaluation, we measured the throughput on the actual machine, and considered the communication overhead. We assumed that a comparison was made with Countermeasure 1, "Protection by Message Authentication Code," and Countermeasure 2, "Protection by TLS." The measurement time was determined to be the time between when each client sends the Find Service and when it receives a response through the Offer Service. Note that in Countermeasure 1 (MAC), there is no procedure after the Find Service and the Offer Service, and in Countermeasure 2 (TLS), since it is necessary to carry out the TLS communication protection procedure in advance, the communication overhead increases. Assuming these matters, we measured the time 100 times for each method and provided the average, minimum, and maximum values.

The measurement results in Table 2 indicate that the proposed protocol can be executed with sufficiently low communication overhead compared to conventional methods.

Table 2. Comparison of SOME/IP-SD session establishment times (ms)

		Countermeasure 1 (MAC)	Countermeasure 2 (TLS)	Proposed method
Measurement time	Minimum	25.990	95.784	45.164
	Average	26.778	95.983	46.552
	Maximum	28.206	99.245	48.655

4-3 Evaluation results 2: Comparison with existing countermeasures and related studies

First, we compared the proposed method with two existing countermeasures, namely Countermeasure 1 (MAC) and Countermeasure 2 (TLS). The comparison results are shown in Table 3. Regarding the client authentication (perspective 1) and the server authentication (perspective 2), Countermeasure 1 (MAC) has the disadvantage that these perspectives are not considered. Regarding the impact of the leakage of keys on other nodes (perspective 3), Countermeasure 1 (MAC) has a relatively low-security level because an attacker can impersonate a node using a leaked key. Regarding the session establishment time (perspective 4), Countermeasure 2 (TLS) has the disadvantage that its establishment time is relatively longer than those in the evaluation results described above. Based on these results, it is concluded that the proposed method has significant advantages over the other countermeasures.

Table 3. Comparison between countermeasures 1 and 2 and the proposed method

Perspective	Countermeasure 1 (MAC)	Countermeasure 2 (TLS)	Proposed method
1. Client authentication	- (without)	+ (with)	+ (with)
2. Server authentication	- (without)	+ (with)	+ (with)
3. Leakage of keys on other nodes	- (affected)	+ (not affected)	+ (not affected)
4. Session establishment time	++ (short)	- (long)	+ (relatively short)

Next, as a comparison with other related studies, we compared the proposed method with the methods described in paper (6) and paper (3). As shown in Table 4, both methods are assumed to use client and server certificates, and there is no significant difference between perspectives 1, 2, and 3. Regarding perspective 4, however, there is a difference in the handling of the Find Service. Specifically, the method described in paper (6) cannot broadcast the Find message, so the client needs to unicast the Find Service to the nodes that may execute the service. During this process, it is necessary to establish sessions with all candidate servers, and if there are many candidate servers, the communication volume becomes huge. Furthermore, in the method described in paper (3), the Find Service is considered not to be used without protection. In this case, communication can be initiated only by the server through the Offer Service, and the frequency of sending the Offer Service by the server may increase communication overhead. For these reasons, we consider that, in the proposed method, a session can be established in a relatively short time while supporting the Find Service. Therefore, we consider that the proposed method provides an optimal balance between security and real-time performance.

Finally, we explain perspectives 5 and 6 in Table 4 to highlight differences from the existing methods. First, regarding the protection of the Find Service (perspective 5), the method described in paper (6) uses unicast for protecting the Find Service, while the method described in paper (3) does not use the Find Service at all. Although the Find Service is optional under the SOME/IP protocol specifications, it is important to protect the Find Service because an attack on the Find Service would lead to leakage of the service group information required by the client. Additionally, since the Find Service is essential as a starting point for communication connection, it should be protected. On the other hand, the protection method after the Offer Service is almost the same, and it is considered that there is no difference between the three methods. Next, regarding the change in protection level (perspective 6), no other paper mentions this point, and it is assumed that protection is provided at a pre-defined service protection level with any method. However, it is considered effective to change the protection level for each SOME/IP service or dynamically adjust it based on the client's state or the communication channel.

Table 4. Comparison with existing studies

Perspective	Paper (6)	Paper (3)	Proposed method
1. Client authentication	+ (with)	+ (with)	+ (with)
2. Server authentication	+ (with)	+ (with)	+ (with)
3. Leakage of keys on other nodes	+ (not affected)	+ (not affected)	+ (not affected)
4. Session establishment time	-- (long)	- (relatively long)	+ (relatively short)
5. Protection of Find Service	+ (Unicast)	- (without)	++ (with)
6. Change in protection level	- (without)	- (without)	+ (with)

Based on these results, we conclude that the method proposed in this paper is effective in protecting SOME/IP-SD while maintaining a balance between security and real-time performance.

5. Consideration

In addition to the protection measures for the SOME/IP-SD protocol mentioned in this paper, there are several other countermeasures. For example, protection measures using IPSec and MACSec could be considered. The use of IPSec and MACSec would require switches that meet automotive requirements, and its application to in-vehicle control systems with strict cost constraints may be achieved in the future. However, these measures could be implemented safely with additional costs. Therefore, the challenge for the future would be comparing this proposed method with other protective measures not discussed in this paper.

6. Conclusion

In this paper, we propose an extension protocol for SOME/IP-SD to achieve a good balance between security and real-time performance. This protocol provides appropriate protection for the sequence of Find and Offer Services, and reduces the communication overhead compared to existing methods. Furthermore, we demonstrate that various protection measures can be dynamically adjusted by introducing service protection levels to protect communications.

In the future, we plan to discuss the application of this protocol in actual vehicles and compare it with other protective measures.

• Raspberry Pi is a trademark or registered trademark of The Raspberry Pi Foundation.

Technical Terms

∗1 Transport layer security (TLS): A protocol for communication requiring security in computer networks, such as the Internet, that provides authentication, encryption, and tampering detection functions. It is often used between a connection-oriented transport layer protocol (usually TCP) and the application layer.

∗2 Datagram transport layer security (DTLS): A protocol based on TLS, designed to provide the same security as TLS when the transport layer is UDP.

References

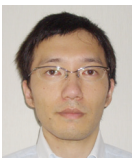
(1) SOME/IP Protocol Specification, AUTOSAR FO R22-11 (2022)
(2) SOME/IP Service Discovery Protocol Specification, AUTOSAR FO R22-11 (2022)
(3) D. Zelle, et al., Analyzing and Securing SOME/IP Automotive Services with Formal and Practical Methods, The 16th International Conference on Availability, Reliability and Security, Vienna, Austria (2021)
(4) Specification of Secure Onboard Communication Protocol, AUTOSAR FO R20-11 (2020)
(5) K. Fukuda, et al., A study on session management server for SOME/IP cybersecurity, ETNET2021, pp.1-8 (2021)
(6) M. Iorio, et al., Protecting In-Vehicle Service : Security-Enabled SOME/IP Middleware, IEEE Vehicular Technology Magazine, Vol. 15, No.3, pp.77-85 (2020)
(7) vsomeip, https://github.com/COVESA/vsomeip

Contributors

The lead author is indicated by an asterisk (∗).

H. UEDA*
• Manager, AutoNetworks Technologies, Ltd.

R. KURACHI*
• Ph.D.
Designated Assistant Professor, Nagoya University



H. TAKADA
• Ph.D.
Executive Director, Professor, Nagoya University

M. INOUE
• Senior Assistant general manager, AutoNetworks Technologies, Ltd.

S. KAMIGUCHI
• AutoNetworks Technologies, Ltd.

K. WADA
• AutoNetworks Technologies, Ltd.