MEDIS-DC STANDARDS

for Integrated Secure Communication Layer Protocols

Frozen Draft

V 1.00

1998.9.2

The Medical Information System Development Center

Contents

1. SCOPE

These standards are established as the communication protocol that is to be applied mainly to use secret keys for reserving a secure communication layer above the transport layer when performing interhospital communications using a regional medical collaboration system or intrahospital online electronic data saving operations. More specifically, these standards define conventions for communication between the transport layer and the secure layer, and for communication with the secure key storage management system to be applied when using something like an IC card as the operation card. The placement of these secure communication standards in the hierarchical structure required is shown in Figure 1.

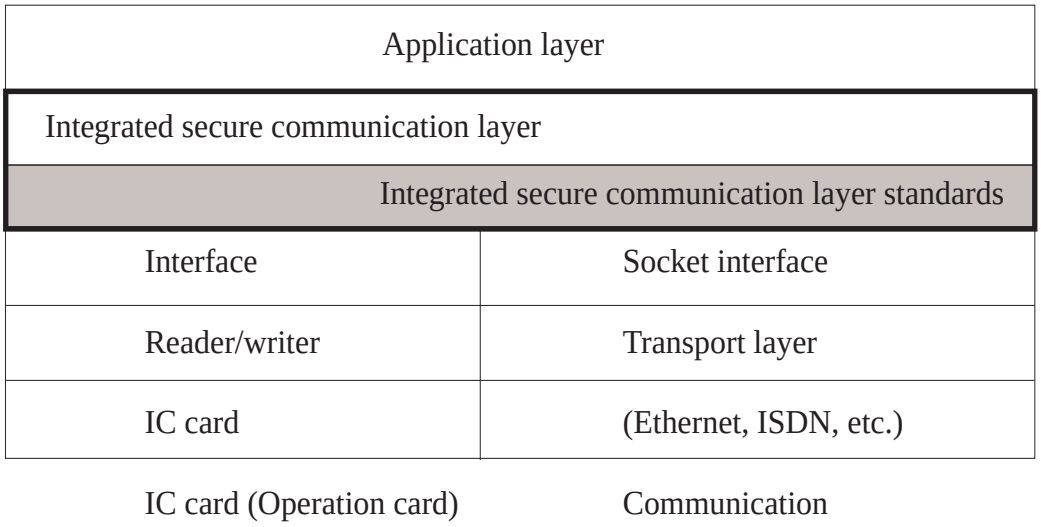

Fig.1 Placement of the integrated secure communication layer standards

2. RELEVANT CODES AND STANDARDS

ISO 7816-4: Identification cards - Integrated circuit(s) cards with contacts Part 4: Interindustry commands for interchange

JIS X 6306: IC cards with External Terminals - Common Commands

3. DEFINITIONS OF TERMS

(1) Integrated Secure Communication Layer: A protocol layer that allows secure communication as directed by the application layer, pursuant to the integrated secure communication layer standards. The abbreviation is "ISCL".

(2) Challenge Code: A code that is obtained when a GET CHALLENGE command is sent to the IC

card. During communication, this code is transmitted to the other entity as the seed for its authentication. After it is received by the other entity, the challenge code is further enciphered using the internal authentication key of the IC card and then returned to the sender. If the challenge code is enciphered correctly, the other entity is authenticated as the correct entity for the communication.

(3) Response code: The authentication code that the other entity returns when it receives the challenge code.

(4) Mutual authentication: Both entities in the communication check one another to confirm that the other entity is the correct entity for the communication.

(5) Communication block size: Maximum size of a data unit which is exchanged between ISCL layer and lower layer. Message block size is under this size. Maximum size which is set to lower layer is this communication size plus 32 bytes.

(6) Message: Total data which are sent in each session.

(7) Split message: Data which divided a message into communication block.

(8) Message block: Unit data which is exchange in communication protocols. The message block consists of a message header and a message data.

(9) Message header: The first part of message block which defines the purpose of the block.

(10) Message data: The latter part of message block which denotes the contents with purpose of the block.

4. BASIC FUNCTIONS

4.1 Outline

There are three types of threats to communication security: "masquerading", "tampering", and "eavesdropping". To ensure security, mutual authentication is provided as a countermeasure against "wearing a disguise", encryption against "eavesdropping", and message authentication against "falsifying". For online electronic data saving or other operations that do not provide an opportunity to eavesdrop, communications can also be performed without using the encryption function.

4.2 Authentication Functions of the IC Card

As shown in Table 1, the authentication commands stipulated in ISO 7816-4 or JIS X 6306 are used as the mutual authentication function and encryption function of the IC card. It is also permitted for these functions to be implemented using media other than the IC card.

Tab.1 Authentication commands of the IC card

Command	Definition
GET CHALLENGE	This command requests output of the challenge code. This command must be used before EXTERNAL AUTHENTICATE is used.
INTERNAL AUTHENTICATE	This command requests card-based calculation and output of an response code using the challenge code received from connected equipment and the internal authentication key stored within the card. This command must be used to make external entity authenticate the validity of the card.
EXTERNAL AUTHENTICATE	This command requests authentication of the response code received from connected entity using the challenge code output from the card and the external authentication key stored within the card. Output of the results is also requested.. This command must be used to make the card authenticate the validity of the external entity.

4.3 Mutual Authentication

Mutual authentication is provided as a countermeasure against "wearing a disguise". Mutual authentication based on these communication standards employs the "challenge-response" method. The procedure is shown in Figure 2.

(1) The "challenge code" is sent to the other entity who requires authentication. The challenge code is a random number that can be obtained by sending the GET CHALLENGE command to the IC card.

(2) The other entity provides the required challenge code calculation using the INTERNAL AUTHENTICATE command and returns a "response code" as the result.

(3) Likewise, the sender provides the required challenge code calculation using the EXTERNAL AUTHENTICATE command. The sender also compares self-calculation results and the response code received from the other entity. If the self-calculation results and the response code match (i.e., both entities know the required calculation method), the other entity is verified as the authentic entity.

(4) The other entity also performs steps (1) to (3) above to complete mutual authentication.

This mutual authentication method is referred to as the four-way mutual authentication scheme. Under these communication standards, however, a three-pass four-way mutual authentication scheme is used, in which the challenge code for mutual authentication is transmitted together with the response code.

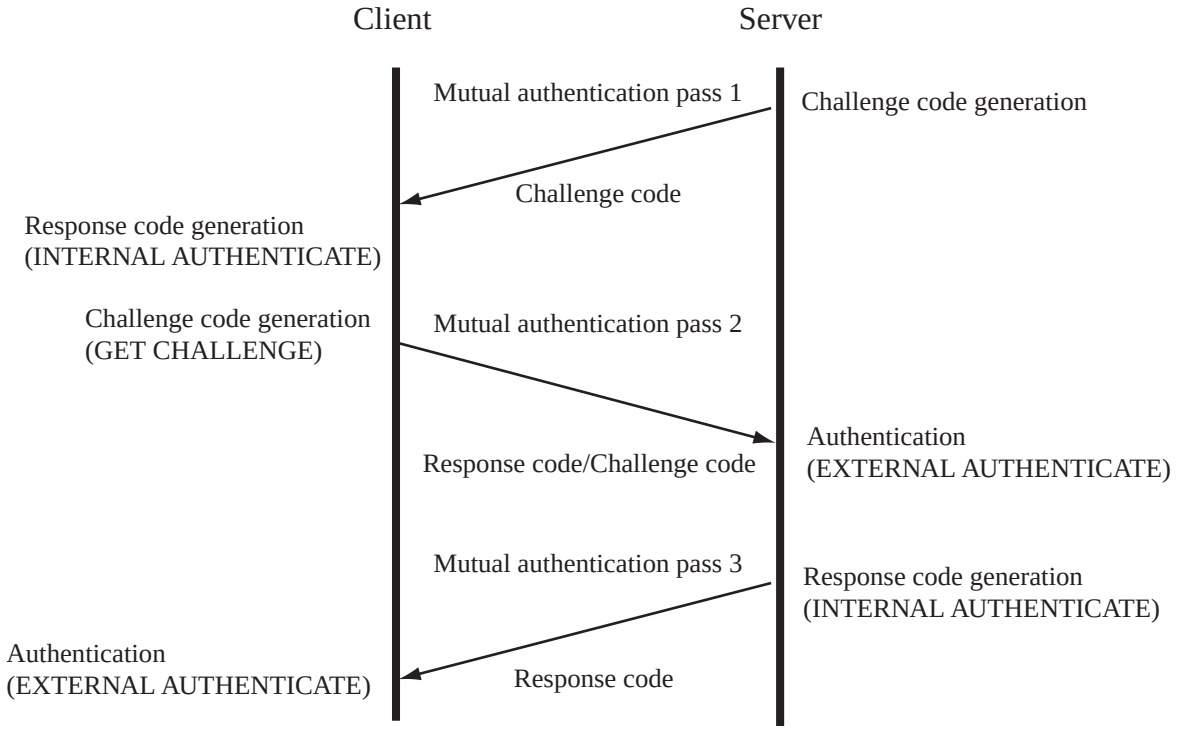

Fig.2 Three-pass four-way mutual authentication scheme

4.4 Encryption

Encryption is provided as a countermeasure against "eavesdropping" to ensure data security. The secret-key scheme and the DES-based block-type data encryption scheme (CBC scheme) is used as the encryption method. If the same encryption key is always used, there is a risk that the communication might be decoded. Under these communication standards, therefore, the encryption key is changed for each message send/receive session (the unit of message sending/receiving is arbitrary) in order to enhance security. That is, session keys are used. As shown in Figure 3, the session key for generating a random number generated using the GET CHALLENGE command is transmitted and then the response codes obtained by issuing the INTERNAL AUTHENTICATE command from each terminal are used as the session keys.. This reduces the risk that the session keys might be obtained by eavesdropping on the communication channel.

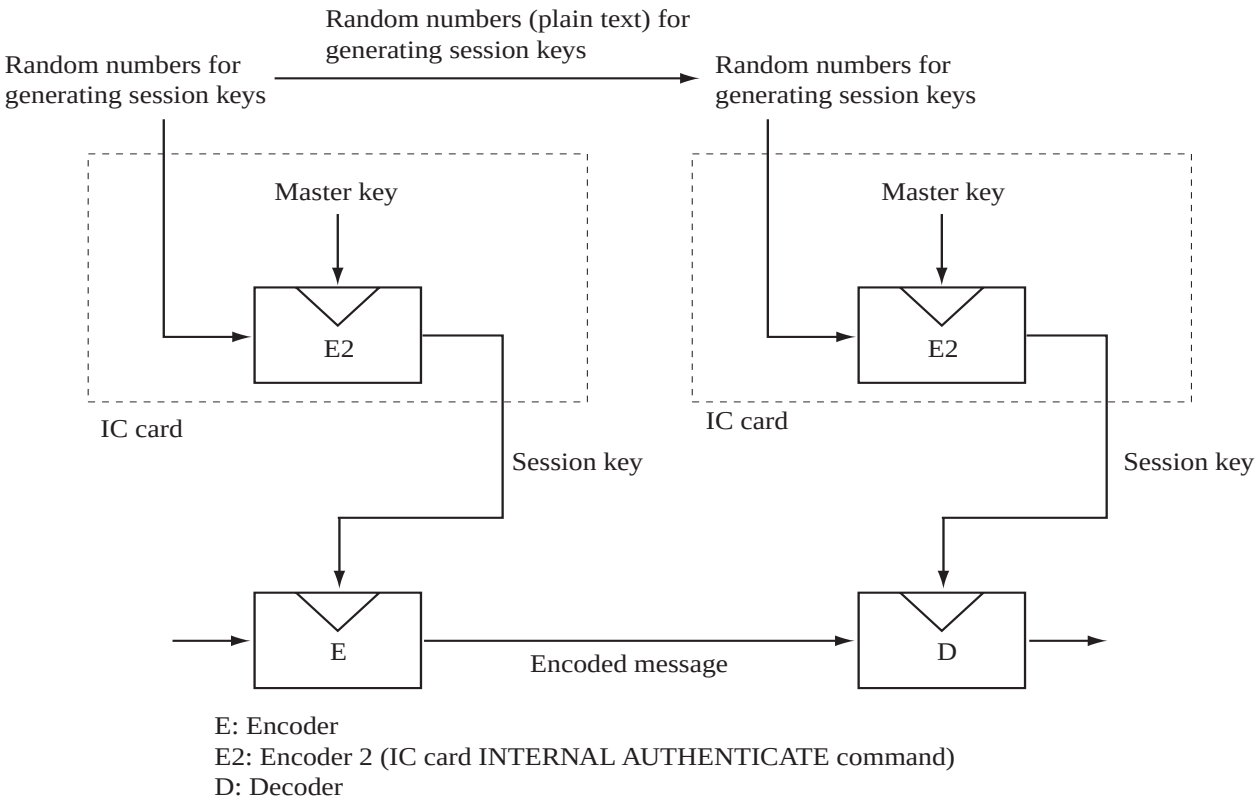

Fig.3 Generating and encoding session keys

4.5 Message Authentication

Message authentication is provided to detect falsification. Under these communication standards, the "message authenticators" that have been generated from messages are collated using a hash function to implement message authentication. A schematic diagram of message authentication is shown in Fig. 4.

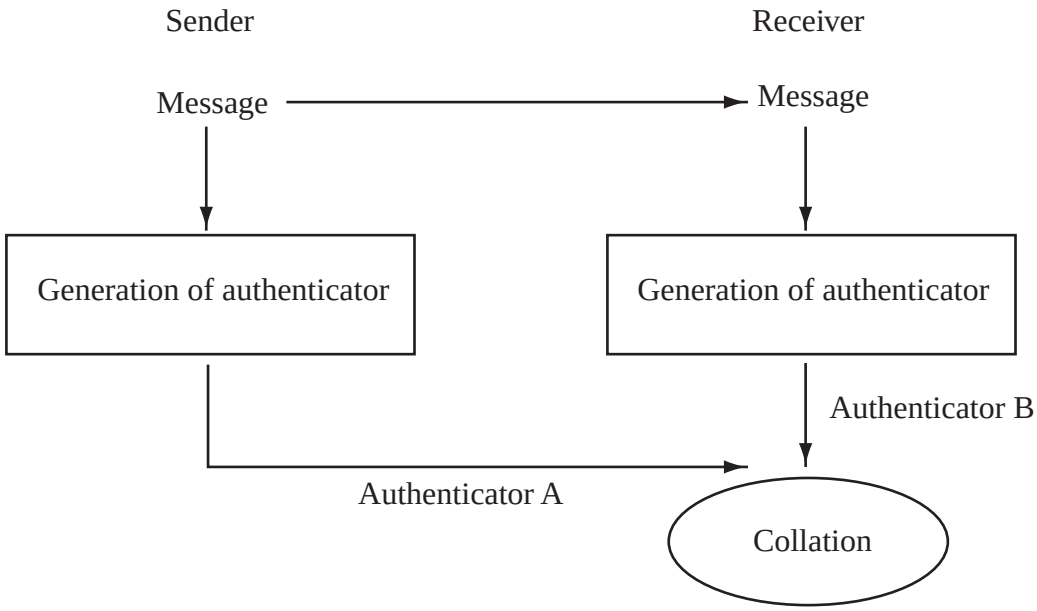

Fig.4 Message authentication with message authenticators

5. MESSAGE BLOCK

5.1 Basic Rules on the Message Block

5.1.1 Structure of the message block

The message block consists of a message header, which defines the purpose of the block, and message data, which denotes the contents of the block. In the remainder of this document, the message block is referred to as the MH, and the message data as the MD. The structure of the message block is shown in Fig. 5.

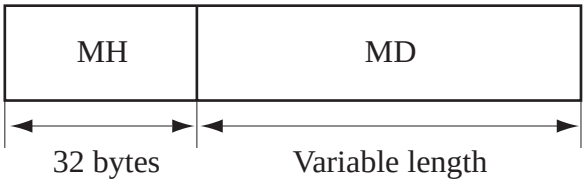

Fig.5 Structure of the message block

The MH has a fixed length of 32 bytes, and the MD is variable length data whose length is specified in the MH. The definition of the MH is given in Section 5.2.

5.1.2 Message block sending/receiving procedure

Under these standards, the message block must be sent/received using the following procedure:

(1) The sender sends the MH before sending the MD.
(2) The receiver assumes that the MH precedes the subsequent receiving processes.
(3) It is assumed that the communications may contain only an MH and not have an MD (when the data length of the MD is zero).

5.2 MH

5.2.1 Structure of the MH

The MH is information that defines the protocol specified under these standards. Table 2 lists components of the MH.

Tab.2 Components of the MH

Byte positions	Length	Field name	Field definition
1-4	4	Indicator	Indicator (Reserved area)
5-8	4	MessageID	Message identifier
9-12	4	dataLength	MD length
13-16	4	option	Option
17-20	4	time Stamp	Time (Reserved area)
21-24	4	errno	Reason code for negative response NAK (Reserved area)
25-32	8	stuff	Reserved area for adjusting the length of the MH to 32 bytes (Reserved area)

5.2.2 Indicator

The indicator is located at the beginning of the MH and has a specific bit pattern to denote the beginning of the MH. The indicator is not used under V1.00.

5.2.3 Message identifier

The message identifier is used to identify the type of MD. The message identifier consists of a "Purpose of Communication" section and a "Communication Type" section. As shown in Fig. 6, the first two bytes are the "Purpose of Communication" section and the last two bytes are the "Communication Type" section.

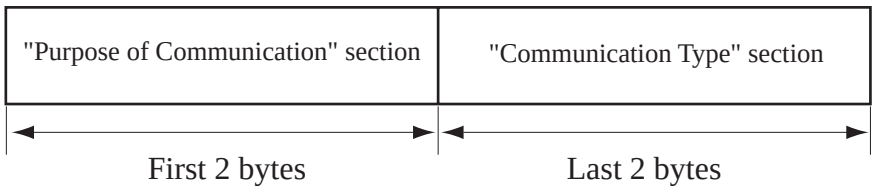

Fig. 6 Structure of the message identifier

(1) "Purpose of Communication" section
The codes assigned to the "Purpose of Communication" section are listed in Table 3.

Tab.3 Codes of the "Purpose of Communication" section

Purpose of communication	Code
Line connection check	0011H
Mutual authentication	0012H
Mutual authentication pass 1	0013H
Mutual authentication pass 2	0014H
Mutual authentication pass 3	0015H
End of mutual authentication	0016H
Message transmission	0020H
Session key for generating random number transmission	0021H
Message authenticator transmission	0023H
Through mode transmission	0026H
Line disconnection	00FFH

(2) "Communication Type" section
The codes assigned to the "Communication Type" section are listed in Table 4.

Tab.4 Codes of the "Communication Type" section

Type	Code
Request	0001H
Notification	0002H
Response	0003H

5.2.4 MD length

The data length of the MD following the MH is represented using a four-byte value.

5.2.5 Option

The options information belonging to the message block is stored in the "option" field.

(1) Result response to "Request"
The results of the response to a message whose "Communication Type" code is "Request" are stored in "option". Results response are listed in Table 5.

Tab.5 List of results response

Type	Results	Code	Remarks
Result response	ACK	00000000H	Positive
	NAK	FFFFFFFFH	Negative

(2) Processing results of the authentication command

Processing results of the authentication command are stored in "option". Processing results are listed in Table 6.

Tab.6 List of processing results

Type	Status	Code
Processing results	Normal end	00000000H
	Abnormal end	Undefined value other than the 'Normal end' code

5.2.6 Time

The time the data was transmitted is shown in the "timeStamp" field. This field is not used under V1.00.

5.2.7 Reason code

The reason code for the NAK result response is shown in the "errno" (error number) field. This field is not used under V1.00.

5.2.8 Stuff

A reserved area for adjusting the length of the MH to 32 bytes is shown in the "stuff" field. This field is not used under V1.00.

5.3 MD

5.3.1 MD list

The message data used under the Integrated Secure Communication Layer Protocols are listed in Table 7.

Tab.7 MD list

Data name	MD length (bytes)	Remarks
ISCL version code and communication block size	20	Used for line connection. See Section 6.2
Mutual authentication method data	Undefined	Used for mutual authentication. See Section 6.3.2
Authentication attributes	Undefined	Used for mutual authentication. See Section 6.3.2
Challenge code	8	Generated using the GET CHALLENGE command.
Response code	8	Generated using the INTERNAL AUTHENTICATE command.
Response code and challenge code	16	Response code and challenge code transmitted at the same time. See Section 6.3.4.
Transmission method data	16	Generated according to the transmission method specified from a higher-level application program.
Receivable message length	4	The maximum message length permissible for one exchange operation in accordance with the message sending/receiving protocol.
Session key for generating a random number	8	Generated using the GET CHALLENGE command.
Message authentication code	Undefined	Generated using the message authenticator generation function. The size depends on the algorithm.
Split message	Undefined	The message that has been delivered from a higher-level application program is split into sections falling within the communication block size defined in the line connection protocol.

5.3.2 MD details

(1) ISCL version code and communication block size The ISCL version code and the communication block size must have the structure shown in Fig. 7. The ISCL version code under these standards must be represented as "MEDIS-ISCL V1.00" (ASCII code), where denotes a space. The data size of the communication block which can be transmitted during one operation under the protocol is shown below.

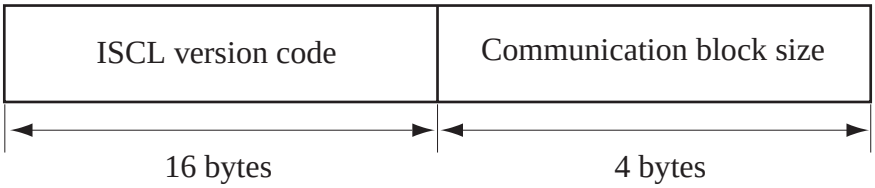

Fig. 7 Structure of the ISCL version code and communication block size

(2) Mutual authentication method data

When requesting mutual authentication, the sender must send mutual authentication method data to the receiver. The structure of the mutual authentication method data is shown in Fig. 8.

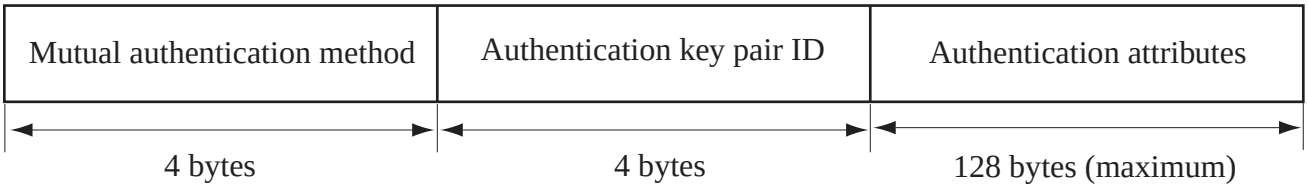

Fig. 8 Structure of the mutual authentication method data

Each of the data items above is described below.

1) Mutual authentication method

Mutual authentication methods are listed in Table 8.

Tab.8 List of mutual authentication methods

mutual authentication methods	Code	Remarks
4-way mutual authentication	00000000H	Default

2) Authentication key pair ID

The key pair IDs to be used for mutual authentication are listed in Table 9.

Tab.9 List of authentication key pair IDs

Name of the authentication key pair	ID
Authentication key pair 1	00000001H
Authentication key pair 2	00000002H
Authentication key pair 3	00000003H
Authentication key pair 4	00000004H
Authentication key pair 5	00000005H
Authentication key pair 6	00000006H
Authentication key pair 7	00000007H
Authentication key pair 8	00000008H

3) Authentication attributes

The authentication attributes are 1 to 128 bytes of attribute data to be used for authentication. Details of the data are not specified under these standards.

(3) Authentication attributes

The authentication attributes are 1 to 128 bytes of attribute data to be used for authentication. Details of the data are not specified under these standards.

(4) Challenge code

This code is an eight-byte code generated using the GET CHALLENGE command.

(5) Response code

This code is an eight-byte code generated using the INTERNAL AUTHENTICATE command.

(6) Response code and challenge code

The response code is an eight-byte code generated using the GET CHALLENGE command, and the challenge code is an eight-byte code generated using the INTERNAL AUTHENTICATE command. The structure of these codes is shown in Fig. 9.

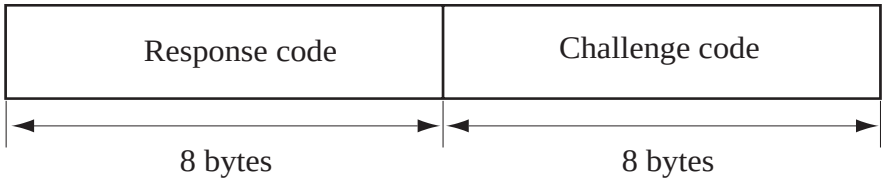

Fig. 9 Structure of the response code and challenge code

(7) Transmission method data

When requesting transmission, the sender must send transmission method data to the receiver. The structure of the transmission method data is shown in Fig. 10.

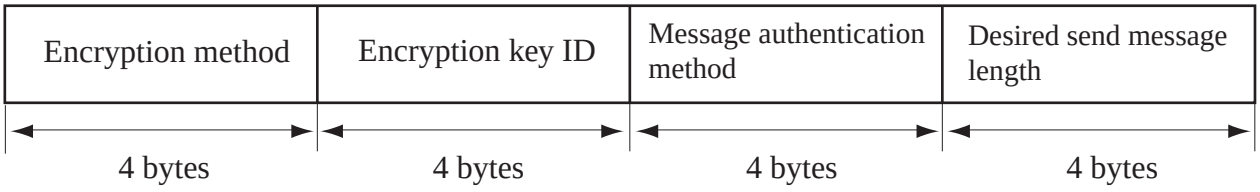

Fig. 10 Structure of the transmission method data

Each of the data items above is described below.

1) Encoding method

encryption methods are listed in Table 10.

Tab.10 List of encryption methods

encryption methods	Code
No encryption	00000000H
DES(CBC)	00001212H

2) Encryption key ID

The master key IDs to be used for encryption and decryption are listed in Table 11.

Tab.11 List of coding key IDs

Name of the coding key	ID
Encryption key 1	00000001H
Encryption key 2	00000002H
Encryption key 3	00000003H
Encryption key 4	00000004H
Encryption key 5	00000005H
Encryption key 6	00000006H
Encryption key 7	00000007H
Encryption key 8	00000008H

3) Message authentication method

Message authentication methods are listed in Table 12.

Tab.12 List of message authentication methods

Message authentication method	Code
No message authentication	00000000H
MD5	00001441H
MESMAC	00004001H

4) Desired send message length

This value refers to the length of the message that the user wishes to send at one time.

(8) Receivable message length

The desired send message length that has been specified in transmission method data by the sender or the maximum length of the message which the receiver can receive, whichever is smaller, must be the actually receivable message length (See Section 6.4). The message length are shown in 4-byte value.

(9) Random number for generating session key

This number is an eight-byte code generated using the GET CHALLENGE command.

(10) Message authentication code

The message authentication code is generated using the message authenticator generation function. The size of this authentication code changes according to the particular type of message authentication code generation algorithm.

(11) Split message

The transmission message that has been delivered from a higher-level application program is split into sections falling within the communication block size defined in the line connection protocol. The transmission message length or the communication block size, whichever is smaller, is used as the MD length (split message length).

5.4 Message Block List

A message block list is given in Table 13.

Tab.13 Message Block List

Message identifier		MH				MD	Remarks	
Purpose of communication	Type	Message identifier (code)	MD length (bytes)	Option			Direction of transmission	Others
				Species	Description			
Line connection check	Request	00110001H	20	-	-	ISCL version code and communication block size	C <-- S	
Line connection check	Response	00110003H	20/16	R	ACK/NAK	ISCL version code and communication block size	C --> S	For NAK, only the ISCL version code is sent with an MD length of 16 bytes
Mutual authentication	Request	00120001H	Undefined	-	-	Mutual authentication method data	C <-- S	The MD length ranges from 9 to 136 bytes
Mutual authentication	Response	00120003H	Undefined	R	ACK/NAK	Authentication attributes	C --> S	The MD length ranges from 1 to 128 bytes
Mutual authentication pass 1	Notification	00130002H	8/0	S	GET CHALLENGE command result code	Challenge code	C <--S	The MD length becomes 0 bytes if a GET CHALLENGE command error occurs
Mutual authentication pass 2	Notification	00140002H	16/0	S	INTERNAL AUTHENTICATE/GET CHALLENGE command result code	Response code or challenge code	C --> S	The MD length becomes 0 bytes if an INTERNAL AUTHENTICATE or GET CHALLENGE command error occurs
Mutual authentication pass 3	Notification	00150002H	8/0	S	EXTERNAL AUTHENTICATE/INTERNA-L AUTHENTICATE command result code	Response code	C <-- S	The MD length becomes 0 bytes if an EXTERNAL AUTHENTICATE or INTERNAL AUTHENTICATE command error occurs
End of mutual authentication	Notification	00160002H	0	S	EXTERNAL AUTHENTICATE command result code	-	C --> S	
Message transmission	Request	00200001H	16	-	-	Transmission method data	SND --> RCV	
Message transmission	Response	00200003H	4/0	R	ACK/NAK	Receivable data length	SND <-- RCV	The MD length is 0 bytes for NAK.
Transmission of a random number for generating session key	Request	00210001H	8/0	S	GET CHALLENGE/INTERNAL AUTHENTICATE command result code	Random numbers for generating session keys	SND --> RCV	The MD length becomes 0 bytes if a GET CHALLENG or INTERNAL AUTHENTICATE command error occurs
Transmission of a random number for generating session key	Response	00210003H	0	S	INTERNAL AUTHENTICATE command result code	-	SND <-- RCV	
Message transmission	Notification	00200002H	Undefined	-	-	Split message	SND --> RCV	Split into sections falling within the communication block size
Message authentication code transmission	Notification	00230002H	Undefined	-	-	Message authentication code	SND --> RCV	The length of the message authentication code depends on the method
Through mode transmission	Notification	00260002H	Undefined	-	-	Split message	SND --> RCV	Split into sections falling within the communication block size
Line disconnection	Request	00FF0001H	0	-	-	-	RQT --> RPS	
Line disconnection	Response	00FF0003H	0	R	ACK/NAK	-	RQT <-- RPS	

Notes:1. Option species ... R: Result response, S: processing results. 2. Direction of transmission ... C: client, S: server, SND: sending terminal, RCV: receiving terminal, RQT: requesting side, and RPS: responding side.

6. PROTOCOLS

The protocols under these standards can be broadly divided into four types:
(1) Line connection protocol
(2) Mutual authentication protocols
(3) Message sending/receiving protocols
(4) Line disconnection protocol

An outline of these protocols is given in Fig. 11.

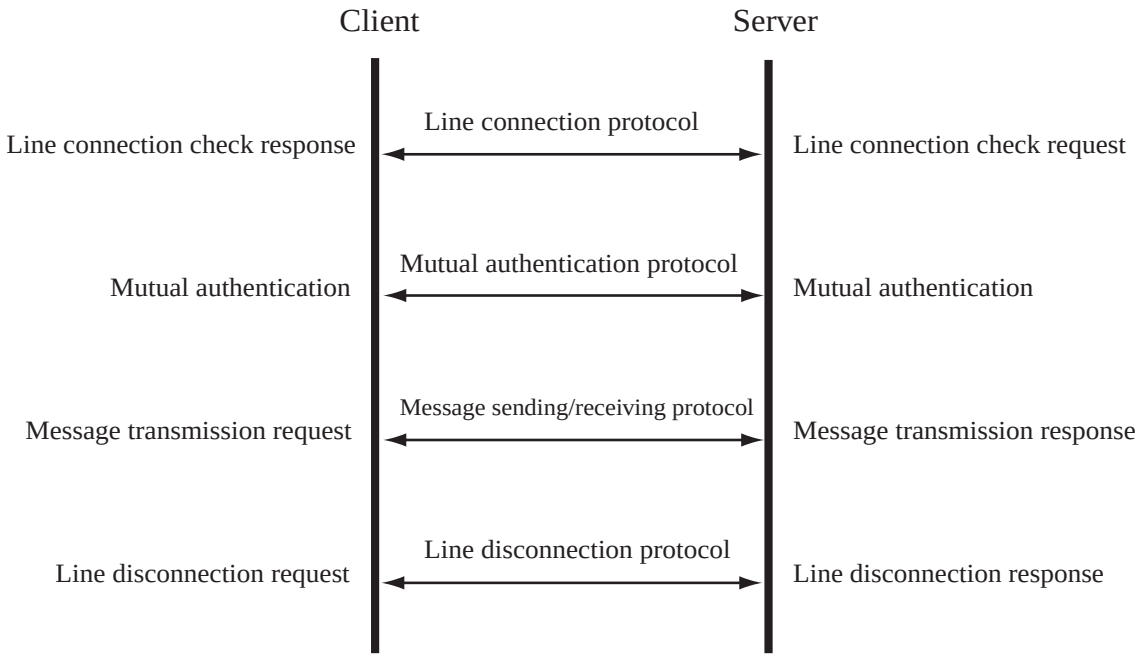

Fig. 11 Outline of the protocols

6.1 Protocol Definition Methods

The definitions of the terms used in this section are shown below.

MH Refers to the entire MH.
MH.msgId Refers to the "messageId" field of the MH.
MH.length Refers to the "dataLength" field of the MH.
MH.opt Refers to the "option" field of the MH.

6.2 Line Connection Protocol

6.2.1 Function

The line connection protocol is the first protocol in ISCL. This protocol is provided to check the ISCL version code and the communication block size.

6.2.2 Sequence

(1) The client issues a connection request.
(2) The server accepts or rejects the connection request.
(3) The server sends a line connection check request and the ISCL version code + communication block size.

"Line connection check request" MH

MH.msgID=Line connection check request MH.length=20 MH.opt=0

Note: The ISCL version code/communication block size must consist of the 16+4 bytes of data shown below.

ISCL version code ("MEDIS-ISCL V1.00") Communication block size (which the server can communicate)

(4) The client receives the line connection check request and the ISCL version code + communication block size. At this time, the sequence terminates abnormally in the following cases:
- The line connection check request is not set in "MH.msgId". (In this case, processing terminates without a line connection check response code being sent in return.)
- The ISCL version code is not correct.

(5) The server communication block size or the client communication block size, whichever is smaller, is used as the actual communication block size.

(6) The client sends a line connection check response and the ISCL version code + communication block size.

"Line connection check response" MH

MH.msgId=Line connection check response MH.length=20(or 16 for NAK) MH.opt=ACK(or NAK for an error)

Note: The data format of the ISCL version code/communication block size must be the same as that given in step (3) above.

(7) The server receives the line connection check response and the ISCL version code + communication block size. The sequence terminates abnormally in the following cases:
- The line connection check response is not set in "MH.msgId".
- The "MH.opt" code is not ACK.
- The communication block size is larger than that of the server.

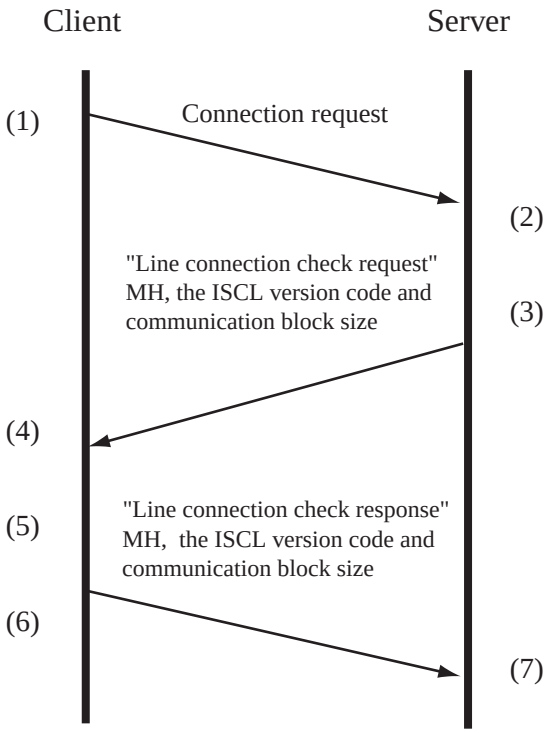

Fig. 12 Line connection protocol

6.3 Mutual Authentication Protocols

6.3.1 Outline of the mutual authentication protocols

An outline of the mutual authentication protocols is given in Fig. 13.

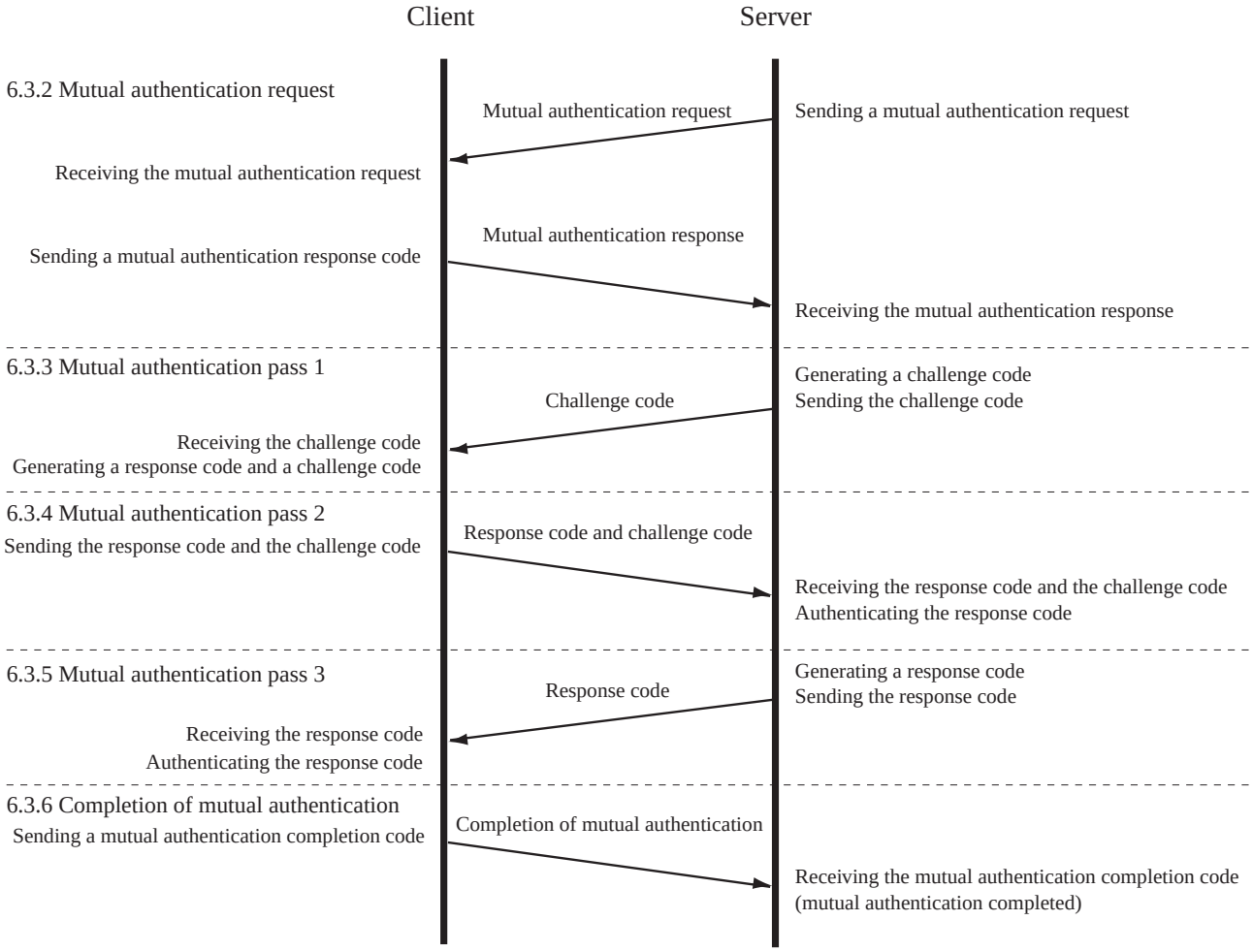

Fig. 13 Outline of the mutual authentication protocols

6.3.2 Mutual authentication request protocol

6.3.2.1 Function

The mutual authentication request protocol is the first protocol in mutual authentication. This protocol is provided to check the method of mutual authentication.

6.3.2.2 Sequence

(1) The server issues a mutual authentication request and mutual authentication method data.

"Mutual authentication request" MH

MH.msgId=Mutual authentication request MH.length=9-136 MH.opt=0

The mutual authentication method data must consist of the (4 x 2 + 1-128) bytes of data shown below.

Mutual authentication method Mutual key pair ID Authentication attributes

(2) The client receives the mutual authentication request and the mutual authentication method data. The sequence terminates abnormally in the following cases:
- The mutual authentication request is not set in "MH.msgId". (In this case, processing terminates without a mutual authentication response code being sent in return.)
- The mutual authentication method is not correct.

(3) The client sends a mutual authentication response code and authentication attributes.

"Mutual authentication response" MH

MH.msgId=Mutual authentication response MH.length=1-128 MH.opt=ACK if the mutual authentication method is correct, or NAK if the method is not correct.

The authentication attribute data must be 1 to 128 bytes long.

(4) The server receives the mutual authentication response. The sequence terminates abnormally in the following cases:
- The mutual authentication response is not set in "MH.msgId".
- The "MH.opt" code is not ACK.

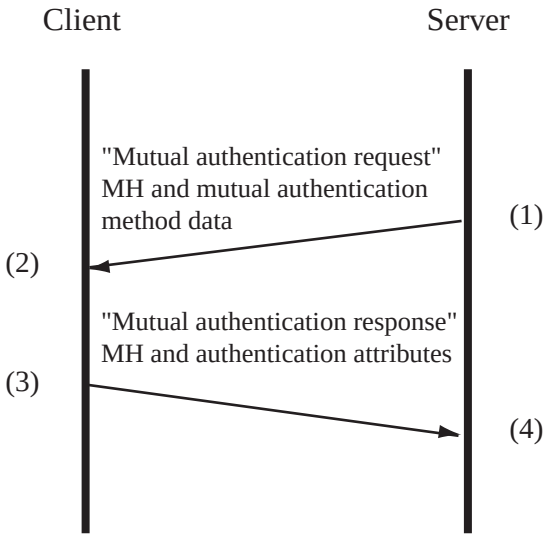

Fig. 14 Mutual authentication request protocol

6.3.2.3 Mutual authentication pass 1 protocol

6.3.2.4 Function

The mutual authentication pass 1 protocol is provided for the server to send a challenge code for the client to generate a response code and a challenge code.

6.3.2.5 Sequence

(1) The server generates a challenge code using the GET CHALLENGE command.
(2) The server sends a mutual authentication pass 1 notification code and the generated challenge code. If, however, the command mentioned in step (1) above is incorrect, the sequence terminates abnormally after transmission of the notification code.

"Mutual authentication pass 1 notification" MH

MH.msgId = Mutual authentication pass 1 notification MH.length = 8 (or 0 if the GET CHALLENGE command was incorrect) MH.opt = GET CHALLENGE command result code

Note: The challenge code must be eight bytes of data.

(3) The client receives the mutual authentication pass 1 notification code and the challenge code. The sequence terminates abnormally in the following cases:
- Mutual authentication pass 1 notification is not set in "MH.msgId".
- "MH.opt" is abnormal end (not zero).

(4) The client uses the received challenge code and generates a response code using the INTERNAL AUTHENTICATE command.
(5) The client generates a challenge code using the GET CHALLENGE command.

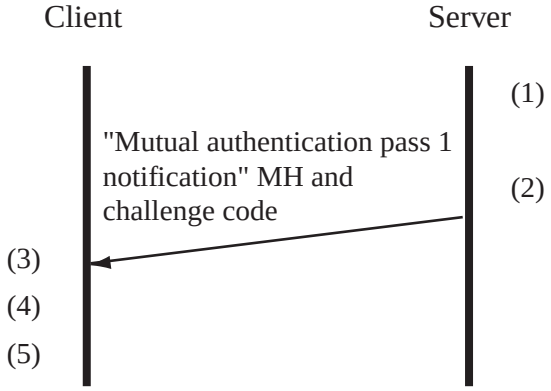

Fig. 15 Mutual authentication pass 1 protocol

6.3.3 Mutual authentication pass 2 protocol

6.3.3.1 Function

The mutual authentication pass 2 protocol is provided for the client to send both the response code it has generated from the previously received challenge code and a self-generated new challenge code, and for the server to authenticate the response code. See Sections 6.7.1 and 6.7.2 for a description of exceptional protocols.

6.3.3.2 Sequence

(1) The client sends a mutual authentication pass 2 notification code and the generated response code + challenge code.

"Mutual authentication pass 2 notification" MH

MH.msgId=Mutual authentication pass 2 notification MH.length=16 MH.opt=0

Note: The response code + challenge code refers to the data of (8 bytes x 2) shown below.

Response code Challenge code

(2) The server receives the mutual authentication pass 2 notification code and the response code + challenge code.
(3) The server authenticates the response code by issuing the EXTERNAL AUTHENTICATE command.

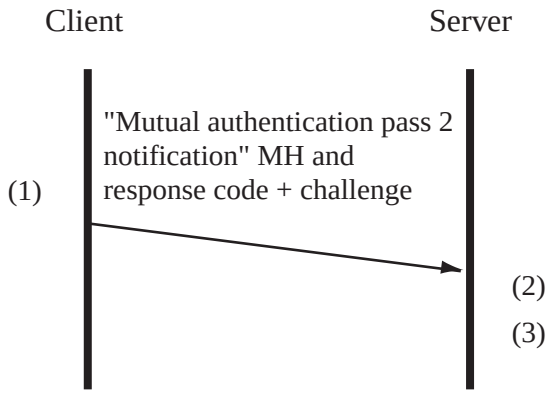

Fig. 16 Mutual authentication pass 2 protocol

6.3.4 Mutual authentication pass 3 protocol

6.3.4.1 Function

The mutual authentication pass 3 protocol is provided for the server to send the response code it has generated from the previously received challenge code, and for the client to authenticate the response code. See Sections 6.7.3 and 6.7.4 for a description of exceptional protocols.

6.3.4.2 Sequence

(1) The server generates a response code using the INTERNAL AUTHENTICATE command. (2) The server sends a mutual authentication pass 3 notification code and the generated response code.

"Mutual authentication pass 3 notification" MH

MH.msgId=Mutual authentication pass 3 notification MH.length=8 MH.opt=0

Note: The response code must be eight bytes of data.

(3) The client receives the mutual authentication pass 3 notification code and the response code. The sequence terminates abnormally in the following cases:
- Mutual authentication pass 3 notification is not set in "MH.msgId".
- "MH.opt" is not zero.

(4) The client authenticates the response code using the EXTERNAL AUTHENTICATE command.

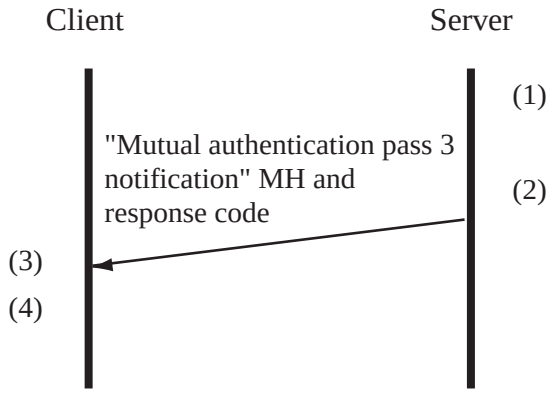

Fig. 17 Mutual authentication pass 3 protocol

6.3.5 Mutual authentication completion protocol

6.3.5.1 Function

The mutual authentication completion protocol is provided to complete mutual authentication by sending authentication results of the mutual authentication pass 3 protocol from the client to the server.

6.3.5.2 Sequence

(1) The client sends a mutual authentication completion notification code.

"Mutual authentication completion notification" MH

MH.msgId = Mutual authentication completion notification MH.length = 0 MH.opt = Authentication results (Result code for the EXTERNAL AUTHENTICATE command in the mutual authentication completion protocol.)

(2) The server receives the mutual authentication completion notification code. The sequence terminates abnormally in the following cases:
- Mutual authentication completion notification is not set in "MH.msgId".
- "MH.opt" is not zero.

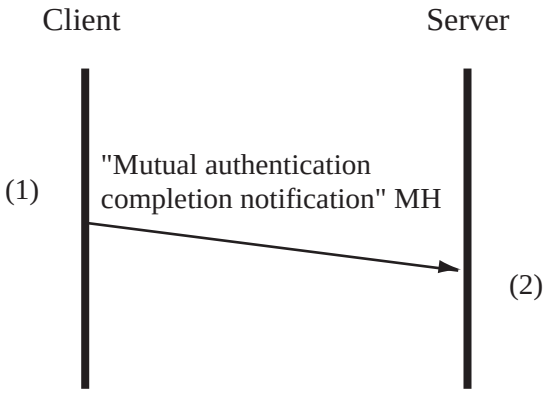

Fig. 18 Mutual authentication completion protocol

6.4 Message Sending/Receiving Protocols

6.4.1 Outline of message sending/receiving protocols

An outline of the message sending/receiving protocols is given in Fig. 19.

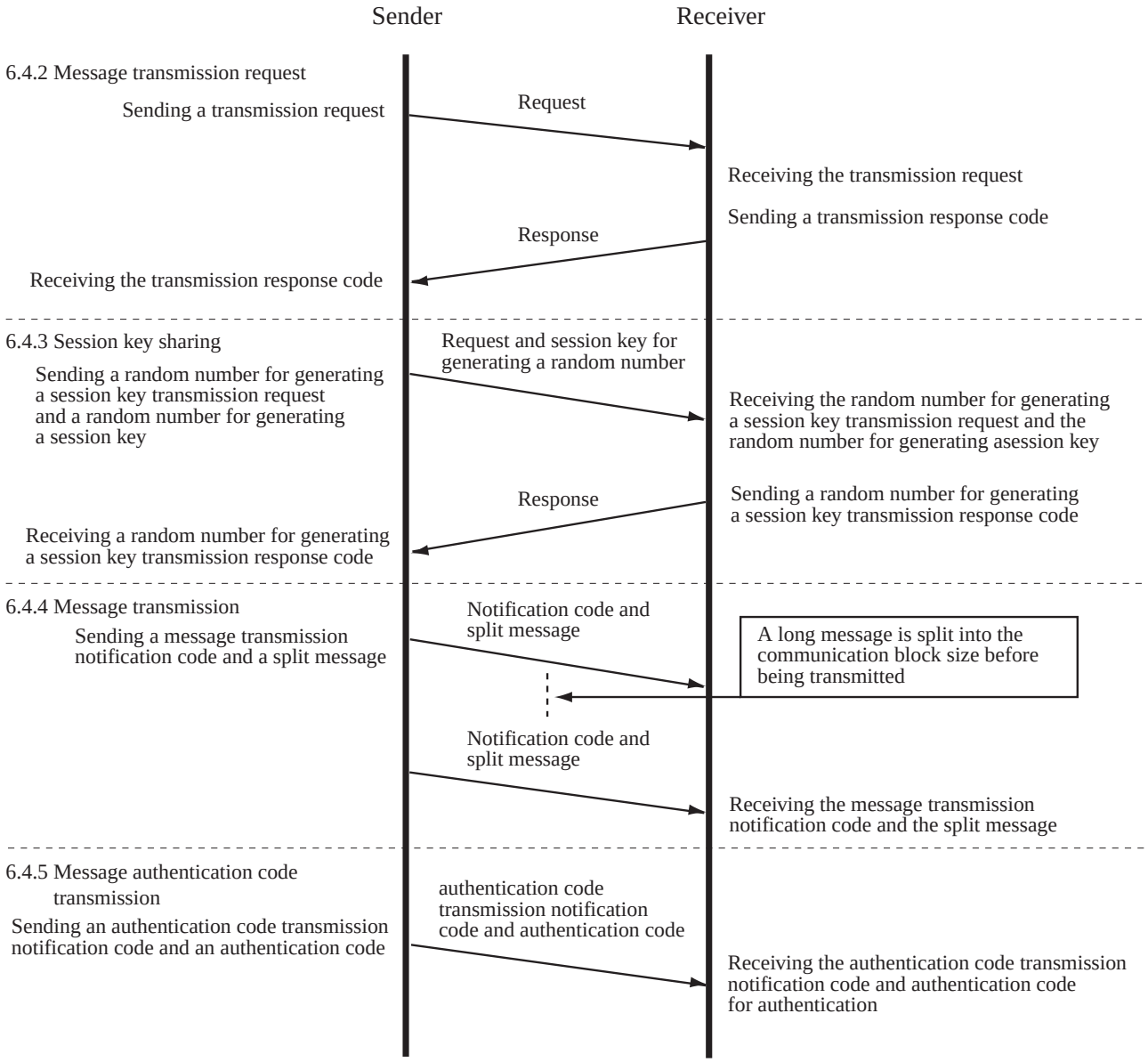

Fig. 19 Outline of message sending/receiving protocols

6.4.2 Message transmission request protocol

6.4.2.1 Function

The message transmission request protocol is provided for the sender to specify the transmission method and request the start of transmission, and for the receiver to respond by specifying a receivable message length. See Section 6.7.5 for exceptional protocols.

6.4.2.2 Sequence

(1) The sender generates a message transmission request and transmission method data.

"Message transmission request" MH

MH.msgId = Message transmission request MH.length = 16 MH.opt = 0

The length of the transmission method data must be (4 bytes x 4).

Encryption method Encryption key ID Message authentication method Desired send message length

(2) The sender sends the message transmission request and the transmission method data.

(3) The receiver receives the message transmission request and the transmission method data. The sequence terminates abnormally in the following cases:
- "MH.msgId" is not a message transmission request.
- "MH.length" is not 16.

(4) The receiver generates message transmission response MH and receivable message length data (4-byte area).

"Message transmission response" MH

MH.msgId = Message transmission response MH.length = 4 (or 0 for NAK) MH.opt = ACK if the encryption method and the MAC method match, or NAK if they do not match

The sender-generated message length data or the receiver-generated receivable message length data, whichever is smaller, is used as the actually receivable message length data.

(5) The receiver sends the message transmission response and the receivable message length data.

(6) The sender receives the message transmission response and the receivable message length data. The sequence terminates abnormally in the following cases:
- "MH.msgId" is not a message transmission response code.
- "MH.length" is not 4.
- "MH.opt" is not ACK.

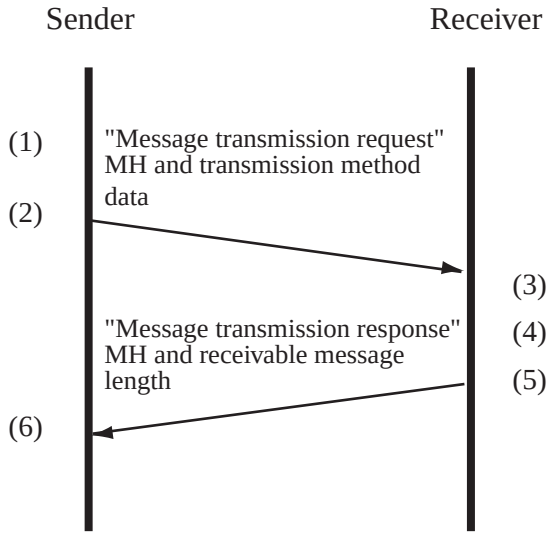

Fig. 20 Message transmission request protocol

6.4.3 Session key sharing protocol

6.4.3.1 Function

The session key sharing protocol occurs only when it is necessary to share session keys. See Section 6.7.6 for exceptional protocols.

6.4.3.2 Sequence

(1) The sender generates a random number for generating a session key using the GET CHALLENGE command and generates a session key from the random number using the INTERNAL AUTHENTICATE command.

(2) The sender sends a random number for generating a session key transmission request and the random number generated above for generating a session key .

"Random number for generating a aession key transmission request" MH

MH.msgId = Random number for generating a aession key transmission request MH.length = 8 MH.opt = GET CHALLENGE or INTERNAL AUTHENTICATE command result code (0 if normal, or other than 0 if abnormal)

(3) The receiver receives the random number for generating a aession key transmission request and the random number for generating a aession key . The sequence terminates abnormally in the following cases:

- "MH.msgId" is not a random number for generating a aession key transmission request.
- "MH.opt" is not 0.

(4) The receiver issues the INTERNAL AUTHENTICATE command and generates a session key from the received the random number for generating a aession key .

(5) The receiver sends a random number for generating a aession key transmission response.

"Random number for generating a aession key transmission response" MH

MH.msgId = Random number for generating a aession key transmission response MH.length = 0 MH.opt = INTERNAL AUTHENTICATE command result code (0 if normal, or other than 0 if abnormal)

Note: The sequence ends here if an error has occurred during generation of the session key.

(6) The sender receives the random number for generating a aession key transmission response. The sequence terminates abnormally in the following cases:
- "MH.msgId" is not a random number for generating a aession key transmission response.
- "MH.opt" is not 0.

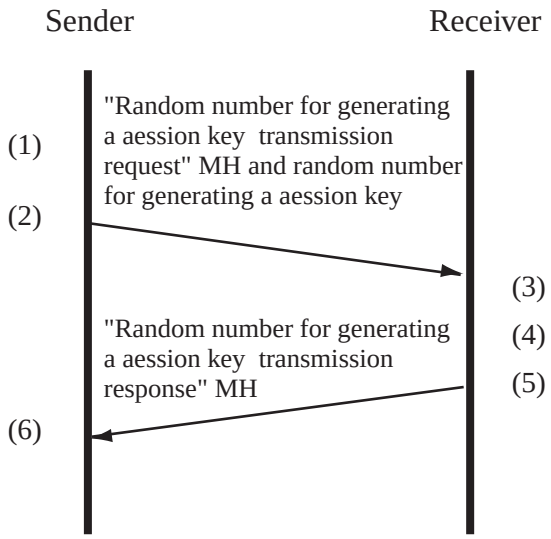

Fig. 21 Session key sharing protocol

6.4.4 Message transmission protocol

6.4.4.1 Function

The message transmission protocol is provided to transmit a message split into the communication block size determined through negotiations in the line connection protocol described in Section 6.2. If the encryption method is specified in the message transmission request protocol described in Section

6.4.2, the entire message is encrypted.

6.4.4.2 Sequence

(1) The sender takes as the correct message length the receivable message length that has been sent from the receiver in response to a message transmission request. See Section 6.4.2 for the receivable message length.

(2) The sender takes the residual message length or the communication block size, whichever is smaller, as the split message length.

(3) The sender takes out the split message for directed length from message. If the encryption method has been specified, that split message is encrypted. Encryption is performed for the entire message.

(4) The sender sends a message transmission notification code and the split message.

"Message transmission notification" MH

MH.msgId = Message transmission notification MH.length = Split message length MH.opt = 0

(5) The receiver receives the message transmission notification and the split message (corresponding block). If the encryption method has been specified, that split message is decrypted. The sequence terminates abnormally in the following cases:
- "MH.msgId" is not a message transmission notification.

(6) The sender repeats steps (2) to (4) above until processing of all message has been completed.

(7) The receiver repeats step (5) above until processing of all receivable message has been completed. See Section 6.4.2 for the receivable message length.

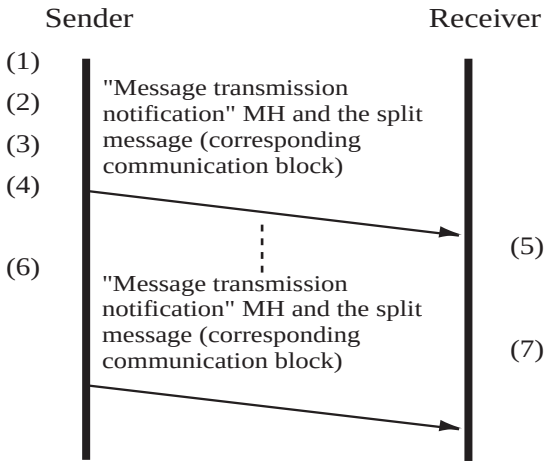

Fig. 22 Message transmission protocol

6.4.5 Message authentication code transmission protocol

6.4.5.1 Function

The message authentication code transmission protocol is provided to transmit a message authentication code from the sender after to sending the message for which the falsification detection function has been specified.

6.4.5.2 Sequence

(1) The sender generates a message authentication code. If the encryption method has been specified in the message transmission request protocol described in Section 6.4.2, the generated message authentication code is encrypted.

(2) The sender sends a message authentication code transmission notification code and the message authentication code.

"Message authentication code transmission notification" MH

MH.msgId = Message authentication code transmission notification MH.length = Message authentication code length MH.opt = 0

(3) The receiver receives the message authentication code transmission notification and the message authentication code. If the encryption method has been specified, the message authentication is decrypted. The sequence terminates abnormally in the following cases:
- "MH.msgId" is not a message authentication code transmission notification.

(4) The receiver authenticates the message. The message authentication code is generated for the entire message. If the encryption method has been specified, the following processes are performed:
- The message authentication code is generated for the original, unencrypted message.
- The generated message authentication code is encrypted.

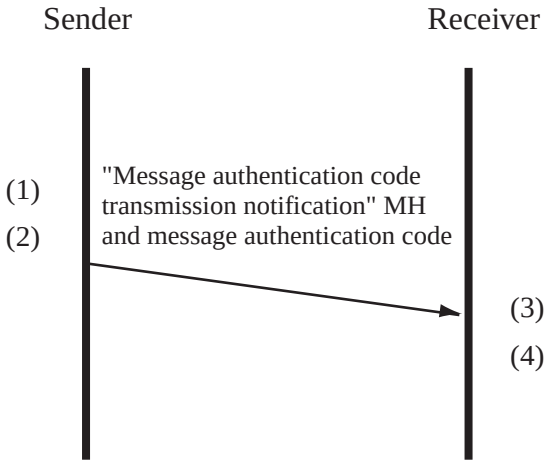

Fig. 23 Message authentication code transmission protocol

6.5 Through Mode Transmission Protocol

6.5.1 Function

The through mode transmission protocol is provided for message transmission whose negotiation procedure has been omitted for high-speed communication. This protocol is independent sequence of the message sending/receiving protocols described in Section 6.4, and does not allow encryption or the detection of falsification. This protocol, although not limited by the communication block size, is limited by the buffer size of lower-level protocols such as sockets.

6.5.2 Sequence

(1) The sender sends a through mode transmission notification code and a message.

"Through mode transmission notification" MH

MH.msgId = Through mode transmission notification MH.length = Message length MH.opt = 0

(2) The receiver receives the through mode transmission notification code and the message. When MH.msgId is for through mode transmission notification, the message data for MH.length is received.

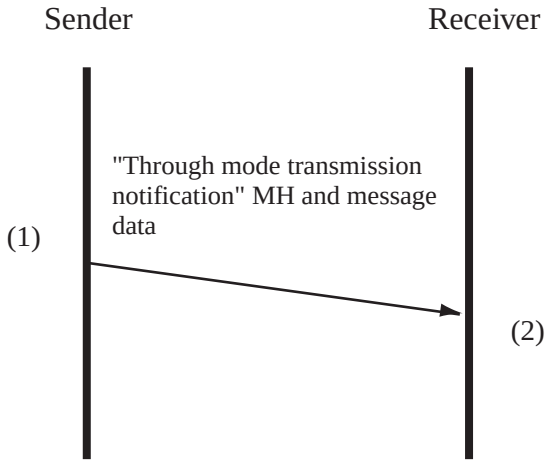

Fig. 24 Through mode transmission protocol

6.6 Line Disconnection Protocol

6.6.1 Function

The line disconnection protocol is provided to respond to a request for line disconnection.

6.6.2 Sequence

(1) The requester sends a disconnection request.

"Disconnection request" MH

MH.msgId = Disconnection request MH.length = 0 MH.opt = 0

(2) The responder receives the disconnection request.
(3) The responder sends the disconnection response.

"Disconnection response" MH

MH.msgId = Disconnection response MH.length = 0 MH.opt = ACK if disconnection is possible, or NAK if it is not possible.

If the line can be disconnected, connection is terminated. If the line cannot be disconnected, this protocol terminates and control moves to the next processing.

(4) The requester receives the disconnection response.
If MH.opt is ACK, connection is terminated. If MH.opt is NAK, this protocol terminates and control moves to the next processing.

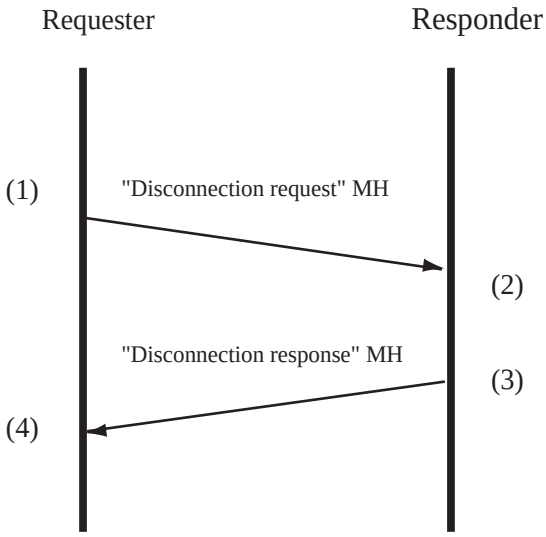

Fig. 25 Line disconnection protocol

6.7 Exceptional Protocols

6.7.1 Exceptional protocol 1 for the mutual authentication pass 2 protocol

6.7.1.1 Function

This protocol is a mutual authentication pass 2 protocol applied if the execution of the INTERNAL AUTHENTICATE command fails in the mutual authentication pass 1 protocol.

6.7.1.2 Sequence

(1) The client sends the following MH and terminates processing:

"Mutual authentication pass 2 notification" MH

MH.msgId = Mutual authentication pass 2 notification MH.length = 0 MH.opt = INTERNAL AUTHENTICATE command result code (not 0)

(2) The server receives the mutual authentication pass 2 notification and then terminates abnormally because "MH.opt" is not 0.

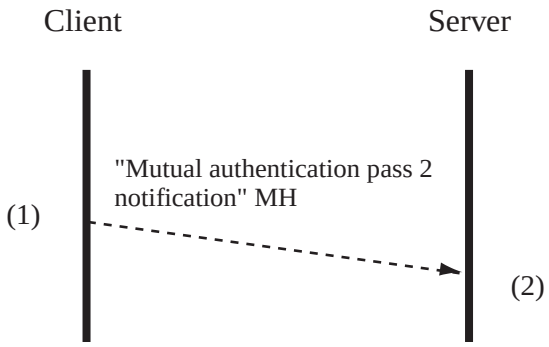

Fig. 26 Exceptional protocol 1 for the mutual authentication pass 2 protocol

6.7.2 Exceptional protocol 2 for the mutual authentication pass 2 protocol

6.7.2.1 Function

This protocol is a mutual authentication pass 2 protocol applied if the execution of the GET CHALLENGE command fails in the mutual authentication pass 1 protocol.

6.7.2.2 Sequence

(1) The client sends the following MH and terminates processing:

"Mutual authentication pass 2 notification" MH

MH.msgId = Mutual authentication pass 2 notification MH.length = 0 MH.opt = GET CHALLENGE command result code (not 0)

(2) The server receives the mutual authentication pass 2 notification and then terminates abnormally because "MH.opt" is not 0.

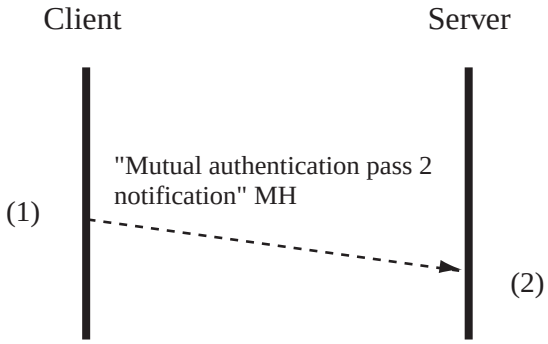

Fig. 27 Exceptional protocol 2 for the mutual authentication pass 2 protocol

6.7.3 Exceptional protocol 1 for the mutual authentication pass 3 protocol

6.7.3.1 Function

This protocol is a mutual authentication pass 3 protocol applied if the execution of the EXTERNAL AUTHENTICATE command fails in the mutual authentication pass 2 protocol.

6.7.3.2 Sequence

(1) The client sends the following MH and terminates processing:

"Mutual authentication pass 3 notification" MH

MH.msgId = Mutual authentication pass 3 notification MH.length = 0 MH.opt = EXTERNAL AUTHENTICATE command result code (not 0)

(2) The server receives the mutual authentication pass 3 notification and then terminates abnormally because "MH.opt" is not 0.

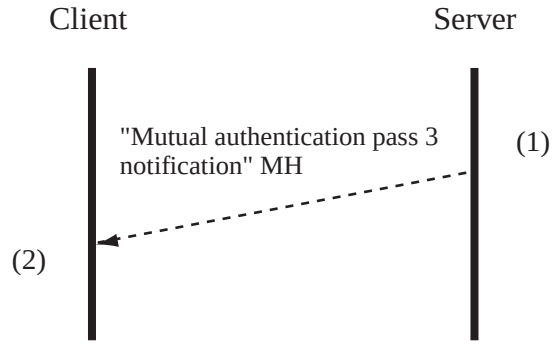

Fig. 28 Exceptional protocol 1 for the mutual authentication pass 3 protocol

6.7.4 Exceptional protocol 2 for the mutual authentication pass 3 protocol

6.7.4.1 Function

This protocol is applied if the execution of the INTERNAL AUTHENTICATE command fails in the mutual authentication pass 3 protocol.

6.7.4.2 Sequence

(1) The server fails to generate response code by INTERNAL AUTHENTICATE command.

(2) The client sends the following MH and terminates processing:

"Mutual authentication pass 3 notification" MH

MH.msgId = Mutual authentication pass 3 notification MH.length = 0 MH.opt = INTERNAL AUTHENTICATE command result code (not 0)

(3) The server receives the mutual authentication pass 3 notification and then terminates abnormally because "MH.opt" is not 0.

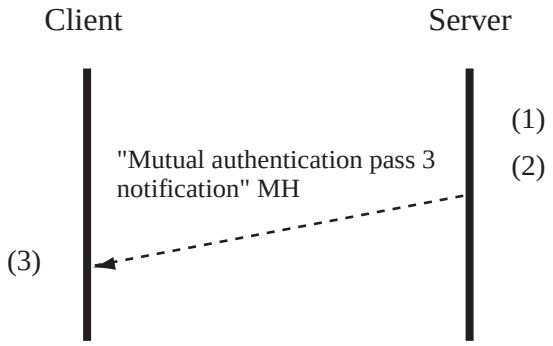

Fig. 29 Exceptional protocol 2 for the mutual authentication pass 3 protocol

6.7.5 Exceptional protocol for the message transmission request protocol

6.7.5.1 Function

This protocol is applied if the transmission method specified in the message transmission request protocol is incorrect.

6.7.5.2 Sequence

(1) The sender generates a message transmission request.

"Message transmission request" MH

MH.msgId = Message transmission request MH.length = 16 MH.opt = 0

(2) The sender generates transmission method data. The transmission method data refers to the following

(4 bytes x 4) of data:

Encryption method Encryption key ID Message authentication method Desired send message length

(3) The sender sends the message transmission request and the transmission method data.
(4) The receiver receives the message transmission request and the transmission method data.
(5) If the encoding method or the message authentication method is incorrect, the receiver sends the following MH and terminates processing:

"Message transmission response" MH

MH.msgId = Message transmission response MH.length = 0 MH.opt = NAK

(6) The sender receives the message transmission response and then terminates abnormally because "MH.opt" is not ACK.

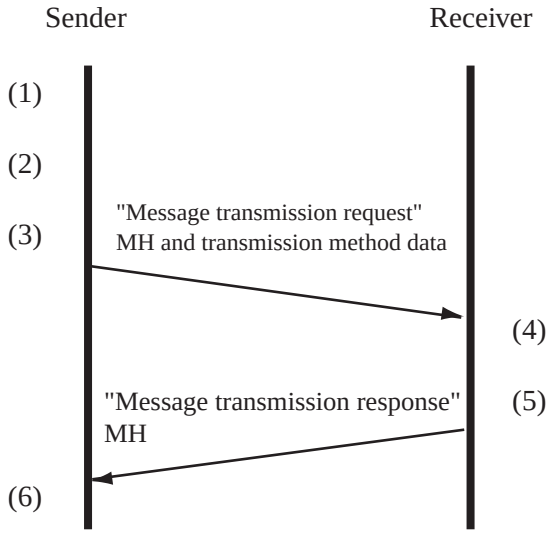

Fig. 30 Exceptional protocol for the message transmission request protocol

6.7.6 Exceptional protocol for the session key sharing protocol

6.7.6.1 Function

This protocol is provided to notify the receiver that an error has occurred if the sender fails to generate

a session key in the session key sharing protocol. This exception protocol occurs only when session key is to be shared.

6.7.6.2 Sequence

(1) The sender generates a random number for generating a session key using the GET CHALLENGE command and generates a session key from the random number using the INTERNAL AUTHENTICATE command.
(2) The sender sends the following MH and terminates processing:

"Random number for generating a session key transmission request" MH

MH.msgId = Random number for generating a session key transmission request MH.length = 0 MH.opt = GET CHALLENGE or INTERNAL AUTHENTICATE command result code (not 0)

(3) The receiver receives the session key for generating a random number transmission response and then terminates abnormally because "MH.opt" is not 0.

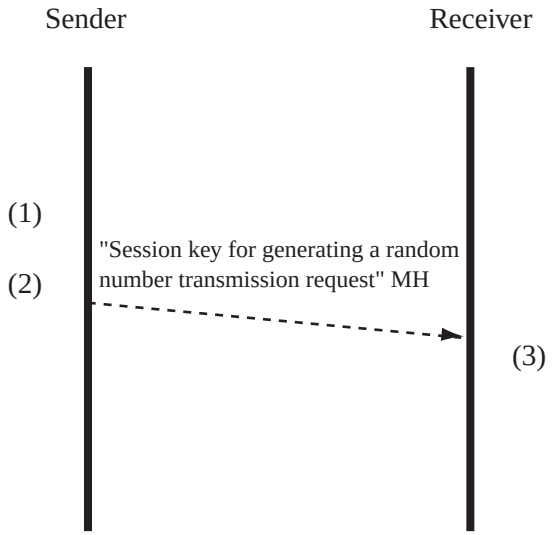

Fig. 31 Exceptional protocol for the session key sharing protocol