MBA 文書 0603 号

MBA Document 0603

The English translation Of MBA Standard 0301

MISAUTH Protocol Specification

The authoritive specification is Japansese one, MBA Standard 0203 (June 2004).

The Protocol Working Group in the Mobile Broadband Association (MBA) reviews a draft proposed by a member of the working group. After reviewing, the working group releases a proposal as a MBA standard through procedures.
This MBA standard 0301 was proposed by Mobile Internet Services, Inc. as `MIS AUTH Protocol Specification' and released through procedures.

モバイルブロードバンド協会

Mobile Broadband Association

www.mbassoc.org

Revision History

１．Terminology and Concept

１．１ Mobile Node

A Mobile Node is represented as “MN.”

An MN finds near base routers and requests a connection to one of them. After connecting to the base router, MN connects to the Internet via the base router.

１．２ Base Router

A Base Router is represented as “BR.”

A BR is fixedly installed and has permanent connectivity to the Internet. A BR processes requests from MNs and behaves as a router between the MNs and the Internet.

１．３ MISAUTH Server

A MISAUTH server is represented as “AS.”

An AS is fixedly installed and processes authentication requests from BRs. After receiving authentication requests, the AS authenticates users and sends back results of the authentications to the BRs.

１．４ Authentication Information

A pair of an account identifier and a password. Subset of NAI defined in RFC2486 is used as an account identifier. The differences between our NAI and the one in RFC2486 are:

1. realm MUST be specified with '@'
2. restriction of characters that can be used in username and realm.

A password is a raw byte stream. The maximum length of a NAI and a password is 253 bytes, respectively.

１．５ Account Identifier

A letter string concatenated of a MIS user name and a MIS domain name. A MIS user name is a letter string consisting of alphabets, numeric characters, ‘,’ and ‘_’. A MIS user name MUST begin with an alphabet character. A MIS domain name is a letter string consisting of alphabets, numeric characters and ‘-’. A MIS domain name CAN be concatenated with ‘.’. A MIS domain name MUST begin with an alphabet character.

１．６ Group

MIS users and BRs can belong to any number of groups including zero. Groups are used to authenticate MIS users. MIS users belonging to a group can use only the BRs belonging to the same group. The AS checks MIS user's group and BR's group at a time of authentication. A group is identified by 32-bit integer.

１．７ MIS Domain

A MIS user belongs to a MIS domain. A MIS domain is included in an account identifier a BR sends as "<MIS user name>@<MIS domain name>." Each AS belongs to MIS domains. An AS locally authenticates MIS users if and only if the MIS users and the AS belong to the same MIS domain. Otherwise, the AS forwards its authentication to an associated AS belonging to the MIS users' domain or a default AS. This function is called proxy. When there is no associated AS and no default AS, the authentication results in failure.

１．８ Security Type

A set of authentication, session-key generation and encryption methods used between an MN and an AS.

２. Introduction

The MISAUTH protocol is a protocol at an application layer to employ an authentication server in a MIS system. The MISAUTH protocol is based upon the RADIUS protocol.

An AS manages authentication information, collective pairs of a MIS user name and a MIS password, on behalf of BRs. Therefore, multiple BRs can easily share account information. Using proxy function, roaming service can be easily provided even though MIS user informationmanaged distributedly.

３. Message Format

All messages of the MISAUTH protocol begin with a MISAUTH protocol header. More than zero objects follow a MISAUTH protocol header. A type of messages is identified by a MISAUTH protocol header. There are three types of messages as follows:

１. Access Request Message

２. Access Accept Message

３. Access Reject Message

An object included in each message is describedlater.

MISAUTH protocol messages are carried in UDP between a BR and an AS as well as between ASes. The UDP port number on which an authentication request is received at an AS is 1812 as a default. Any port number can be used when requests are sent by a BR or an AS. The maximum length of a MISAUTH protocol message is 66535 bytes.

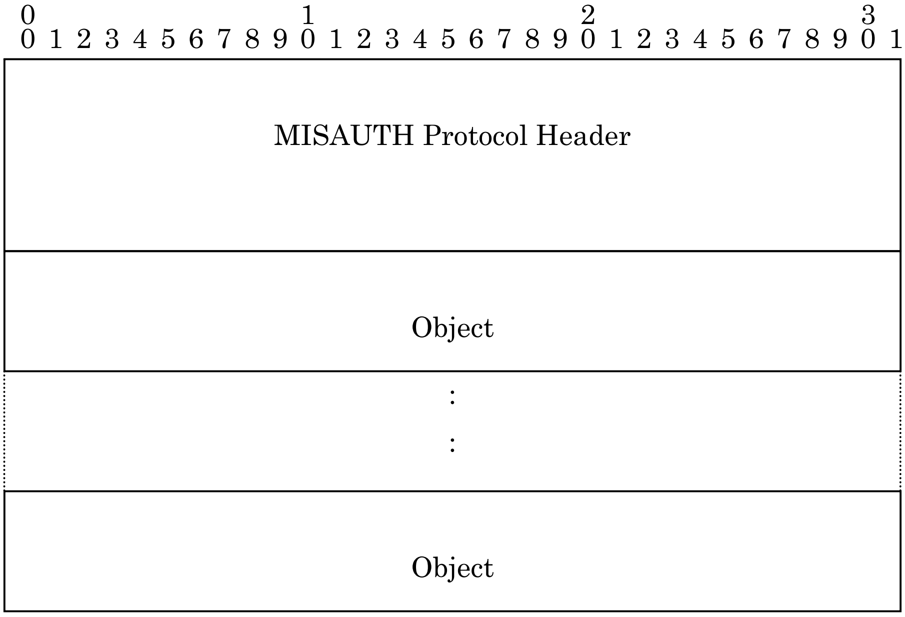

Fig. 1 Message Format

３． １ MISAUTH Protocol Header

Length of a MISAUTH protocol header is 20 bytes. The format is as follows:

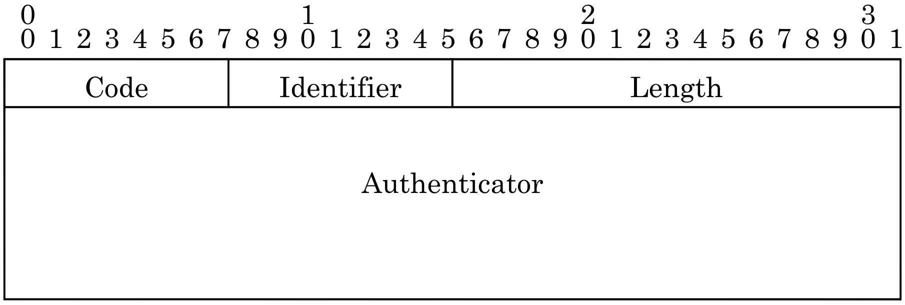

Fig. 2 MIS Protocol Header Format

Code Field (1 byte)

Indicates a type of this MISAUTH message. The field length is a 1 byte and unsigned 8-bite integer.

1	Access Request
2	Access Accept
3	Access Reject

Identifier Field (1 byte)

Represents association between an access request message and its reply message. Values in an identifier field of both messages are the same. Length of the field is 1 byte and the value is an unsigned 8-bit integer. Each message is identified by a set of a source IP address, a destination IP address, a port number and an identifier field in its message.

Length Field (2 byte)

The total length of a MISAUTH message in byte begins with a

MISAUTH protocol header. Field length is 2 bytes and its value is an unsigned 16-bit integer.

An actually received MISAUTH message can be longer than the length indicated in a length field. In this case, extra trailing parts of the message must be ignored.

On the other hand, if a MISAUTH message is shorter than the length field value, it means an error and its message MUST be ignored.

Authenticator (16 bytes)

Each association of AS-AS and AS-BR shares a shared secret in advance. When an AS sends a reply message to a BR or another AS, it calculates a value to store in the authenticator field in the message. The value is calculated by applying HMAC-MD5 to the whole message while the authenticator field is set to all zero. When receiving a reply message, an AS or BR calculates its authenticator and checks if the received message is valid. Length of this field is 16 bytes.

３．２ MISAUTH Protocol Object

Format of each object following a MISAUTH protocol header is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
|     Type      |    Length     |           Value...            |
+---------------+---------------+-------------------------------+
```

Fig. 3 MISAUTH Protocol Object Format

Type Field (1 byte)

Is a 8-bit unsigned integer and Indicates a type of the object.

1	NAI
4	Base Router IPv4 Address
33	Proxy Request
95	Base Router IPv6 Address
200	Authentication Data
201	Authentication Hash Value
202	Seed of Session Key
203	Session Key
204	Geographical Information
205	Security Type
206	Group Type

Length Field (1 byte)

Is a 8-bit unsigned integer and indicates length of this object. The length includes the type and length fields. The minimum value is 2. Length of the length field is 1 byte.

Value Field (Variable length: length – 2 byte)

Represents data of the object. Length of this field is variable and the value is (length – 2). If a value in the length field is 2, the value field does not exist. The maximum length of this field is 253 bytes.

４．Object Format

４．１ NAI Object

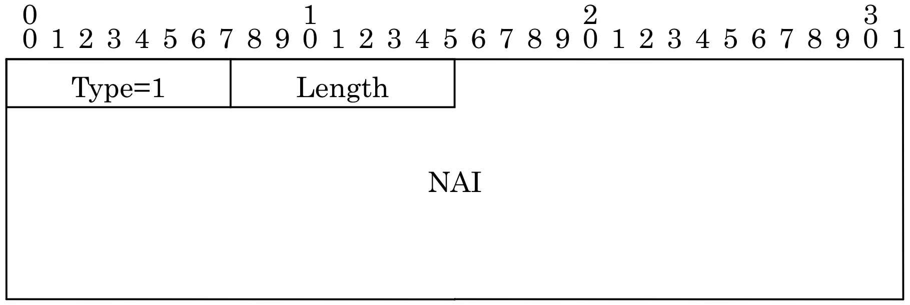

Fig. 1 NAI Object Format

Type is 1.

Length is equal to or more than 3.

The objects indicates a user identifier for authentication. Length of this field is variable. The value of NAI is a byte stream. If the value is terminated with a null character, its null character is also regarded as a part of the identifier. Therefore, a null character as a terminator MUST NOT included in the NAI field.

４．２ Base Router IPv4 Address Object

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
|    Type=4     |   Length=6    |         IPv4 address          |
+---------------+---------------+-------------------------------+
|         IPv4 address          |
+-------------------------------+
```

Fig. 2 Base Router IPv4 Address Object Format

Type = 4

Length = 6

The object indicates an IPv4 address of the base router which sends a MISAUTH message including this object. An IPv4-type address can be stored. An object whose length is not 6 is ignored.

4．3 Proxy Request Object

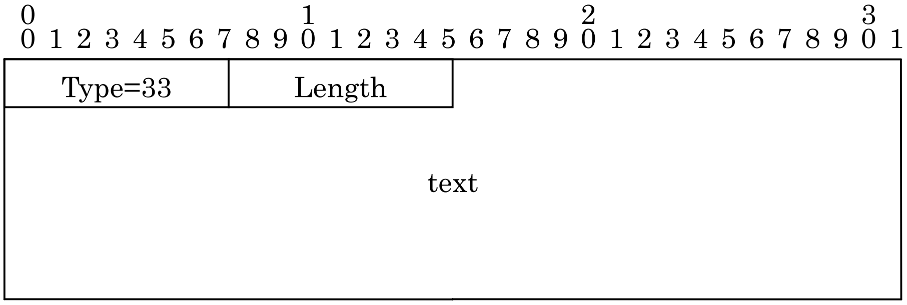

Fig. 3 Proxy Request Object Format

Type = 33

Length is equal to or more than 2.

An AS acting as a proxy server MUST add this object into an authentication request when forwarding the request to another AS. If the forwarded request with the object returns to its original AS, the AS MUST remove the object. A value in this object is subject to the proxy server. All ASes except for the AS adding this object MUST not process in accordance with this object. This object can be included more than once when a message goes through multiple proxy servers.

４． ４ Base Router IPv6 Address Object

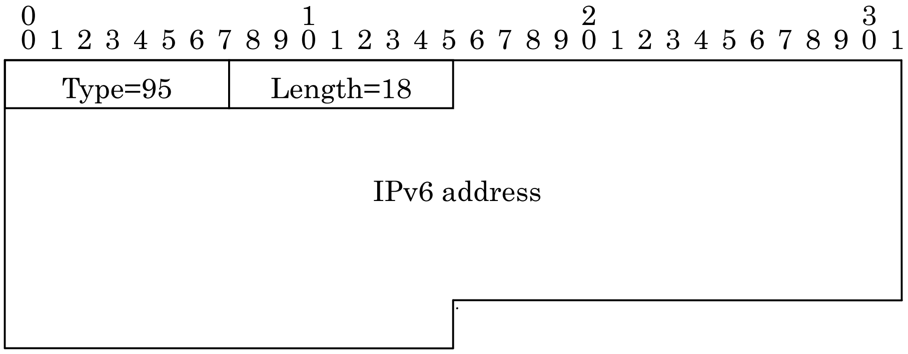

Fig. 4 Base Router IPv6 Address Object Format

Type = 95

Length=18

This indicates an IPv6 address of a BR which sends a message including this object. An object whose length is not 18 is ignored. When a BR has multiple IPv6 addresses, this object can be included more than once in a message.

４．５ Authentication Data Object

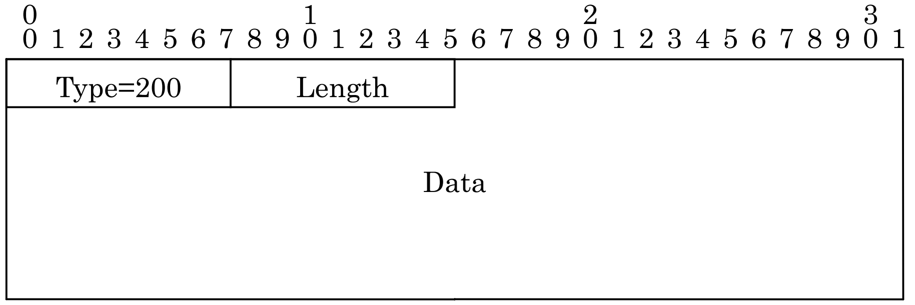

Fig. 8 Authentication Data Object

Ttype = 200

Length is equal to or more that 2.

This indicates data to be used to authenticate a user. Length of this object is variable and equal to or less than 253 byte.

４．６ Authentication Hash Value Object

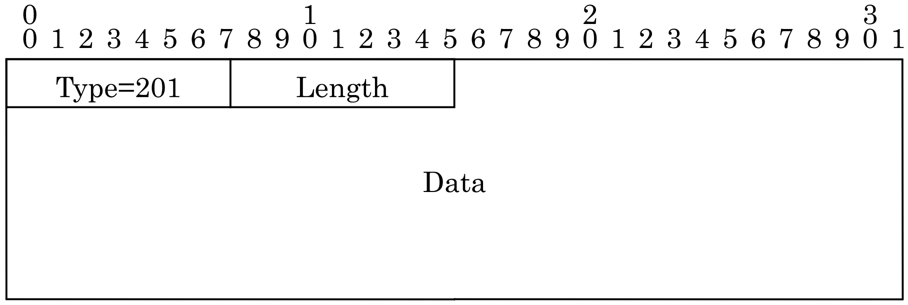

Fig. 5 Authentication Hash Value Object Format

Type = 201

Length is equal to or more than 2.

This is a hash value to be used to authenticate a user. Length is variable and equal to or less than 253 byte. The method to calculate a hash value is shared between a BR and an AS as specified in a security type object defined later.

4． 7 Seed of Session Key Object

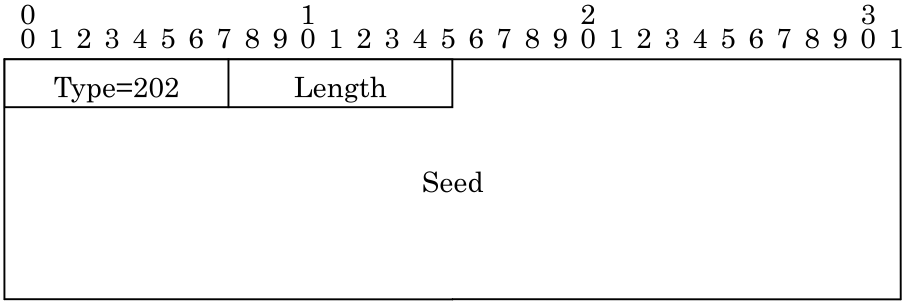

Fig. 6 Seed of Session Key Object Formats

Type = 202

Length is equal to or more than 2.

This is a seed of a session key sent to BR. A session key is generated when authentication succeeds. Length is variable and equal to or less than 253 bytes. An MN and an AS share a security type to generate session key.

４．８ Session Key Object

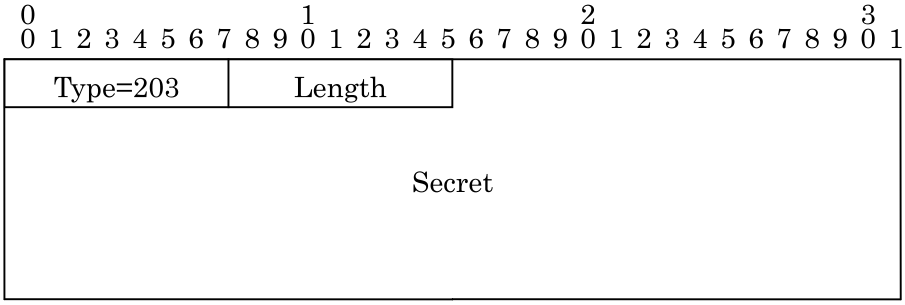

Fig. 7 Session Key Object Format

Type = 203

Length is equal to or more than 2.

This is a session key that an AS sends to a BR on successful authentication. Length is variable. Normally, a session key is encrypted. Security type object described later helps a BR and an AS share how to generate a session key and how to encrypt a session key.

4．９ Geographic Information Object

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td>type=204</td><td>length=14</td><td colspan="2">latitude</td></tr>
<tr><td colspan="2">latitude</td><td colspan="2">longitude</td></tr>
<tr><td colspan="2">longitude</td><td colspan="2">height S</td></tr>
<tr><td colspan="2">height G</td><td colspan="2"></td></tr>
</table>

Fig. 8 Geographic Information Object Format

Type = 204

Length=14

Latitude — Latitude in signed 32-bit integer. Represented in 1/65536 degree.

Longitude — Longtitude in signed 32-bit integer. Represented in 1/65536 degree.

Height S — Height from sea in signed 16-bit integer. Represented in a meter.

Height G — Height from the ground in signed 16-bit integer. Represented in a meter.

This object indicates geographic location information by latitude, longitude and height. Sign of north latitude and east longitude are represented as positive, south latitude and west longitude as negative. Note that 0x80000000 means “no information.” A positive value means over the sea surface or over the ground. A negative value means under the sea surface or under the ground. Note that 0x8000 means “no information.” Length is 14. If length is not 14, this object is ignored.

４．１０ Security Type Object

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
|   Type=205    |   Length=4    |            method             |
+---------------+---------------+-------------------------------+
```

Fig. 9 Security Type Object

Type =205

Length = 4

The object represents a set of an authentication type, a way to generate session key and a way to encrypt data used with a BR which sends a MISAUTH message including this.“method” is an unsigned 8-bit integer and represents a security type. Length is 4. If length is not 4, this object is ignored.

４．１１ Group Type Object

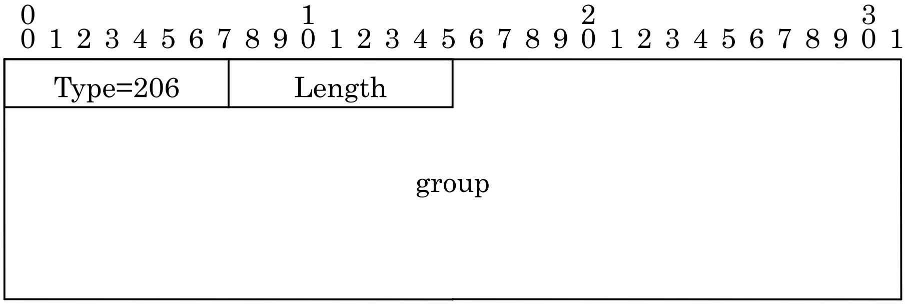

Fig. 10 Group Type Object Format.

Type = 206

This indicates a group that a BR belongs to. A group is in 32-bit integer. All the groups can be listed in this object when a BR belongs to multiple groups. Length is (2 + 4n). If length is not equal to (2 + 4n), this object is ignored.

５．Message Format

５．１ Access Request Message

Access Request Message MUST include following objects:

NAI Object
Authentication Data Object
Authentication Hash Value Object
Seed of Session Key Object
Security Type Object
Geographic Information Object

Access Request Message CAN include following objects:

Group Type Object
Proxy Request Object
Base Router IPv4 Address Object
Base Router IPv6 Address Object

Messages which do not include mandatory objects are regarded as invalid and are rejected. Objects which are not necessary are all ignored. Only Proxy Request Object and Base Router IPv6 Address Object can be included more than once in a message. When the other objects are duplicated, only the last object is valid and the other duplicated objects are ignored.

５．２ Access Accept Message

Access Accept Message MUST include following objects:

Session Key Object
Authentication Hash Value Object

Access Accept Message CAN include following object:

Proxy Request Message

Other objects are ignored when included.

５．３ Access Reject Message

There is no mandatory object that MUST be included in Access Reject Message.
Access Reject Message CAN include following object:

Proxy Request Object

６．Process Flow

６．１ Successful Authentication Case

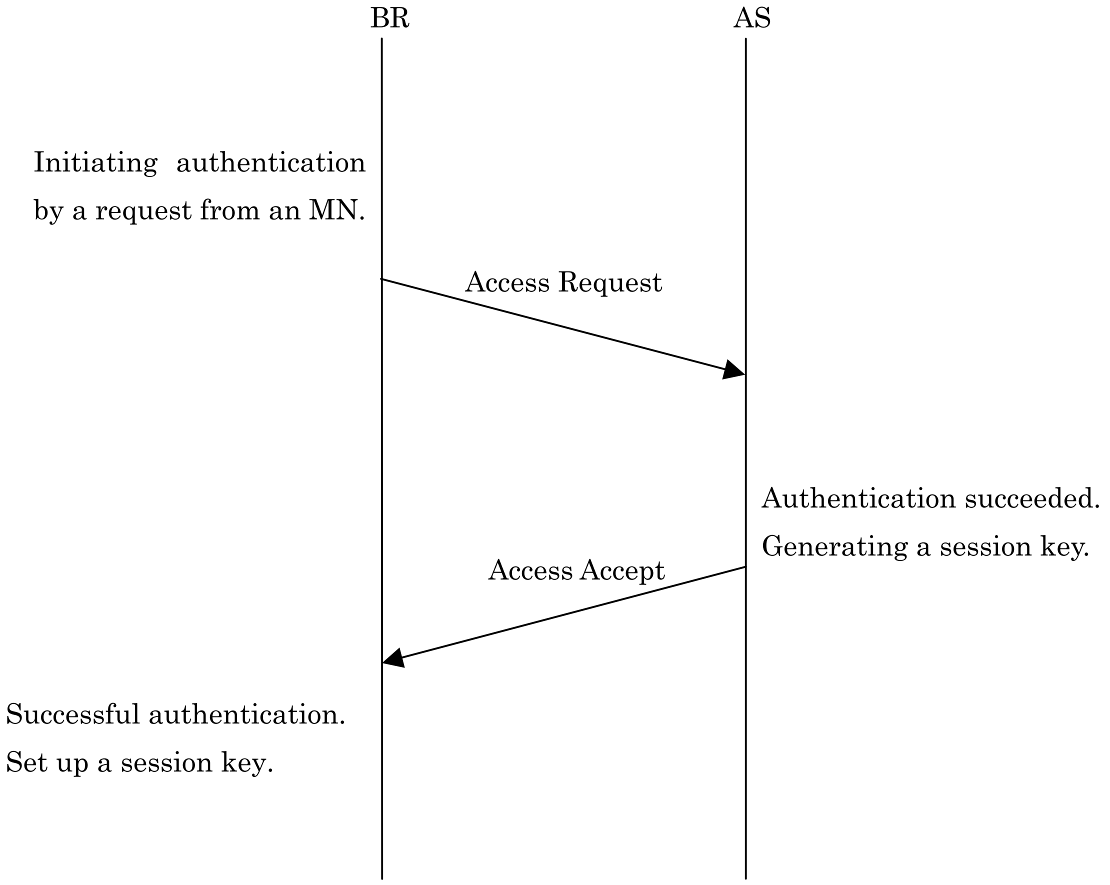
BR
AS
Initiating authentication
by a request from an MN.
Access Request
Authentication succeeded.
Generating a session key.
Access Accept
Successful authentication.
Set up a session key.

Fig. 11 Process Flow of Successful Authentication.

Process flow in a case of successful authentication as follows:

１．An MN requests to be authenticated.

２．A BR sends to an authentication request to an AS.

３．Authentication succeeds.

４．The AS generates a session key.

５．The AS sends an Authentication Success message including the session key to the BR.

６．Authentication succeeds.

Note that an AS locally authenticates MIS users when the domain its user belongs to is the same as one that the AS belongs to.

６． ２ Failure Authentication Case

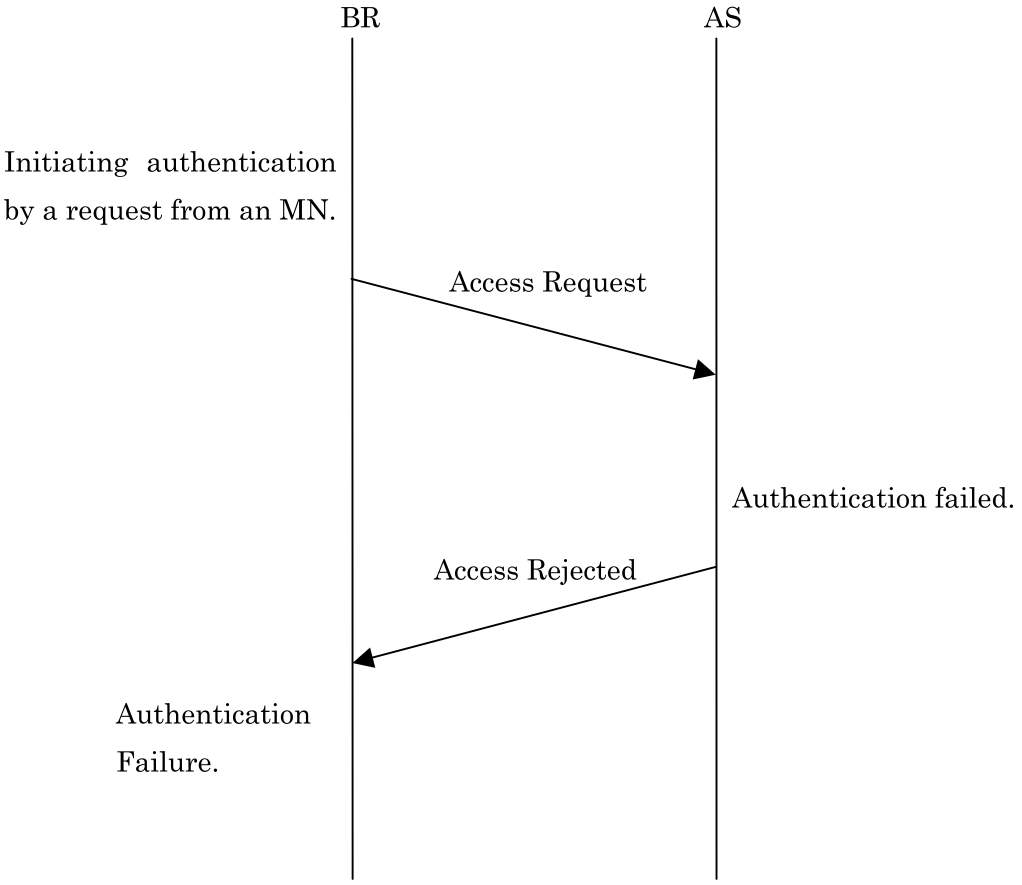

Fig. 12 Process Flow of Failure Authentication.

Process flow in case of failure authentication as follows:

１． An MN requests to be authenticated.
２． A BR sends an authentication request to an AS.
３． Authentication failed.
４． The AS sends authentication failure message to the BR.
５． Authentication failed.

Note that an AS locally authenticates MIS users when the domain its user belongs to is the same as one that the AS belongs to.

６．３ No Response from AS Case.

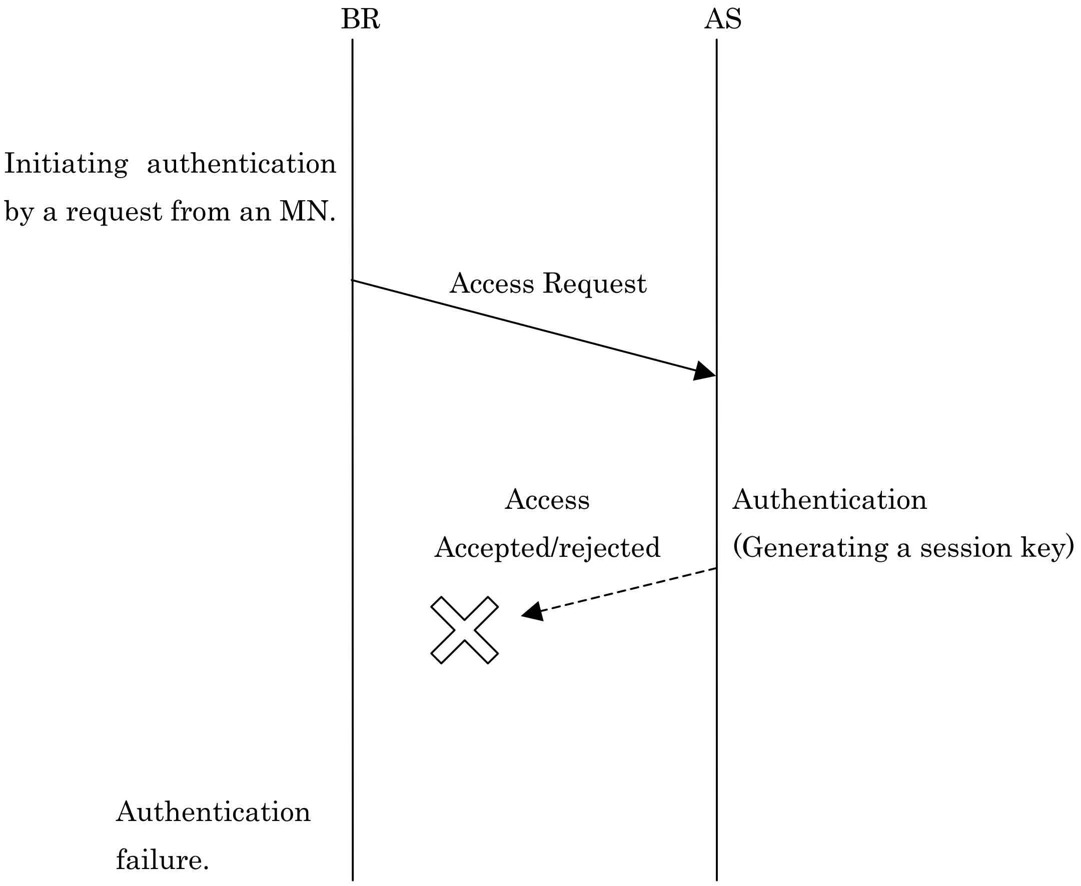
BR
AS
Initiating authentication by a request from an MN.
Access Request
Access Accepted/rejected
Authentication (Generating a session key)
Authentication failure.

Fig. 13 Process Flow of No Response from AS

Process flow when a BR receives no reply from an AS as follows:

１. An MN initiates to be authenticated.
２. A BR sends an authentication request to an AS.
３. Authentication succeeds/failed.
４. (When authentication succeeds) the AS generates a session key.
５. The AS sends a reply message to the BR.

In this case, authentication fails due to timeout at an MN. This occurs when an AS cannot reply for a request for some reasons or when a reply from the AS is delayed due to network conditions, etc.

６． ４ Process Flow with Proxy Function among ASes

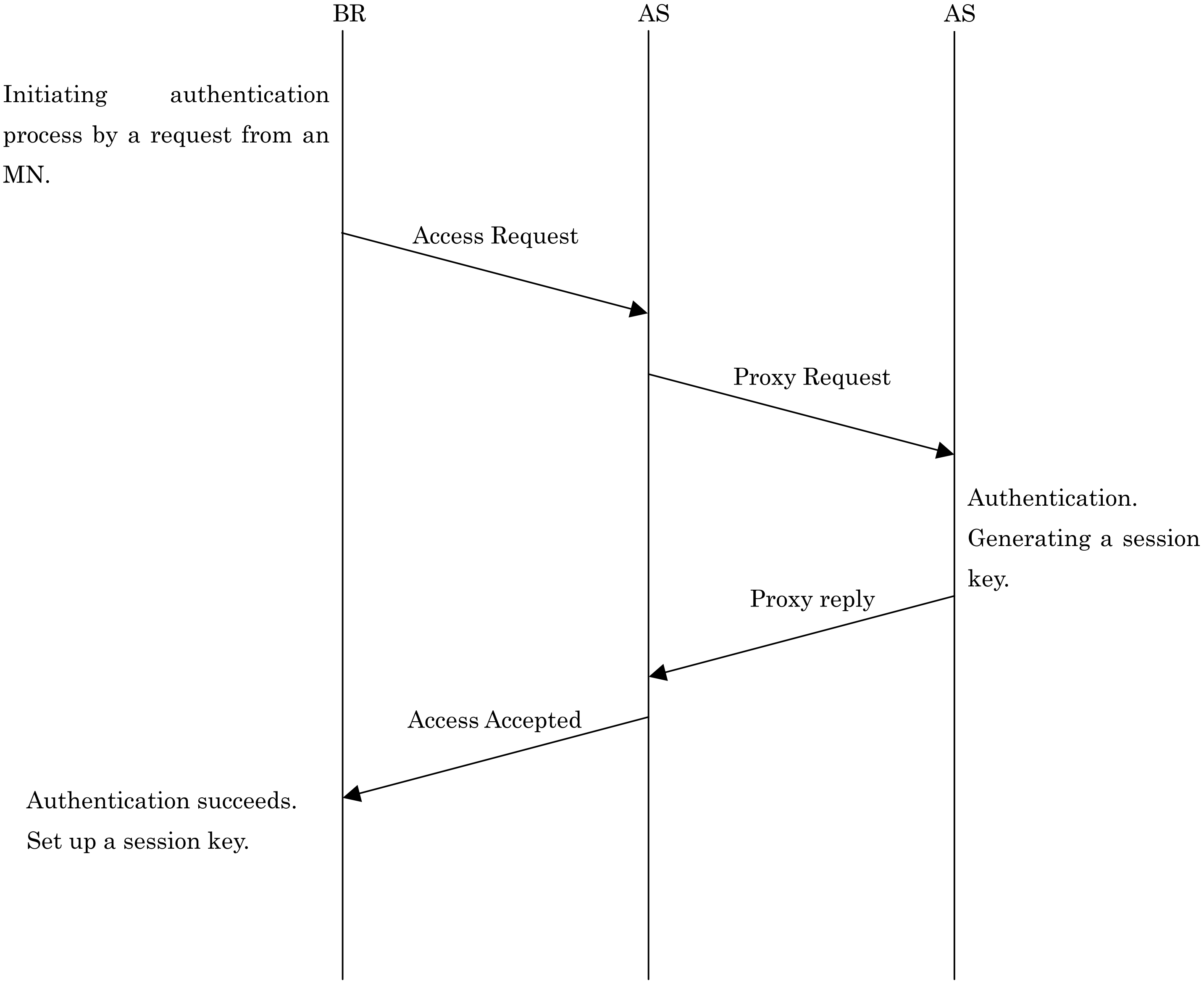

Fig. 14 Process Flow with Using Proxy Function.

Process flow when the first AS acts as a proxy as follows:

１． An MN requests to be authenticated.

２． A BR sends an authentication request to an AS.

３. The AS sends a proxy request to another AS.
４. Authentication succeeds.
５. The AS generates a session key.
６. The AS sends the session key to the associated AS.
７. The AS sends the session key to the BR.
８. Authentication succeeds.

Note that an AS forwards an authentication request from MIS users to associated another AS when the domain its user belongs to is different from one that the AS belongs to.

7 . Authentication by AS

An AS authenticates a MIS user based upon an Access Request message sent by a BR and user information that the AS knows in advance. User information consists of a MIS user name and a MIS password. An authentication request includes authentication data and its hash value calculated by a security type specified in the Security Type object. The AS calculates a hash value of the whole message by the security type specified in a Security Type object sent by the BR. If authentication data included in the request message is same as the one locally calculated, authentication succeeds. The encryption type specified in a Security Type object is described in section 12.

8．Generating Session Key

an MN and AS independently calculates a hash value by the same seed, same secret key and same hash function. The secret key is shared between the MN and the AS in advance. The seed and the security type is informed by the BR to the AS. The seed is specified in Seed of Session Key object and the security type is specified in Security Type object. Both of them are included in an access request message sent by the BR.

After succeeding in authentication, the AS calculates a session key and sends an Access Accepted message which includes the session key in the Session Key objects. An AS normally encrypts a session key in order to be secure. The encryption algorithm is shared between the BR and the AS by a Security Type object sent by the BR. An encryption algorithm specified in a Security Type object is described in section 12.

９． Avoiding Duplicated Login

In this document, duplicated login is that multiple users login with the same account in order to invalidly use network resources. An AS detects and avoids this duplicated login as described in this section.

９．１ Black List

In order to Detect duplicated login, an AS registers its user ID into a black list in the AS. The AS rejects a request from users listed in the black list for 10 minutes since the user ID is registered. A user ID is removed from the black list after 10 minutes since the user ID is registered.

９．２ Detecting Duplicated Login

1. If it passes 10 minutes or more after the last authentication, an AS regards its authentication as new one and does not check duplicated login.
2. If moving distance is within (radio reachable distance x 2 x margin), an AS regards it as an MN reconnects the same or another neighboring BR. In this case, the AS does not check duplicated login.
3. If elapsed time is zero after the last authentication, an AS regards its user ID as duplicated and rejects the request. In addition, the AS adds its user ID into a black list.
4. If moving speed is more than (MN's speed limit x margin), an AS regards its user ID as duplicated and rejects the request. In addition, the AS adds its user ID into a black list.

- Currently, radio reachable distance is configured to 200 m.
- Currently, margin is configured to 2.
- Currently, MN's speed limit is configured to 60km/h.
- Moving distance can be calculated by following formula:

$$DISTANCE = 2 \times R \times \sqrt{\sin\left(\frac{la2 - la1}{2}\right)^2 + \cos(l2) \times \cos(l1) \times \sin\left(\frac{lo2 - lo1}{2}\right)^2}$$

R	Radius of globe (6367000.0)		
la1	Latitude of location 1	la2	Latitude of location 2
lo1	Longitude of location 1	lo2	Longitude of location 2.

９．３ Limitation

1. When users do duplicated login within a few hundred meters (e.g. when one user uses multiple terminals or when multiple users are in the same area), an AS cannot detect duplicated login because it cannot be differentiated between handover and duplicated login.
2. There may be an error in calculation of moving distance in a long distance because it is approximated by a linier line. (note that this is enough to detect duplicated login).

９．４ Future Works

Detection of duplicate login should be improved by making MN's speed limit configurable based upon MN's circumstance (currently, MN's speed limit is constant).

１０. Proxy Function

Proxy function is that an AS forwards an authentication request from an BR or another AS to the third AS. When the AS forwards an authentication request by the proxy function, the AS MUST add a Proxy Request object at the end of the original received message. The value in its object depends on an implementation of the AS acting as proxy server, all ASes except for the AS adding the Proxy Request object MUST not handle processes dependent on this object. When the AS receives a reply from other ASes, the AS MUST remove the Proxy Request object that the AS itself added before.

When an AS forwards an authentication request by the proxy function, the AS forwards one authentication request to other ASes at a time without any special handling.

Even if an AS receives no reply from proxy servers, the AS MUST NOT handle special process but leave it as it timeouts at the original source of the authentication request.

An AS CAN reject an authentication request that includes too many Proxy Request objects. The maximum number of accepted Proxy Request objects depends on AS implementations.

A session key included in an Access Accept message is encrypted by an encryption algorithm specified in a Security Type object sent by a BR. The encryption algorithm specified by the Security Type object is described in section 12.

- Currently, the maximum number of Proxy Request objects in an authentication request is configured to 64.

１１．Multiplexing Authentication Server

An AS can be multiplexed in order to make authentication reliable and fast. In this case, a BR and an AS SHOULD act as defined in this section.

１１．１ BR Behavior

When a BR is configured to send an authentication request to multiple ASes, the BR acts as follows:

１．The BR sends authentication requests to all configured Ases at once.
２．The BR takes the first repl from the ASes.

１１．２ AS Behavior

When an AS acts as proxy server and is configured to forward an authentication request to multiple ASes, the AS acts as follows. Note that the forwarding AS MUST be configured to specify priority of each ASes belonging to the other MIS domains that the forwarding AS does not belong to.

１．When using a proxy function, the AS forwards an authentication request to all configured proxy ASes at each 19-21 (pseudo random number) times of forwarding authentication requests.
２．If the AS does not forward the authentication request to all configured ASes, the AS forwards it to a prior proxy AS. The prior proxy AS is determined after excluding proxy ASes which do not respond for the last 5 minutes even though requests are sent to them for the last 5 minutes.
３．If there is no proxy AS matching with the condition 2., the forwarding AS forwards the authentication request to the proxy AS whose sending time is the oldest among the proxy ASes. If the last sent times are the same, a higher prioritized AS is preferable.

１２． Security Type

Security Types are as follows:

HMAC-MD5/HMAC-MD5/HMAC-MD5	1

The right number means a value of security type included in a Security Type object. This section describes security type in detail.

１２． １ HMAC-MD5/HMAC-MD5/HMAC-MD5

HMAC-MD5 is used for authentication, HMAC-MD5 is used to generate a session key, and HMAC-MD5 and exclusive OR are used to encrypt data.

１２． １． １ Generating Hash Value for Authentication

HMAC-MD5 is used to calculate a hash value for authentication. A 16-byte byte stream, hash value is calculated by applying HMAC-MD5 to authentication data included in an Authentication Request message. During calculation, a MIS password shared between an MN and an AS in advance is used as a key of HMAC-MD5. When the calculated byte stream is the same as a value in Authentication Data Hash Value object, authentication succeeds.

１２． １． ２ Generating Session Key

When authentication succeeds, an MN and an AS use HMAC-MD5 to generate and share a session key. The MN and AS apply HMAC-MD5 with a MIS password shared between them to seed of the session key included in an Authentication Request message. Then, the MN and AS acquire a 16-byte byte stream that is used as a session key.

１２． １． ３ Encryption of Session Key

When authentication succeeds, HMAC-MD5 and exclusive OR are used in order to encrypt a session key delivered from an AS to a BR or a proxy AS. The AS apply HMAC-MD5 with a secret shared with the BR or the proxy AS to an authentication hash value included in the Authentication Request message. Then, the ASapplies exclusive OR to the result of HMAC-MD5 and the session key and get a 16-byte byte stream, encrypted session key. The AS sends the encrypted session key to the proxy AS or the BR. The authenticating AS MUST put an authentication hash value into a reply message in order to decrypt the encrypted session key.

１３．Appendix

１３．１ Figure and Table Number