1. **IEEE P802.21 Media Independent Handover Services**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Solution for LB7 Cmt#130** | | | | |
| **Date:** 2013-08-23 | | | | |
| **Author(s):** | | | | |
| **Name** | **Affiliation** | **Address** | **Phone** | **Email** |
| Yoshikazu Hanatani | Toshiba | 1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582 Japan |  | [yoshikazu.hanatani@toshiba.co.jp](mailto:yoshikazu.hanatani@toshiba.co.jp) |

Abstract

This is a contribution to solve the comment #130 in 802.21-13-0113-12-MuGM-lb7-commentary-file.

1. Overview
2. Normative references
3. Definitions
4. Abbreviations and acronyms
5. General Architecture
6. MIHF services
   1. General
   2. Service management
7. Service access points (SAPs) and primitives
   1. Introduction
   2. SAPs
   3. MIH\_LINK\_SAP primitives
   4. MIH\_SAP primitives
      1. MIH\_Capability\_Discover
      2. MIH\_Register
      3. MIH\_DeRegister
      4. MIH\_Event\_Subscribe
      5. MIH\_Event\_Unsubscribe
      6. MIH\_Link\_Detected.indication
      7. MIH\_Link\_Up.indication
      8. MIH\_Link\_Down.indication
      9. MIH\_Link\_Parameters\_Report.indication
      10. MIH\_Link\_Going\_Down.indication
      11. MIH\_Link\_Handover\_Imminent.indication
      12. MIH\_Link\_Handover\_Complete.indication
      13. MIH\_Link\_PDU\_Transmit\_Status.indication
      14. MIH\_Link\_Get\_Parameters
      15. MIH\_Link\_Configure\_Thresholds
      16. MIH\_Link\_Actions
      17. MIH\_Net\_HO\_Candidate\_Query
      18. MIH\_MN\_HO\_Candidate\_Query
      19. MIH\_N2N\_HO\_Query\_Resources
      20. MIH\_MN\_HO\_Commit
      21. MIH\_Net\_HO\_Commit
      22. MIH\_N2N\_HO\_Commit
      23. MIH\_MN\_HO\_Complete
      24. MIH\_N2N\_HO\_Complete
      25. MIH\_Get\_Information
      26. MIH\_Push\_Information
      27. MIH\_Push\_key
      28. MIH\_LL\_Auth
      29. MIH\_Net\_HO\_Bcast\_Commit
      30. MIH\_Configuration\_Update
      31. MIH\_MN\_Group\_Manipulate
      32. MIH\_Net\_Group\_Manipulate
      33. MIH\_Pull\_Credential
          1. MIH\_Pull\_Credential.request
          2. MIH\_Pull\_Credential.indication
          3. MIH\_Pull\_Credential.response
             1. Function

This primitive is generated by an MIH User in order to deliver a credential for IEEE 802.21a to an MN or other PoS.

* + - * 1. Semantics of service primitive

MIH\_Pull\_Credential.response (

DestinationIdentifier,

EncryptedCredential

)

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| DestinationIdentifier | MIHF\_ID | Specifies the requestor of the credential. |
| EncryptedCredential | ENCRYPTED\_KEY | Encrypted credential used for creating an EAP-generated MIH SA. |

* + - * 1. When generated

An MIH User generates this primitive using a leaf key corresponding with the credential requester.

* + - * 1. Effect on receipt

Upon receipt of this primitive, the MIHF on the PoS generates an MIH\_Pull\_Credential response message to the destination MN or PoS.

* + - 1. MIH\_Pull\_Credential.confirm
         1. Function

This primitive is generated by an MIHF that receives an MIH\_Pull\_Credential response, in order to inform of the credential received by the MIH User.

* + - * 1. Semantics of service primitive

MIH\_Pull\_Credential.confirm (

SourceIdentifier,

Credential

)

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| SourceIdentifier | MIHF\_ID | Identifies the remote MIHF that invoked MIH\_Pull\_Credential response. |
| Credential | CERTIFICATE | X.509 certificate |

* + - * 1. When generated

The MIHF that receives an MIH\_Pull\_Credential response message generates this primitive to indicate the credential.

* + - * 1. Effect on receipt

After verification, validated credential keys within their expiration period can be utilized for IEEE 802.21a.

* + 1. MIH\_Push\_Credential
    2. MIH\_Revoke\_Credential
  1. MIH\_NET\_SAP primitives

1. Media independent handover protocol
   1. Introduction
   2. MIH protocol description
   3. MIH protocol identifiers
   4. MIH protocol frame format
   5. Message parameter TLV encoding
   6. MIH protocol messages
      1. MIH messages for service management
         1. MIH\_Capability\_Discover request
         2. MIH\_Capability\_Discover response
         3. MIH\_Register request
         4. MIH\_Register response
         5. MIH\_DeRegister request
         6. MIH\_DeRegister response
         7. MIH\_Event\_Subscribe request
         8. MIH\_Event\_Subscribe response
         9. MIH\_Event\_Unsubscribe request
         10. MIH\_Event\_Unsubscribe response
         11. MIH\_Auth indication
         12. MIH\_Auth request
         13. MIH\_Auth response
         14. MIH\_Termination\_Auth request
         15. MIH\_Termination\_Auth response
         16. MIH\_Push\_key request
         17. MIH\_Push\_key response
         18. MIH\_LL\_Auth request
         19. MIH\_LL\_Auth response
         20. MIH\_Configuration\_Update indication
         21. MIH\_MN\_Group\_Manipulate request
         22. MIH\_MN\_Group\_Manipulate response
         23. MIH\_Net\_Group\_Manipulate request
         24. MIH\_Net\_Group\_Manipulate indication
         25. MIH\_Net\_Group\_Manipulate response
         26. MIH\_Pull\_Credential request
         27. MIH\_Pull\_Credential response

The corresponding MIH primitive of this message is defined in 7.4.33.3.

This message is used by the MIHF to deliver a credential from a PoS used for creating an EAP-generated MIH SA. EncryptedCredential is decrypted by the leaf key of the MN.

|  |
| --- |
| MIH Header Fields (SID=1, Opcode=2, AID=13 ) |
| **Source Identifier** = sending MIHF ID  (Source MIHF ID TLV) |
| **Destination Identifier** = receiving MIHF ID  (Destination MIHF ID TLV) |
| EncryptedCredential  (EncryptedCredential TLV) |

* + - 1. MIH\_Push\_Credential request

The corresponding MIH primitive of this message is defined in 7.4.34.1.

This message is used by the MIHF to deliver a credential encrypted by the leaf key that the MIH node identified by the Destination Identifier holds to the MIH node.

|  |
| --- |
| MIH Header Fields (SID=1, Opcode=1, AID=14 ) |
| **Source Identifier** = sending MIHF ID  (Source MIHF ID TLV) |
| **Destination Identifier** = receiving MIHF ID  (Destination MIHF ID TLV) |
| Credential  (Credential TLV) |

* + - 1. MIH\_Push\_Credential response
      2. MIH\_Revoke\_Credential request
      3. MIH\_Revoke\_Credential response
      4. MIH\_Link\_Detected indication
      5. MIH\_Link\_Up indication
      6. MIH\_Link\_Down indication
      7. MIH\_Link\_Parameters\_Report indication
      8. MIH\_Link\_Going\_Down indication
      9. MIH\_Link\_Handover\_Imminent indication
      10. MIH\_Link\_Handover\_Complete indication
    1. MIH messages for command service

1. MIH protocol protection
2. Proactive authentication

# 

# 

# 

# 

# 

# (normative) Data type definition

## Derived data types

### Data type for security

***Change Table F.24 as follows:***

Table F.24—Data type for security

|  |  |  |
| --- | --- | --- |
| Data type name | Derived from | Definition |
| CERTIFICATE | OCTET\_STRING | Provides a X.509 Certificate |
| CERT\_SERIAL\_NUMBER | OCTET\_STRING | Provides X.509 formatted certificate serial number which are unique by certificate authority. |
| CERT\_STATUS | ENUMERATED | This indicates the status of the certificate being pushed or revoked  0: Not Present – indicates that certificate is not present  1: Certificate Valid – indicates that certificate is present and that the associated public key is being used to verify signatures  2: Certificate Revoked  3: Certificate Expired |
| COMPLETE\_SUBTREE | LIST (GKB\_INDEX) | The data type for the complete subtree part of a GKB. See 9.4.2.1 for the details. |
| ENCRYPTED\_KEY | OCTET(16) | This is the base data type for GROUP\_KEY\_DATA. This store a key of 16 octets encrypted with an AES key of 16 octets. |
| ID\_TYPE | ENUMERATED | The type of security association.  0: TLS-generated;  1: EAP-generated  2: GKB-generated |
| GKB\_INDEX | SEQUENCE(  NODE\_BIT\_LENGTH,  NODE\_INDEX  ) | This is the base data type for COMPLETE\_SUBTREE. |
| GROUP\_KEY\_DATA | LIST (ENCRYPTED\_KEY) | The data type for the key data part of a GKB. See 9.4.2.1 for the details. |
| GROUP\_KEY\_UPDATE\_FLAG | ENUMERATED | This indicates if the group key is to be updated  0: Key is not to be updated  1: Key is to be updated |
| GROUP\_MGT\_ACTION | ENUMERATED | This indicates a manipulation command.  0: Join the group.  1: Leave the group. |
| GROUP\_STATUS | ENUMERATED | This indicates a status of group manipulation command.  0: Join operation successful  1: Unauthorized to join the group  2: Leave operation successful  3: Unchanged |
| MIH\_SEC\_CAP | SEQUENCE(  TLS\_CAP,  EAP\_CAP,  MULTICAST\_CAP,  ) | Represents the MIH security capabilities. |
| MULTICAST\_CAP | UNSIGNED\_INT(2) | A multicast ciphersuite. Available multicast ciphersuites are defined in 9.4.6. |
| NODE\_BIT\_LENGTH | UNSIGNED\_INT(1) | This stores the bit length of the following NODE\_INDEX. |
| NODE\_INDEX | CHOICE (  UNSIGNED\_INT(1),  UNSIGNED\_INT(2),  UNSIGNED\_INT(3),  UNSIGNED\_INT(4)  ) | This stores the index of a node of the binary tree. See 9.4.2.1 for the details. |
| RESPONSE\_FLAG | ENUMERATED | This indicates if an answer is required  0: No response is needed  1: Response is needed |
| SIGNATURE | OCTET\_STRING | A digital signature data. |
| SUBGROUP\_RANGE | CHOICE(  SEQUENCE(  UNSIGNED\_INT(1),  UNSIGNED\_INT(1)),  SEQUENCE(  UNSIGNED\_INT(2),  UNSIGNED\_INT(2)),  SEQUENCE(  UNSIGNED\_INT(3),  UNSIGNED\_INT(3)),  SEQUENCE(  UNSIGNED\_INT(4),  UNSIGNED\_INT(4))) | A range of valid leaf identifiers in a complete subtree of a GKB. The first integer indicates the lowest value of the range. The second integer indicates the highest value of the range. |
| VERIFY\_GROUP\_KEY | SEQUENCE (  OCTETS(16),  OCTETS(16)  ) | The first OCTET(16) is arbitrary data, which is an input message to AES-CMAC (defined in RFC-4493). The second OCTET(16) is the MAC value for the first OCTET(16) to be verified. |

# 

# 

# 

# 

# 

# (normative) MIH protocol message code assignments

***Change Table L.2 as follows:***

Table L.2 —Type values for TLV encoding

|  |  |  |
| --- | --- | --- |
| TLV type name | TLV type value | Data Type |
| Aux Data | 79 | OCTET\_STRING |
| Configuration Data | 80 | OCTET\_STRING |
| Credential Revocation Signature | 81 | SIGNATURE |
| Credential | 82 | CERTIFICATE |
| Credential Serial Number | 83 | CERT\_SERIAL\_NUMBER |
| Credential Status | 84 | CERT\_STATUS |
| Complete Subtree | 85 | COMPLETE\_SUBTREE |
| Encrypted Credential | 86 | ENCRYPTED\_KEY |
| Group Action | 87 | GROUP\_MGT\_ACTION |
| Group Identifier | 88 | MIHF\_ID |
| Group Key Data | 89 | GROUP\_KEY\_DATA |
| Group\_Status | 90 | GROUP\_STATUS |
| Multicast Address | 91 | TRANSPORT\_ADDRESS |
| Multicast Ciphersuite | 92 | MULTICAST\_CAP |
| Multicast Link Action List | 93 | LIST(MULTICAST\_ACTION\_REQ) |
| Multicast Link Identifier | 94 | NET\_TYPE\_INC |
| Response Flag | 95 | RESPONSE\_FLAG |
| Sequence Number | 96 | OCTET\_STRING |
| Signature | 97 | SIGNATURE |
| Subgroup Range | 98 | SUBGROUP\_RANGE |
| Verify Group Key | 99 | VERIFY\_GROUP\_KEY |

# (normative) Protocol implementation conformance statement (PICS) proforma

# 

# 

# 