|  |  |
| --- | --- |
| Project | **IEEE 802.21a**  **<https://mentor.ieee.org/802.21>** |
| Title | **Suggested Remedies on 802.21a/D02** |
| DCN | 21-10-0021-00-0sec |
| Date Submitted |  |
| Source(s) | Yoshihiro Ohba (Toshia) |
| Re: |  |
| Abstract | This document contains remedies to 802.21a/D02 associated with the author’s LBa comments. |
| Purpose | Proposes changes in the current draft |
| Notice | This document has been prepared to assist the IEEE 802.21 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 802.21 may make this contribution public. |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#_blank) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock#_blank)> and in *Understanding Patent Issues During IEEE Standards Development* [http://standards.ieee.org/board/pat/faq.pdf](http://standards.ieee.org/board/pat/faq.pdf#_blank) |

|  |  |
| --- | --- |
| Ref. | 001 |
| DRAFT SECTION | 7.4.31 (MIH\_Start\_Auth), 7.4.32 (MIH\_Auth), 7.4.33 (MIH\_Termination\_Auth) |
| Modification type | DELETE |
| Text | Remove above mentioned clauses. |

|  |  |
| --- | --- |
| Ref. | 002 |
| DRAFT SECTION | 9.1.2.1 **MIH service access authentication** |
| Modification type | REPLACE |
| Text | This clause describes MIH service access authentication process. ~~The primitives are specified in Clause 7.4.~~  In this standard, it is assumed that EAP [RFC3748] or EAP Re-authentication (ERP) [RFC5296] is used as the authentication protocol with an MN as the peer and a PoS as the authenticator. An EAP server may be used as a backend server.  For the interface between an MN and a PoS, the MIH protocol is acting as an EAP lower layer. That is, at the MN, an EAP message is generated at an MIH user and then passed to the MIHF. When it reaches the PoS, the MIHF in the PoS will pass the EAP message to the ~~MIH user~~ MIHF to process it. For an EAP message from the PoS to the MN, it will also be generated by the ~~MIH user~~ MIHF ~~and passes to the MIHF~~ in the PoS. At the MN, the EAP message is passed to the ~~MIH user~~ MIHF to process. The protocol stack is illustrated in Figure 33, where it is assumed that an EAP server is employed. After a successful authentication, a master session key (MSK) is exported to the lower layer, that is, MIH layer. In 9.1.2.2, we will see that an MSK is used to further derive MIH message protection keys.    **Figure 33—Protocol Stack of Service Access Authentication (with an EAP Server)**  The authentication is divided to the following phases:  a)*Capability Discovery Phase*. In this phase, both the MN and the PoS exchange unprotected MIH messages for an MN to discover services a PoS can provide.  b)*Media Independent Service Authentication phase*. This phase is entered after the capability discovery phase and before exchanging any MIH messages other than those used in the capability discovery phase. Before starting the MIH access authentication, the MN and the PoS perform a negotiation in order to agree on a ciphersuite and other useful parameters to be used in the authentication and MIH message protection. The negotiation can be initiated either by the MN or by the PoS. Once the negotiation is completed, the MN (acting as the EAP peer) authenticates against the PoS (acting as an EAP authenticator). To achieve this, EAP is transported by MIH protocol to the current PoS, which manages the MN's communications. In order to carry out the authentication the PoS may use a backend authentication server (acting as an EAP server) to ver­ify the MN's credentials. In this document, it is assumed that the EAP methods employed can export keying material (i.e. MSK). Thus, after performing the authentication, keying material (i.e. MSK) will be shared between the MN and the PoS. Specifically, the keying material is exported to MN's and PoS' lower layer (MIH layer) and used to protect the rest of the communication. The protection mechanisms will be specified in Clause 9.2. The protected message format is specified in 8.4.3. In  order to preserve the security of the exported keying material, the exported MSK is used as a root key to derive new session keys which are used to protect the MIH PDUs. The key hierarchy is described in 9.1.2.2. Note that the authentication procedure could be based on an EAP re-authentica­tion (ERP) in order to perform a fast authentication. In this case, a rMSK is used as the root key to derive the key hierarchy.  c)*Service Access phase*. At this point, the MN is authenticated and authorized to use the MIH services, agreed and provided by the PoS. The MIH protocol is protected by using the keying material obtained in the Media Independent Service Authentication Phase. This phase is related to 9.1.2.2 for key derivation and 9.2 for protecting MIH protocol.  d)*Termination phase*. When the MN or the PoS desires to terminate the security association before the security association lifetime expires, either the MN or the PoS can request to terminate.  These phases with the MN initiated authentication and with the network initiated authentication are respec­tively illustrated in Figure 34 and Figure 35. In both figures, only the protocol interface between an EAP peer and an EAP authenticator is described. The interface with an EAP server is not included. ~~The primitives for MIH access authentication are specified in 7.4.~~  (Figures 34 and 35 remain the same.) |

|  |  |
| --- | --- |
| Ref. | 003 |
| DRAFT SECTION | 8.4.3 Protected MIH protocol frame format |
| Modification type | REPLACE |
| Text | An MIH Security (MIHS) PDU is an MIH PDU that has an MIHS header, followed by optional Source and Destination MIHF-ID TLVs, followed by an optional Session ID TLV, followed by a security TLV. Figure 29a shows a protected MIH protocol frame.  (Revise Figure 29a as follows.)    **Figure 29a—Protected MIH frame format** |

|  |  |
| --- | --- |
| Ref. | 004 |
| DRAFT SECTION | 8.4.3.1 |
| Modification type | REPLACE |
| Text | (Change the 4th paragraph as follows.)  Once a (D)TLS handshake is completed, an MIH SA is established, which is determined by the ciphersuite negotiated in the (D)TLS handshake. The structure of MIHS PDU in existence of an MIH SA is shown in Figure 29d, where the unprotected MIH ~~PDU~~ service specific TLVs ~~is~~ are carried and protected as (D)TLS application data. ~~Notice that the original MIH header, called “inner header”, is protected.~~ An MIHS header~~, called “outer header”,~~ is ~~added~~ used as MIH security header. The TLS protection can be integrity protected, encrypted, or both. If it is integrity protected, then a message integrity code (MIC) is also included in the security TLV. In this stan­dard, the message integrity code is the same as the message authentication code, for which the acronym MAC is already used for media access control.  (Change Figure 29d as follows.)    **Figure 29d—MIHS PDU in Existence of MIH SA by TLS** |

|  |  |
| --- | --- |
| Ref. | 005 |
| DRAFT SECTION | 8.4.3.2 |
| Modification type | REPLACE |
| Text | (Replace Figure 29e as follows) |

|  |  |
| --- | --- |
| Ref. | 006 |
| DRAFT SECTION | 8.4.2 Fragmentation and reassembly |
| Modification type | REPLACE |
| Text | (Also move this clause after Clause 8.4.3 “Protected MIH protocol frame format”.)  **8.4.2 Fragmentation and reassembly**  **8.4.2.1 General**  The MIH fragmentation mechanism is defined using 'M' (More Fragment) and 'FN' (Fragment Number) fields of the MIH protocol header.  An MIH message is fragmented only when MIH message is sent natively over an L2 medium such as Ethernet. The message is fragmented when the message size exceeds aFragmentationThreshold. The size of each of the fragments is the same except the last one, which may be smaller. The maximum fragment size is defined as the maximum value of aFragmentationThreshold, which shall be equal to the Maximum Transmission Unit (MTU) of the link layer that is on the path between two MIHF nodes. When the MTU of the link layer between two MIHF nodes is known, the maximum fragment size is set to the MTU. The method of determining such an MTU is outside the scope of this standard. When the MTU of the link layer between two MIHF nodes is unknown, the maximum fragment size is set to the minimum MTU of 1500 octets. When MIH message is sent using an L3 or higher layer transport, L3 takes care of any fragmentation issue, and the MIH protocol does not handle fragmentation in such cases.  Figure 29 shows the components of the fragmented MIH protocol frame. The MIH protocol payload carries a Source MIHF Identifier TLV and a Destination MIHF Identifier TLV or a Session ID TLV followed by a fragment payload.  Based on the fragment size, the fragment payload may not be aligned on a TLV boundary, i.e., TLVs other than the source MIHF identifier and destination MIHF identifier TLVs and the Session ID TLV may not be complete within the fragment payload. The fragment size may be smaller than the maximum fragment size and shall be larger than the value that can generate more than 128 fragments.  (Revise Figure 29 as follows)  **Figure 29—Fragmented MIH/MIHS PDU format**  **8.4.2.2 Fragmentation**  When an MIH message is fragmented, the fragmentation is performed within 'Transmit()' procedure in the MIH transaction protocol state machines. The MIH or MIHS header , the source MIHF identifier TLV and  destination MIHF identifier TLV or the Session ID TLV of the original message are copied to each fragment. However the 'variable payload length', 'more fragment', and 'fragment number' fields are updated accordingly for each fragment.  Variable payload length of each fragment indicates the number of octets in the MIH protocol payload of that fragment.  'More fragment' and 'fragment number' fields of each fragment are set according to the description in Table 23.  When data are to be transmitted, the number of octets in the fragment shall be determined by the fragment size and the number of octets in the multi-fragment message that have yet to be assigned to a fragment at the  instant the fragment is constructed for the first time. Once a fragment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the destination MIHF.  No retransmission by the MIH protocol (defined in 8.2) is performed for any single fragment of a multifragment message.  **8.4.2.3 Reassembly**  The destination MIHF reassembles the received fragments into an original message. Reassembly is performed outside the MIH transaction state machines. 'MsgIn' and 'MsgInAvail' variables are set only after  successful reassembly. An MIHF shall be capable of receiving fragments of arbitrary length.  The following fields are used for reassembling fragments:  — S bit  — MIH message ID  — Transaction ID  — Source MIHF identifier TLV  — Destination MIHF identifier TLV  — Session ID TLV (when Source and Destination MIHF identifiers are not present)  — More fragment  — Fragment number  When any fragment of a multi-fragment message has arrived first, the destination MIHF starts a timer referred to as ReassemblyTimer. If this ReassemblyTimer expires before all fragments have been received,  the destination MIHF discards those fragments that it has received. A duplicate fragment is discarded.  An example of an original MIH message and fragmented MIH messages is shown in Annex K. |

|  |  |
| --- | --- |
| Ref. | 007 |
| DRAFT SECTION | 8.4.3.1 |
| Modification type | REPLACE |
| Text | (Change the first three paragraphs as follows.)  This sub-clause describes securing the MIH message using TLS [RFC5246] or DTLS [RFC4347].  The transport protocol for (D)TLS is the MIH protocol. When the MIH protocol transport is reliable, TLS is used. Otherwise, DTLS is used. The transport protocol entities to be associated with a TLS session are MIHF peers and identified by MIHF identifiers. The TLS handshake takes place over the MIH protocol and as a result, an MIH SA that contains TLS master key and its child keys, TLS random values and the TLS cipher suite negotiated in the TLS handshake is established between the peers. MIH\_Auth indication message is used for TLS handshake. The detailed description about the protocol interface of using (D)TLS is provided in Clause 9.1.1. The MIH\_Auth indication message format is defined in Clause 8.6.1.11.  ~~The structure of MIHS PDU during TLS handshake is shown in Figure 29c.~~  (Remove Figure 29c.) |

|  |  |
| --- | --- |
| Ref. | 008 |
| DRAFT SECTION | 8.6.1.11 MIH\_Start\_Auth request |
| Modification type | REPLACE |
| Text | 8.6.1.11 MIH\_Auth indication  This message is used for an MIHF to perfom TLS handshake with another MIHF to establish a TLS-generated MIH SA or to communicate with another MIHF to initiate an MIH service authentication through EAP. In the former case, an AuthenticationContent shall be contained to carry a TLS record of type handshake or change cipher spec.   |  | | --- | | MIH Header field (SID=1, Opcode=3, AID=6) | | Source Identifier = sending MIHF ID  (Source MIHF ID TLV) | | Destination Identifier = receiving MIHF ID  (Destination MIHF ID TLV) | | AuthenticationContent (optional)  (Authentication TLV) | |

|  |  |
| --- | --- |
| Ref. | 009 |
| DRAFT SECTION | 3.17, 8.4.3.2, 9, 9.1.2, 9.1.2.1, 9.1.2.2, 9.1.3, 10, 10.2 |
| Modification type | REPLACE |
| Text | [1] In Clause 3.17:  “For ~~MIH service access authentication~~ an EAP-generated MIH SA, a PoS serves as an authenticator. Moreover, when ~~a service access authentication establishes~~ keys for media specific proactive authentications are generated from an EAP-generated MIH SA, a PoS provides key distribution service for media specific authenticators.”  [2] In Clause 8.4.3.2:  “An MIH security association (SA) can be established through ~~an MIH service access authentication~~ EAP. An MIH SA is specified for a pair of MIHFs. It incudes a ciphersuite used for the protections. Figure 29e shows a protected MIH PDU for an EAP-generated SA. The protection procedure is specified in 9.2.2.”  [3] In Clause 9:  “Two methods are specified for establishing protections for MIH messages. One of them is to use TLS or DTLS to establish the protections and another is to use EAP ~~as an MIH service access authentication~~ to establish MIH security associations (SAs).”  [4]In Clause 9.1.2:  **“9.1.2 ~~Key establishment through an MIH access authentication~~ Protections established through EAP**  If MIH service is subscription based and provided by a service provider, then a MIH service access authenti­cation may be needed to authorize the service to a mobile node. In this case, an access authentication may include a key establishment subroutine to allow a PoS to obtain a master session key so that MIH security associations can be established through the master session key between the MN and the PoS. The following sub-clauses describe authentication and key establishment procedure for an EAP-generated MIH SA.  [5] In Clause 9.1.2.1:  **“9.1.2.1 MIH service access authentication**  This clause describes MIH service access authentication process through EAP. The primitives are specified in Clause 7.4.”  [6] IN Clause 9.1.2.2  “Upon a successful MIH service access authentication through EAP, the authenticator, PoS, obtains a master session key (MSK) or a re-authentication master session key (rMSK).”  [7] In Clause 9.1.3:  “Depending on how the MIH specific protection keys are established, an MIH SA can be established through a (D)TLS handshake or an ~~MIH service access~~ EAP authentication. When an MIH SA is established via a (D)TLS handshake as specified in 9.1.1, the TLS master key and the keys derived from the master key, all the TLS parameters, and TLS ciphersuite negotiated in the TLS handshake form an MIH SA.  When an MIH SA is established through an EAP ~~method~~ authentication with key establishment, the SA consists of the keys, the key derivation functions, and the ciphersuite. The key derivation functions, encryption algorithms, and integrity algorithms are specified in Table 24.  [8] In Clause 10:  “The second option is to bundle the media access proactive authentication to MIH service access authentication using EAP.”  [9] In Clause 10.2:  **“10.2 Bundle media access autehntication to MIH service access authentication using EAP**  This clause specifies an optimization option by bundling the media access authentication with a MIH service access authentication using EAP when the trust relationship between media specific network access provider and the MIH service provider allows. In this case, at the end of a successful service access authentication, a PoS will derive not only keys for MIH message protection as defined in 9.1.2.2 but also a key called media specific root key (MSRK). This key will be further used to derive a key or keys called media specific pairwise master keys (MSPMKs) to be used by a target PoA or PoAs. The bundling option is purely for optimization and is optional to use.” |