|  |  |
| --- | --- |
| Project | **IEEE 802.21a** **<https://mentor.ieee.org/802.21>** |
| Title | **Proactive Authentication and MIH Security**  |
| DCN | **21-09-0102-00-0Sec** |
| Date Submitted | **July 05, 2009** |
| Source(s) | Subir Das, Ashutosh Dutta (Telcordia), Toshikazu Kodama(Toshiba) |
| Re: | IEEE 802.21 Session #33 in San Francisco, USA |
| Abstract | This document elaborates 21-09-0066-00-0Sec proposal: Proactive Authentication techniques and MIH protocol level security mechanisms  |
| Purpose | Specific functional requirements need to be developed for the IEEE 802.21 devices to provide the necessary reliability, availability, and interoperability of communications with different operator networks. In addition, guidelines for using MIH protocol need to be developed so that vendors and operators can better understand the issues, pros, and cons of implementing IEEE 802.21 for supporting various mobility handover scenarios. |
| 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#6.3) <[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)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> |

# Introduction

## Scope

 The scope of this document is to propose a solution based on the requirements described in 21-08-0172-02-0sec-mih-security-technical-report

## Purpose

The purpose of this document is to describe the proposed approaches for proactive authentication and MIH protocol level security. In particular, this document addresses the following functionalities:

|  |  |  |
| --- | --- | --- |
| **Work Item #** | **Supported Functionality** | **Note** |
| 1 | Proactive Re-Authentication | Yes |
| 1 | EAP Pre-authentication | Yes |
| 1 | Key Hierarchy and Derivation 1 | Yes |
| 1 | Higher-Layer Transport for MN-CA, MN-SA and SA-CA signaling | Yes |
| 1 | Link-Layer Transport for MN-SA signaling | Yes |
| 1 | Authenticator Discovery Mechanism | No\* |
| 1 | Context Binding Mechanism | Yes |
| 2 | Access Authentication | Yes |
| 2 | MIH-Specific Authentication | Yes |
| 2 | Key Hierarchy and Derivation 2 | Yes |
| 2 | MIH-Specific Protection | Yes |
| 2 | Protection by MIH Transport Protocol | No |
| 2 | Visited Domain Access | No\* |

## Terminologies

**EAP:** Extensible Authentication Protocol

 **ERP** : EAP Re-Authentication Protocol

**SA :** Serving Authenticator

**CA :** Candidate Authenticator

## Definitions

**Authentication Process :** The cryptographic operations and supporting data frames that perform the authentication

**Media Specific Authenticator and Key Holder (MSA-KH):** Media specific authenticator and key holder is an entity that facilitates authentication of other entities attached to the other end a link specific to a media.

**Media Independent Authenticator and Key Holder (MIA-KH):** Media Independent authenticator and key holder is an entity that interacts with MSA-KH and facilitates proactive authentication of other entities attached to the other end of a link of a MSA-KH.

**Proactive Authentication :** An authentication process that is performed between MIA-KH and other entities attached to the other end of a link of a MSA-KH. This process occurs when the other entities intend to perform a handover to another link.

**Serving MIA-KH :** The MIA\_KH that is currently serving to a mobile node which is attached to an access network

**Candidate MIA-KH:** The MIA-KH that is serving to an access network which is in the mobile node’s list of potential candidate access networks.

**MIH Security Association (SA):** An MIH SA is the security association between the peer MIH entities

## References

[RFC4748] H. Levkowetz, Ed. and et al, Extensible Authentication Protocol (EAP), RFC 3748

[RFC5296] V. Narayan and L. Dondeti, “EAP Extensions for EAP Re-authentication Protocol (ERP)” RFC 5296

[RFC4306] C. Kaufman, Ed, “Internet Key Exchange (IKEv2) Protocol:”, RFC 4306

[RFC5246] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2” , RFC 5246

[RFC4347] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security”, RFC 4347

[RFC5295] J. Saloway, et al, “Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)” RFC 5295

[IEEE802.21] IEEE P802.21 Std-2008, IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover Services.

# Proactive Authentication

Proactive authentication is a process by which an entity can perform a-priori network access authentication with a media independent authenticator and key holder (MIA-KH) that is serving a candidate network. The entity performs such authentication in anticipation of handover to the neighboring networks. Proactive authentication can be performed in two ways: i) Direct Proactive Authentication (Figure 1 ) whereby the authentication signaling is transparent to the serving MIA-KH and ii) Indirect Proactive Authentication (Figure 2 ) whereby the serving MIA-KH is aware of the authentication signaling. In each case either EAP (Extensible Authentication Protocol) [ RFC4798 ] or ERP ( EAP Re-Authentication Protocol) [RFC5296 ] can be used as authentication protocol.

Figure 1 and Figure 2 show the relationship between different functional entities and their involvement during proactive authentication signaling. For direct proactive authentication, mobile

Figure 1: Direct Proactive Authentication [Ref: Technical Report]

Figure 2: Indirect Proactive Authentication [Ref: Technical Report]

node directly communicates with the candidate MIA-KH (Figure 1) and for indirect proactive authentication, mobile node first communicates with the serving MIA-KH. . Serving MIA-KH then communicates with the candidate MIA\_KH on behalf of mobile node.

## Proactive Authentication Architecture

Figure 3 and Figure 4 depict two example logical architectures for proactive authentication. The Media Independent Authenticator and Key Holder (MIA-KH) is the entity that facilities the authentication prior to handover to candidate networks. In this architecture, the authentication functionalities are added within Media Independent Handover Function (MIHF) and the new entity is called as enhanced POS (e.g., PoS+).

Figure 3: Proactive Authentication Architecture: Example A

The Media Specific Authenticator and key holder (MSA-KH) is responsible for authenticating devices for access to a specific access network and the proposed architecture assumes no change of such existing mechanisms. The difference between and is that in , two access networks are managed by one MIA-KH and hence there exists one PoS while in , each access network has their own MIA-KH and hence two separate PoSes are required. also has one additional interface called RP5 per MIH communication model [IEEE Std 802.21.2008).

Figure 4: Proactive Authentication Architecture: Example B

This proposal supports both direct and indirect proactive authentication including network-initiated and mobile-initiated procedures. The sequence of operation involves:

* MN attaches to the access network with access specific authentication procedures
* During handover preparation stage, MN discovers the candidate authenticators
* Depending upon the reachability of the media independent authenticator, MN performs either direct or indirect proactive authentication using RP1 interface
* Once media independent authentication is successfully performed, the media specific keys are either pushed to or pulled from MSA-KH. Interface\_MIA-KH\_MSA-KH is used to perform this operation.
* MN executes the handover by performing the media specific secure association (e.g., 4-way handshake for 802.11) and attaches to one of the candidate networks known as target network.
* After connection establishment, MN reregisters with the PoS

Following assumptions are made throughout the rest of this section:

**Assumptions**

* Authenticator is a MIH PoS
* MIH protocol is used for carrying EAP and ERP
* MIHF-ID of MN is used as the media-independent identity of the MN
* MIHF-ID of authenticator is used as the media-independent identity of the authenticator
* Media Independent Authenticator holds MSK (Master Session Key) or rMSK (Re-authentication MSK) generated by EAP
* MSK or rMSK is used for deriving media-independent pair-wise master key (MI-PMK)
* When MN hands over to the target MSA-KH and it has a media-specific PMK (MS-PMK) derived from an MI-PMK for the target MSA-KH, it runs media-specific secure association using the MS-PMK.

### Proactive Authentication using EAP

In this section we describe the procedures using EAP as proactive authentication protocol.

#### Direct Proactive Authentication

In this scenario, MN directly performs the authentication with the media independent candidate authenticator. The assumption here is that MN either knows the candidate authenticator or discovers through MIH Information Service. Candidate authenticator must be reachable directly from the MN via an IP link.

#### Call Flows

Figure 5 describes the message flows between MN and Media independent authenticator for network initiated direct proactive authentication. It covers both example A and example B architectures. For example B architecture, Serving MIA-KH and candidate MIA-KH authenticators are two separate entities and they use interface RP5 to communicate each other. Two new MIH message types : i) MIH Pro\_Auth Request (EAP) message and ii) MIH Pro\_Auth Response message are proposed for carrying EAP messages over MIH. The first MIH Pro\_Auth request message is initiated by the network that carries the EAP\_Identity\_Request message which is followed by a MIH Pro\_Auth Response message from the MN. The PoA-Link-Addr-List and MN-Link-Addr-list are necessary in the final request/response message in order for securely binding the keys with the link layer identities.

Figure 5: Network Initiated Direct Proactive Authentication (EAP)

Figure 6 describes the message flows between MN and Media independent authenticator for mobile initiated proactive authentication. The important difference from the previous one is that the trigger comes from the MN that generates the MIH Pro\_Auth Request and sends it to the candidate authenticator directly. The remaining call flows are similar to network initiated direct proactive authentication as described in Figure 5.

Figure 6: Mobile Initiated Direct Proactive Authentication (EAP)

#### Indirect Proactive Authentication

In this scenario, MN cannot perform the authentication directly with the media independent candidate authenticator. The serving authenticator takes part in forwarding the messages either to the MN (in case of network initiated authentication) or candidate MIA-KH (in case of mobile initiated authentication). The assumption here is that MN either knows the candidate authenticator or discovers through MIH Information Service but MN cannot reach to the candidate authenticator directly via an IP link.

#### Call Flows

Figure 7 describes the message flows between MN and Media independent authenticators for network initiated indirect proactive authentication. As described earlier, it covers both example A and example B architectures and for example B architecture, Serving MIA-KH and candidate MIA-KH entities are two separate entities and they use interface RP5 to communicate each other. The first MIH Pro\_Auth request message is initiated by the candidate MIA-KH and sent it to serving MIA-KH which is then forwarded to the MN.

**Figure 7: Network Initiated Indirect Proactive Authentication (EAP)**

MN generates MIH Pro\_Auth Response message and subsequent EAP messages are carried over request and response messages. The PoA-Link-Addr-List and MN-Link-Addr-list are used for securely binding the key with the link layer identities.

Figure 8: Mobile Initiated Indirect Proactive Authentication (EAP)

Figure 8 depicts the mobile initiated indirect proactive authentication in which trigger comes from the MN that generates the MIH Pro\_Auth Request message and sends it to the serving MIA-KH. Candidate MIA-KH receives this message from serving MIA-KH and sends the MIH Pro\_Auth Response message to the serving MIA-KH which is then forwarded to the MN. The remaining call flows are similar to network initiated indirect proactive authentication as described in .

### Proactive Authentication using ERP

In this section we describe the procedures using ERP as proactive authentication protocol.

#### Direct Proactive Authentication

In this scenario, MN directly performs the authentication with the media independent candidate authenticator. The assumption here is that MN either knows the candidate authenticator or discovers through MIH Information Service. Candidate authenticator must be reachable directly from the MN via an IP link.

#### Call Flows

Figure 8 describes the message flows between MN and Media independent authenticator for network initiated direct proactive authentication. It covers both example A and example B architectures and for example B architecture, Serving MIA-KH and candidate MIA-KH authenticators are two separate entities and they use interface RP5 to communicate each other. A new MIH message type called MIH Pro\_Auth Indication is proposed for initiating ERP messages exchanges over MIH. This triggers the MN to generate the MIH Pro\_Auth request (ERP) message. The PoA-Link-Addr-List and MN-Link-Addr-list are necessary in the request/response message in order for securely binding the keys with the link layer identities.

Figure 9: Network Initiated Direct Proactive Authentication (ERP)

 describes the message flows between MN and Media independent authenticator for mobile initiated proactive authentication. The main difference from the previous one is that the trigger comes from the MN that generates the MIH Pro\_Auth Request and sends it to the candidate authenticator directly. Finally candidate MIA-KH sends the MIH Pro\_Auth Response with the authentication success or failure.

Figure 10: Mobile Initiated Direct Proactive Authentication (ERP)

#### Indirect Proactive Authentication

In this scenario, MN cannot perform the authentication directly with the media independent candidate authenticator. The serving authenticator takes part in forwarding the messages either to the MN (in case of network initiated authentication) or candidate MIA-KH (in case of mobile initiated authentication). The assumption here is that MN either knows the candidate authenticator or discovers through MIH Information Service but MN cannot reach to the candidate authenticator directly via an IP link.

#### Call Flows

Figure 11 describes the message flows between MN and Media independent authenticators for network initiated indirect proactive authentication. The first MIH Pro\_Auth request message is initiated by the candidate MIA-KH and sent it to serving MIA-KH which is then forwarded to the MN. MN generates MIH Pro\_Auth Response message and subsequent EAP messages are carried over request and response messages. The PoA-Link-Addr-List and MN-Link-Addr-list are used for securely binding the key with the link layer identities.

Figure 11: Network Initiated Indirect Proactive Authentication (ERP)

Figure 12 shows the mobile initiated indirect proactive authentication in which trigger comes from the MN that generates the MIH Pro\_Auth Request message with ERP and sends it to the serving MIA-KH. Candidate MIA-KH receives this message from serving MIA-KH and sends the MIH Pro\_Auth Response message with ERP to the serving MIA-KH which is then forwarded to the MN. The PoA-Link-Addr-List and MN-Link-Addr-list are necessary in the request/response message for securely binding the keys with the link layer identities.

Figure 12: Mobile Initiated Indirect Proactive Authentication (ERP)

## Attachment to Target Authenticator

After the authentication is performed and mobile node decides to execute the handover, it chooses one of the candidate networks and switch to that access network. This candidate network becomes the target network and the authenticator that serves this access network is called the target media specific authenticator and key holder (MSA-KH). Mobile node then performs the media specific secure association (SA) assuming that the target MSA has obtained the right set of keys from the target media independent authenticator and key holder(MIA-KH) for the mobile node.

### Call Flows

Figure 13 depicts the call flows between MN, target MSA-KH and target MIA-KH. Once the proactive authentication is successfully performed, MIA-KH generates per mobile node media specific keys that can either be pushed to MSA-KH or pulled by the MSA-KH. Once the keys are available at the MSA-KH, mobile node can perform the media specific security association as soon as it switches to the network without needing to perform a full authentication. Once the secure association is successful and an IP connection is established, MN registers with the MIA-KH in order for the MIA-KH to correctly register the mobile node as its serving node.

Figure 13: Attachment to Target MSA-KH (EAP/ERP)

## Proactive Authentication Termination

The purpose of the proactive authentication termination is to ensure that mobile node and candidate/target/serving authenticator terminates the session and corresponding state machines are synchronized. At this point MI-PMK and MS-PMK are either cached or deleted.

### Direct Proactive Authentication Termination

Direct proactive authentication termination allows both network and mobile node to directly terminate the authentication states.

#### Call Flows

Figure 14 shows the call flows for both network initiated and mobile initiated termination procedures. The purpose of including the integrity check is to verify the authenticity of the termination request and response.

Figure 14: Direct Proactive Authentication Termination(EAP/ERP)

### Indirect Proactive Authentication Termination

Indirect proactive authentication termination allows both network and mobile node to terminate the authentication states via the serving MIA-KH.

#### Call Flows

Figure 15 shows the call flows for both network initiated and mobile initiated termination procedures. The purpose of including the integrity check is to verify the authenticity of the termination request and response. In both cases, serving MIA-KH forwards the termination request either to the MN or to the candidate MIA-KH.

Figure 15: Indirect Proactive Authentication Termination

# Primitives

In this section, we outline the new primitives and corresponding parameters that are required to enable the proactive authentication [IEEE802.21].

## Event Primitives

Table 1 describes the list of link events.

Table : List of Event Primitives

|  |  |  |
| --- | --- | --- |
| Link Events  | Link event type | Description |
| Link\_Pro\_Auth\_Key\_Install | Local  | This event indicates that a key distribution request for a MS-PMK has been generated by a media-specific access function. |

### Link\_Pro\_Auth\_Key\_Install.Indication

#### Function

#### This notification is delivered when a layer 2 connection establishment is attempted on the specified link interface Semantics of service primitive

 Link\_Pro\_Auth\_key\_Install.indication (

 LinkIdentifier

 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Link Identifier | LINK\_TUPLE\_ID | Identifier of the link associated with the event. |

#### When generated

This notification is generated when a layer 2 connection is established for the specified link interface or after the completion of a successful proactive authentication.

#### Effect of Receipt

The MIHF shall pass this link notification to the MIH User(s) that has subscribed for this notification. The MIH User(s) either push or pull the media specific keys.

## MIH Events

Table 2 describes the list of MIH events

Table : List of MIH Events

|  |  |  |
| --- | --- | --- |
| MIH Events  | MIH event type | Description |
| MIH\_Pro\_Auth\_ Result | Local  | Success or failure of proactive authentication (EAP or ERP) |

### MIH\_Pro\_Auth\_ Result.indication

#### Function

The MIH\_Pro\_Auth\_ Result is sent to local MIHF users to notify them of a local event, or is the result of the receipt of an MIH\_Pro\_Auth\_Request message to indicate to the remote MIHF users, who have subscribed to this remote event.

#### Semantics of service primitive

MIH\_Pro\_Auth\_ Result.indication (
 SourceIdentifier

 MobilenodeIdentifier

 CandidateIdentifier(Optional)

 MobileLinkIdentifiers

 PoALinkIdentifiers

 Status

 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Source Identifier | MIHF\_ID | This identifies the invoker of this primitive which can be either the local MIHF or a remote MIHF. |
| Mobile Node Identifier | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (optional) | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Mobile Link Identifiers | LIST(LINK\_ID) | Link identifiers of MN. |
| PoA Link Identifiers | LIST(LINK\_ID) | Link identifiers of PoA. |
| Status  | STATUS | Status of (Success or failure) of proactive authentication |

#### When generated

This primitive is generated by the local or remote MIHF when an MIH\_Pro\_Auth request message is received.

#### Effect on receipt

The MIHF shall pass this link notification to the MIH User(s) that has subscribed for this notification.

## Command Primitives

##  describes the MIH commands.

Table : List of MIH Commands

|  |  |  |
| --- | --- | --- |
| MIH Commands | MIH Command type | Description |
| MIH\_Pro\_Auth\_Start | Remote | Starting Proactive authentication |
| MIH\_Pro\_Auth\_Termination | Remote | Terminating proactive authentication |
| MIH\_Pro\_Auth\_key\_Install | Local | Installing proactive authentication keys |

### MIH\_Pro\_Auth\_Start.request

#### Function

The primitive is invoked by an MIH User to indicate to a local MIHF or a peer MIH User about its intent of proactive authentication

#### Semantics of service primitive

MIH\_Pro\_Auth\_Start.request (
 DestinationIdentifier,
 MobileNodeIdentifier(Optional)
 CandidateIdentifier(Optional)
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Destination Identifier | MIHF\_ID | This identifies a remote MIHF that will be the destination of this request.  |
| Mobile Node Identifier (Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional) | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |

#### When generated

This primitive is invoked by an MIH User to communicate with a local MIHF or remote MIH User about its intent of proactive authentication.

#### Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH\_Pro\_Auth request message to the remote MIHF identified by the Destination Identifier. The remote MIHF forwards the request as an indication to the MIH User.

### MIH\_Pro\_Auth\_Start.indication

#### Function

This primitive is used by an MIHF to indicate to an MIH User that an MIH\_Pro\_Auth request message was received from a remote MIHF.

#### Semantics of service primitive

MIH\_Pro\_Auth\_Start.indication (
 SourceIdentifier,

 MobileNodeIdentifier
 CandidateIdentifier
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Source Identifier | MIHF\_ID | This identifies the invoker of this primitive which can be either the local MIHF or a remote MIHF.  |
| Mobile Node Identifier (Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional)  | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |

#### When generated

This primitive is generated by an MIHF on receiving an MIH\_Pro\_Auth request message from a peer MIHF.

#### Effect on receipt

An MIH User receiving this indication shall invoke an MIH\_Pro\_Auth\_Start.response primitive towards the remote MIHF indicated by the Source Identifier in the request message.

### MIH\_Pro\_Auth\_Start.response

#### Function

This primitive is used by the MIHF on an MN to respond to an MIH\_Pro\_Auth request message from a remote MIHF in the network.

#### Semantics of service primitive

MIH\_Pro\_Auth\_start.response (
 DestinationIdentifier,
 MobileNodeIdentifier (Optional),
 CandidateIdentifier (optional),
 Status,
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Destination Identifier | MIHF\_ID | This identifies a remote MIHF that will be the destination of this request.  |
| Mobile Node Identifier | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier  | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Status  | STATUS | Status of (Success or failure) of proactive authentication |

#### When generated

The remote MIH User invokes this primitive in response to an MIH\_Pro\_Auth\_Start.indication from its MIHF.

#### Effect on receipt

The MIHF sends an MIH\_Pro\_Auth response message to the peer MIHF as indicated in the Destination Identifier.

### MIH\_Pro\_Auth\_Start.confirm

#### Function

This primitive is used by the MIHF to confirm that an MIH\_Pro\_Auth response message was received from a peer MIHF.

#### Semantics of service primitive

MIH\_Pro\_Auth\_Start.confirm (
 SourceIdentifier,
 MobileNodeIdentifier(Optional),
 CandidateIdentifier(Optional),
 Status,
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Source Identifier | MIHF\_ID | This identifies the invoker of this primitive which can be either the local MIHF or a remote MIHF. |
| Mobile Node Identifier (Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional)  | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Status  | STATUS | Status of (Success or failure) of proactive authentication |

#### When generated

This primitive is generated by the MIHF on receiving an MIH\_Pro\_Auth response message from a peer MIHF.

#### Effect on receipt

On receiving the primitive the entity that originally initiated the proactive authentication request decides to carry out the proactive authentication or abort it based on the primitive. However, if Status does not indicate “Success or Failure”, the recipient ignores any other returned values and, instead, performs appropriate error handling.

### MIH\_Pro\_Auth\_Termination.request

#### Function

The primitive is invoked by an MIH User to indicate to a peer MIH User about the termination of proactive authentication.

* Semantics of service primitive

MIH\_Pro\_Auth\_Termination.request (
 DestinationIdentifier,
 MobileNodeIdentifier(Optional)
 CandidateIdnetifier(Optional)

 KeyCache
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Destination Identifier | MIHF\_ID | This identifies a remote MIHF that will be the destination of this request.  |
| Mobile Node Identifier (Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional) | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Key Cache  | Boolean  | This indicates if a key caching is requested or not: True: key caching required; False: key caching not required ; |

#### When generated

This primitive is invoked by an MIH User to communicate with a remote MIH User about the termination of proactive authentication.

#### Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH\_Pro\_Auth\_Termination request message to the remote MIHF identified by the Destination Identifier. The remote MIHF forwards the request as an indication to the MIH User.

### MIH\_Pro\_Auth\_Termination.indication

#### Function

This primitive is used by an MIHF to indicate to an MIH User that an MIH\_Pro\_Auth\_Termination request message was received from a remote MIHF.

#### Semantics of service primitive

MIH\_Pro\_Auth\_Termination.indication (
 SourceIdentifier,

 MobileNodeIdentifier(Optional)
 CandidateIdnetifier(Optional)

 KeyCache
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Source Identifier | MIHF\_ID | This identifies the invoker of this primitive which can be either the local MIHF or a remote MIHF.  |
| Mobile Node Identifier(Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional) | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Key Cache  | Boolean  | This indicates if a key caching is requested or not: True: key caching required; False: key caching not required ; |

#### When generated

This primitive is generated by an MIHF on receiving an MIH\_Pro\_Auth\_Termination request message from a peer MIHF.

#### Effect on receipt

An MIH User receiving this indication shall generate a MIH\_Pro\_Auth\_Termination.response towards the remote MIHF indicated by the Source Identifier in the request message.

### MIH\_Pro\_Auth\_Termination.response

#### Function

This primitive is used by the MIHF on an MN to respond to an MIH\_Pro\_Auth request message from a remote MIHF in the network.

#### Semantics of service primitive

MIH\_Pro\_Auth\_Termination.response (
 DestinationIdentifier,
 MobileNodeIdentifier(optional),
 CandidateIdentifier(Optional),
 Status,
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Destination Identifier | MIHF\_ID | This identifies a remote MIHF that will be the destination of this request.  |
| Mobile Node Identifier (Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional) | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Status  | STATUS | Status of (Success or failure) of proactive authentication |

#### When generated

The remote MIH User invokes this primitive in response to an MIH\_Pro\_Auth\_Termination.indication from its MIHF.

#### Effect on receipt

The MIHF sends an MIH\_Pro\_Auth response message to the peer MIHF as indicated in the Destination Identifier.

### MIH\_Pro\_Auth\_Termination.confirm

#### Function

This primitive is used by the MIHF to confirm that an MIH\_Pro\_Termination response message was received from a peer MIHF.

#### Semantics of service primitive

MIH\_Pro\_Auth\_Termination.confirm (
 SourceIdentifier,
 MobileNodeIdentifier(Optional),
 CandidateIdentifier(Optional),
 Status,
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Source Identifier | MIHF\_ID | This identifies the invoker of this primitive which can be either the local MIHF or a remote MIHF. |
| Mobile Node Identifier (Optional) | MIHF\_ID | This identifies a mobile node MIHF that will be the destination of this request.  |
| Candidate Identifier (Optional) | MIHF\_ID | This identifies a remote MIHF that will be the future destination. |
| Status  | STATUS | Status of (Success or failure) of proactive authentication |

#### When generated

This primitive is generated by the MIHF on receiving an MIH\_Pro\_Auth response message from a peer MIHF.

#### Effect on receipt

On receiving the primitive the entity that originally initiated the proactive authentication request decides to terminate the proactive authentication or abort it based on the primitive. However, if Status does not indicate “Success or Failure”, the recipient ignores any other returned values and, instead, performs appropriate error handling.

### LInk\_Pro\_Auth\_Key\_Install.request

#### Function

The primitive is invoked by an MIH User to indicate a local MIHF to install the media specific keys to media specific authenticator

#### Semantics of service primitive

Link\_Pro\_Auth\_Key\_Install.request (
 DestinationLinkIdentifierwithKeys

 )

Parameters:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Destination Link Identifier with keys | LIST(SEQUENCE(LINK\_TUPLE\_ID, MS-PMK) | Identifier of the link(s) for which keys need to be installed and the media specific keys  |

#### When generated

This primitive is invoked by an MIHF to indicate a local layers to install the keys to media specific authenticator.

#### Effect on receipt

Upon receipt of this primitive, the lower layer generates a Link\_Pro\_Auth\_Key\_Install.comfirm primitive to MIHF.

### Link\_Pro\_Auth\_Key\_Install.confirm

#### Function

This primitive is used by the MIHF to confirm that a Link\_Pro\_Auth\_Key\_Install.request was received.

#### Semantics of service primitive

Link\_Pro\_Auth\_Key\_Install.confirm (
 Status,
 )

Parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Status  | STATUS  | Status of key install request received |

#### When generated

This primitive is generated by the lower layer on receiving an Link\_Pro\_Auth\_Key\_Install.request from MIHF.

#### Effect on receipt

On receiving the primitive the MIHF decides to keep the proactive authentication states or abort it based on the status.

## Key Generation Algorithm

* KDF : Key derivation function specified in RFC 5246
	+ The default KDF (i.e., IKEv2 PRF+ with based on HMAC-SHA-256 ) is used unless explicitly negotiated between peer MIHFs
* MI-PMK = KDF(MK, “MI-PMK” | RAND\_P | RAND\_A | length)
	+ Length of MI-PMK is 64 octets
	+ MK (Master Key): MSK or rMSK
	+ RAND\_P: A random number generated by peer
	+ RAND\_A: A random number generated by authenticator
* MS-PMK =KDF(MI-PMK, “MS-PMK” | MN\_LINK\_ID | POA\_LINK\_ID | length)
	+ Length of MS\_PMK depends on each media. In the case of 802.11, Length=32.
	+ MN\_LINK\_ID: Link identifier of MN, encoded as LINK\_ID data type
	+ POA\_LINK\_ID: Link identifier of media-specific authenticator, encoded as LINK\_ID data type
* PRF+ : PRF+ key expansion specified in IKEv2 [RFC4306]
* PRF+ (K,S) = T1 | T2 | T3 | T4 | ...
* Where:
* ‘|’ means concatenation
* T1 = PRF (K, S | 0x01)
* T2 = PRF (K, T1 | S | 0x02)
* T3 = PRF (K, T2 | S | 0x03)
* T4 = PRF (K, T3 | S | 0x04)
* …
* continuing as needed to compute the required length of key material. The default PRF is taken as HMAC-SHA-256 [SHA256]. Since PRF+ is only defined for 255 iterations it may produce up to 8160 octets of key material.

## Key Distribution Mechanism

**TBD**

**Note:** Authors are seeking Task Group’s guidance here. It would be good to have some discussions since this would require media specific interactions. Should this require a standard or can it be left as implementation or deployment specific choice?

# Securing MIH Protocol Messages

This section proposes the MIH protocol message security using two use cases: i) when MIH service access control is applied and ii) when MIH service access control is not applied. We propose to use the existing protocols for authentication and key management that will greatly reduce the risk of introducing security flaws.

## Proposed Approach

Our proposed approach consists of following steps:

* + Use EAP over MIH protocol for MIH service authentication. In this case PoS acts as an authenticator and also runs AAA client
	+ Use TLS [RFC5246] or DTLS [RFC4347] for authentication, key establishment and ciphering. TLS handshake is carried out over MIH protocol and an MIH SA is established between two MIHF peers. (D)TLS will provide cipher suites negotiation. Once MIH SA is defined within MIH protocol, there is no need to have MIH transport level security

## MIH Security Use case with access control

We assume that MIH service access control is applied through an access controller. The access control is applied through an access authentication with the MIH service provider through an Authentication Server (AS), e.g., an EAP Server or an AAA server. Upon a successful authentication, the MN is authorized to access the MIH services through PoS’es . The access authentication includes a key establishment procedure so that keys are established between the MN and the Authentication Server. When proactive authentication is supported, proactive authentication and authentication for MIH services use the same AS.

### Call Flows

Figure 16 describes the call flows when MIH service access control is applied and then MIH SA is bootstrapped after a successful authentication.

Figure 16: MIH Security with Access Control

## MIH Security Use Case without Access Control

We assume that the MIH service access control is not applied through any access controller. The mutual authentication may be based on a pre-shared key or a trusted third party like certificate authority. The authentication is MIH specific. The MN and the PoS will conduct a mutual authentication and key establishment of MIH specific keys.

### Call flows

Figure 17 describes the MIH security call flows with any access control to MIH services.

Figure 17: MIH Security without Access Control

## Security Capability Discovery

The following security-related capability is defined for MIH capability discovery.

* Data Type: MIH\_SEC\_CAP
* Derived from BITMAP(16)
	+ Bit 0: Direct pre-authentication
	+ Bit 1: Indirect pre-authentication
	+ Bit 2: Direct re-authentication
	+ Bit 3: Indirect re-authentication
	+ Bit 4: MIH SA with access control
	+ Bit 5: MIH SA without access control
	+ Bit 6-15: Reserved

The following parameter is added to MIH\_Capability\_Discover.{request,response} primitives.

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| SupportedSecurityCapList | MIH\_SEC\_CAP | List of supported MIH security capabilities on the local MIHF. |

The following parameter is added to MIH\_Capability\_Discover.{indication,confirm} primitives.

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| SupportedSecurityCapList | MIH\_SEC\_CAP | List of supported MIH security capabilities on the remote MIHF. |

* 1. **Key Hierarchy and Derivation**

The key hierarchy for MIH SA is illustrated in **Figure 18**. When access control for MIH services is enabled, the (D)TLS PSK is derived from MSK or rMSK that is generated between an MIHF acting an EAP peer and an EAP server and transported to the other MIHF acting an EAP authenticator. The derived PSK becomes a (D)TLS credential for (D)TLS handshake to mutually authenticate the MIHF peers and establish (D)TLS key material for protecting MIH messages using (D)TLS. The PSK is derived as follows.

PSK = KDF(MK, “TLS-PSK” | RAND\_P | RAND\_A | length)

MK (Master Key) is either MSK or rMSK depending on whether EAP or ERP is used for the access authentication. RAND\_P is a random number generated by the MIHF acting an EAP peer. RAND\_A is a random number generated by the MIHF acting an EAP authenticator. The length is the length of the PSK in octets. The default length is 64. KDF is the key derivation function defined in [RFC5295]. The default KDF, i.e., IKEv2 PRF+ with based on HMAC-SHA-256, is used unless explicitly negotiated between the peer MIHFs.

When access control for MIH services is not enabled (Use Case 2), pre-configured TLS credentials are used for (D)TLS handshake to mutually authenticate the MIHF peers and establish (D)TLS key material for protecting MIH messages using (D)TLS.

**Figure 18: Key Hierarchy for MIH SA**

* 1. **MIH Protocol Extensions**

TLS or DTLS is used for securing the MIH protocol. The transport protocol for (D)TLS in this case is the MIH protocol itself. 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. Therefore, the transport address of an MIHF can change over the lifetime of a TLS session as long as the mapping between the transport address and MIHF identifier of an MIHF is maintained. The following subsections describe extensions to the MIH protocol for use of (D)TLS.

* + 1. **TLS TLV**

TLS (Transport Layer Security) TLV is a new TLV of type OCTET\_STRING carrying a (D)TLS message. Once an MIH SA is established, the entire raw MIH PDU excluding Source and Destination MIHF Identifier TLVs, must be protected with the TLS key material of the MIH SA and carried in the payload of the TLS TLV as the TLS application data.

* + 1. **Session ID TLV**

Session ID (Identifier) TLV is a new TLV of type OCTET\_STRING carrying a (D)TLS session identifier [RFC 5246] that is assigned as a result of a TLS handshake.

* + 1. **Security Capability TLV**

Security Capability TLV is a new TLV of type MIH\_SEC\_CAP carrying security capabilities of an MIHF. This TLV is carried in MIH\_Capability\_Discover request and response messages.

* + 1. **MIH Security PDU**

An MIH Security (MIHS) PDU is an MIH PDU that has an MIHS header, followed by followed by optional Source and Destination MIHF-ID TLVs, followed by an optional Session ID TLV, followed by a TLS TLV. An MIHS header is an MIH protocol header containing the following information.

* Version: the version of MIH protocol
* Ack-Req: 0
* Ack-Rsp: 0
* UIR: 0
* M:0
* FN:0
* SID: 5 (Security Service)
* Opcode: 2 (Indication)
* TID: 0

A Session ID TLV is associated with the pair of MIHFs associated with the MIH SA. Therefore, Source and Destination MIHF Identifier TLVs do not need to be carried in an MIHS PDU in existence of an MIH SA, and a Session ID TLV is carried instead. Source and Destination MIHF Identifier TLVs are carried in a MIHS PDU in absence of an MIH SA or when the sender’s transport address has been changed. In the latter case, the mapping between the sender’s transport address and the MIHF identifier shall be updated, and an MIH Registration request or response message may be contained in the TLS TLV.

The structure of MIHS PDU during TLS handshake is shown in

**Figure 19**. The structure of MIHS PDU in existence of an MIH SA is shown in . The structure of MIHS PDU upon Transport Address Change is shown in .

**Figure 19: MIHS PDU during TLS handshake**

**Figure 20: MIHS PDU in existence of MIH SA**

**Figure 21: MIHS PDU upon Transport Address Change**

## MIH Protocol Messages

### Message Types

Table 4 lists the new MIH messages types [IEEE802.21]

Table 4: MIH New Message Types

|  |  |
| --- | --- |
| Message name | Action ID |
| MIH\_Pro\_auth Request | xx |
| MIH\_Pro\_auth Response | yy |
| MIH\_Pro\_auth Indication | zz |
| MIH\_Pro\_auth\_Termination Request | kk |
| MIH\_Pro\_auth\_Termination Response | LL |
| MIH\_Security Indication | JJ |

Table 5 lists the messages that need extension

Table 5: MIH Message Extension

|  |  |
| --- | --- |
| Message Name  | Action ID  |
| Capability\_Discover\_Request | 1 |
| Capability\_Discover\_Response | 1 |

For the messages in Table 5, an additional Supported Security Cap List parameter is carried in a Security Capability List TLV of type MIH\_SEC\_CAP.

### MIH\_Pro\_Auth Request

|  |
| --- |
| MIH Header Fields (SID=3, Opcode=1, AID-xx) |
| Source Identifier = sending MIHF ID(Source MIHF ID TLV) |
| Destination Identifier = receiving MIHF ID(Destination MIHF ID TLV) |
| Candidate Identifier = candidate MIHF ID (optional)(Candidate MIHF ID TLV) |
| Mobile Node Identifier = mobile node MIHF ID (Optional )(Mobile node MIHF ID TLV) |
| Supported Protocol (Optional)(EAP or ERP TLV) |
| Result (Optional)(Result TLV)  |
| Lifetime (Optional)(Lifetime TLV)  |
| Integrity Check (IC) (Optional)(IC TLV) |
| Link address list of Point of attachment (Optional)(POA Link Address List TLV)  |
| Link address list of Mobile Node (Optional)(MN Link Address List TLV) |

### MIH\_Pro\_Auth Response

|  |
| --- |
| MIH Header Fields (SID=3, Opcode=2, AID-xx) |
| Source Identifier = sending MIHF ID(Source MIHF ID TLV) |
| Destination Identifier = receiving MIHF ID(Destination MIHF ID TLV) |
| Candidate Identifier = candidate MIHF ID (optional)(Candidate MIHF ID TLV) |
| Mobile Node Identifier = mobile node MIHF ID (Optional )(Mobile node MIHF ID TLV) |
| Supported Protocol (Optional)(EAP or ERP TLV) |
| Result (Optional)(Result TLV)  |
| Lifetime (Optional)(Lifetime TLV)  |
| Integrity Check (IC) (Optional)(IC TLV) |
| Link address list of Point of attachment (Optional)(POA Link Address List TLV)  |
| Link address list of Mobile Node (Optional)(MN Link Address List TLV) |

### MIH\_Pro\_Auth Indication

|  |
| --- |
| MIH Header Fields (SID=3, Opcode=3, AID-xx) |
| Source Identifier = sending MIHF ID(Source MIHF ID TLV) |
| Destination Identifier = receiving MIHF ID(Destination MIHF ID TLV) |
| Candidate Identifier = candidate MIHF ID (optional)(Candidate MIHF ID TLV) |
| Mobile Node Identifier = mobile node MIHF ID (Optional )(Mobile node MIHF ID TLV) |
| Supported Protocol (Optional)(EAP or ERP TLV) |

### MIH\_Pro\_Auth\_Termination Request

|  |
| --- |
| MIH Header Fields (SID=3, Opcode=1, AID-xx) |
| Source Identifier = sending MIHF ID(Source MIHF ID TLV) |
| Destination Identifier = receiving MIHF ID(Destination MIHF ID TLV) |
| Candidate Identifier = candidate MIHF ID (optional)(Candidate MIHF ID TLV) |
| Mobile Node Identifier = mobile node MIHF ID (Optional )(Mobile node MIHF ID TLV) |
| Integrity Check (IC) (IC TLV) |

### MIH\_Pro\_Auth\_Termination Response

|  |
| --- |
| MIH Header Fields (SID=3, Opcode=2, AID-xx) |
| Source Identifier = sending MIHF ID(Source MIHF ID TLV) |
| Destination Identifier = receiving MIHF ID(Destination MIHF ID TLV) |
| Candidate Identifier = candidate MIHF ID (optional)(Candidate MIHF ID TLV) |
| Mobile Node Identifier = mobile node MIHF ID (Optional )(Mobile node MIHF ID TLV) |
| Integrity Check (IC) (IC TLV) |

### MIH\_Security Indication

|  |
| --- |
| MIH Header Fields (SID=5, Opcode=2, AID-xx) |
| Source Identifier = sending MIHF ID (optional)(Source MIHF ID TLV) |
| Destination Identifier = receiving MIHF ID (optional)(Destination MIHF ID TLV) |
| Session Identifier = session id (optional)(Session ID TLV) |
| TLS = transport layer security(TLS TLV) |

# Conclusion

This proposal proposes solution with reference to the call for proposal of IEEE 802.21a. This proposal covers the comprehensive requirements as listed at the beginning of this proposal. We request the Task group to evaluate and adopt the proposed approaches as solution to the security requirements of IEEE 802.21a.