|  |  |
| --- | --- |
| INPUT CONTRIBUTION | |
| Group Name:\* | TDE |
| Title:\* | Interop Test Cases for Security |
| Source:\* | KETI and EGM |
| Contact: | JaeSeung Song, KETI, [jssong@sejong.ac.kr](mailto:jssong@sejong.ac.kr)  Sherzod Elamanov, KETI, [selamanov@gmail.com](mailto:selamanov@gmail.com)  Franck Le-Gall, EGM, [franck.le-gall@eglobalmark.com](mailto:franck.le-gall@eglobalmark.com)  Ahmed ABID, EGM, [ahmed.abid@eglobalmark.com](mailto:ahmed.abid@eglobalmark.com) |
| Date:\* | 2020-10-06 |
| Abstract:\* | The contribution proposes some Interoperability Test Descriptions for the security features to be added to TS-0013 |
| Agenda Item:\* | TBD |
| Work item(s): |  |
| Document(s)  Impacted\* | TS-0013 |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Decision requested or recommendation:\* | Incorporate the proposed text into TS-0013 |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

1. **Introduction**

This contribution consists of some test descriptions about Security to be included into the TS-0013.

The clause numbers and the TD identifier names in the following proposal may change when included in to TS-0013.

1. **Proposal**

### -----------------------Start of change 1-------------------------------------------

# Test Descriptions

## 8.4 Security management

### 8.4.1 Secure AE Registration

#### 8.4.1.1 PSK Security Association Establishment Framework

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_01 |
| **Objective:** | | | AE uses Provisioned Symmetric Key Security Association Establishment Framework to enable mutual authentication with the Registrar CSE. Registrar CSE performs AE authorization check on incoming AE registration request. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.2.2.1  oneM2M TS-0001 [1], clauses 9.6.29, 9.6.19, 9.16.20 |
|  | | | |
| **Pre-test conditions:** | | | * AE and Registrar CSE are pre-Provisioned with Kpsa = 123456,KpsaId = test@onem2m.com and Cipher Suites = TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256, TLS\_PSK\_WITH\_AES\_128\_CCM\_8 * Registrar CSE is provisioned with Service Subscribed Profile and Service Subscribed Node Resources. * Service Subscribed Node contains csi <Registrar CSE-ID> and rlk < URI of serviceSubscribedAppRule > attributes. * Registrar CSE is configured with <serviceSubscribedAppRule> resource having a CredentialD, APP-ID and AE-ID with the following values: * <m2m:asar rn="asar"> * <aci>00-test@onem2m.com</aci> * <aai>APP01</aai> * <aae>AE-ID</aae> * </m2m:asar> |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send a primitive to the Registrar CSE |
| 2 | Mca | PRO Check Primitive | Security Association Establishment |
| PRO Check TCP | TLS Handshake   * Cipher Suite:TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2 * KpsaId = test@onem2m.com |
| PRO Check UDP | DTLS Handshake   * Cipher Suite:TLS\_PSK\_WITH\_AES\_128\_CCM\_8 * Version: DTLS v1.2 * KpsaId = test@onem2m.com |
| 3 |  | IOP Check | Check if possible that Handshake was successful |
| 4 | Mca | PRO Check Primitive | * op = 1 (Create) * to = {CSEBaseName} * fr = AE-ID * rqi = (token-string) * ty = 2 (AE) * pc = Serialized representation of <AE> resource |
| 5 |  | IOP Check | Check that APP-ID, AE-ID, Credential ID are in <serviceSubscribedAppRule>  Check if possible that the <AE> resource is created in registrar CSE. |
| 6 | Mca | PRO Check Primitive | * rsc = 2001 (CREATED) * rqi = (token-string) same as received in request message * pc = Serialized representation of <AE> resource |
| 7 |  | IOP Check | AE indicates successful operation |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

### 8.4.1 Authentication

#### 8.4.1.1Authentication using the Provisioned Symmetric Key Security Association Establishment Framework

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 02 |
| **Objective:** | | | AE establishes mutual authentication with the Registrar CSE using Provisioned Symmetric Key Security Association Establishment Framework. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.2.2.1 |
|  | | | |
| **Pre-test conditions:** | | | * AE and Registrar CSE are pre-Provisioned with Kpsa = 123456,KpsaId = test@onem2m.com and Cipher Suites = TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The TLS client on AE sends a Client Hello Handshake message |
| 2 | Mca | PRO Check TCP | Client Hello handshake message   * Handshake Type = 0x01 (Client Hello) * Cipher Suite:TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2 |
| 3 | Mca | PRO Check TCP | Server Hello handshake message   * Handshake Type = 0x02 (Server Hello) * Cipher Suite:TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2   Server Hello Done handshake message   * Handshake Type = 0x0e (Server Hello Done) |
| 4 |  | Stimulus | The TLS client on AE sends Client Key Exchange, Change Cipher Spec, Finished messages |
| 5 | Mca | PRO Check TCP | The TLS client Key Exchange handshake message   * Handshake Type = 0x10 (Client Key Exchange) * psk\_identity = test@onem2m.com * Version: TLS v1.2   Client Change Cipher Spec message   * Content type = 0x14 (Change Cipher Spec)   Client Finished handshake message   * Handshake Type = 0x14 (Client Finished) * Version: TLS v1.2 |
| 6 |  | IOP Check | Check that The TLS server authenticated the Client by validating Verify Data  Check that AE associated the established TLS session with the CSE-ID |
| 7 | Mca | PRO Check TCP | Server New Session Ticket handshake message   * Handshake Type = 0x04 (New Session Ticket) * psk\_identity = test@onem2m.com * Version: TLS v1.2   Server Change Cipher Spec message   * Content type = 0x14 (Change Cipher Spec)   Server Finished handshake message   * Handshake Type = 0x14 (Client Finished) * Version: TLS v1.2 |
| 8 |  | IOP Check | Check that The TLS client authenticated the Server by validating Verify Data |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.1.2Authentication using the Provisioned Symmetric Key Security Association Establishment Framework

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 03 |
| **Objective:** | | | AE establishes mutual authentication with the Registrar CSE using Certificate-Based Security Association Establishment Framework. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.2.2.2 |
|  | | | |
| **Pre-test conditions:** | | | * The Registrar CSE uses the CSE-ID certificate signed by a root CA certificate * AE uses the AE-ID certificate signed by a root CA certificate * Cipher Suite = TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The TLS client on AE sends a client Hello Handshake message |
| 2 | Mca | PRO Check TCP | The TLS client sends a Hello handshake message to the TLS server   * Handshake Type = 0x01 (Client Hello) * Cipher Suite: TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2 |
| 3 | Mca | PRO Check TCP | The TLS server sends Hello, Certificate, Key Exchange, Certificate Request, Hello Done messages to the TLS client  Server Hello handshake message   * Handshake Type = 0x02 (Server Hello) * Cipher Suite: TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2   Server Certificate handshake message   * Handshake Type = 0x0b (Server Certificate) * Certificate: the Registrar CSE certificate   Server Key Exchange handshake message   * Handshake Type = 0x0c (Server Key Exchange) * Public key: ECDHE generated key   Server Certificate Request handshake message   * Handshake Type = 0x0d (Certificate Request)   Server Hello Done handshake message   * Handshake Type = 0x0e (Server Hello Done) |
| 4 |  | IOP Check | The TLS client on AE checks if the certificate of the Server is valid |
| 5 |  | Stimulus | The TLS client on AE sends Certificate, Client Key exchange, Certificate Verify, Change Cipher Spec, Finished messages |
| 6 | Mca | PRO Check TCP | Client Certificate handshake message   * Handshake Type = 0x0b (Client Certificate) * Certificate: AE certificate   Client Key Exchange message   * Handshake Type = 0x10 (Client Key Exchange) * Public key: ECDHE generated key   Client Certificate Verify message   * Handshake Type = 0x0f (Certificate Verify)   Client Change Cipher Spec message   * Content type = 0x14 (Change Cipher Spec)   Client Finished handshake message   * Handshake Type = 0x14 (Client Finished) |
| 7 |  | IOP Check | The TLS server on CSE checks if the certificate of the Client is valid |
| 8 | Mca | PRO Check TCP | The TLS server sends New Session Ticket, Change Cipher Spec, and Finished messages to the TLS client  Server New Session Ticket message   * Handshake Type = 0x04 (New Session Ticket)     Server Change Cipher Spec message   * Content type = 0x14 (Change Cipher Spec)   Server Finished message   * Handshake Type = 0x14 (Client Finished) * Version: TLS v1.2 |
| 9 |  | IOP Check | Check that The TLS client authenticated the Server by validating Verify Data |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

### 8.4.2 Authorization

#### 8.4.2.1Authorization using selfPrivileges

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 04 |
| **Objective:** | | | AE accesses <accessControlPolicy> resource using its selfPrivileges credentials |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0001 [1] clause 9.6.2.0 |
|  | | | |
| **Pre-test conditions:** | | | * CSEBase resource has been created in registrar CSE with name {CSEBaseName} * AE has created an <AE> resource on registrar CSE with name {AE} * accessControlPolicy resource has been created in registrar CSE under <AE> resource with name {accessControlPolicyName} * selfPrivileges attribute of {accessControlPolicyName} contains the following access control tuple:   acor = AE-ID  acop = 63 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send an accessControlPolicy Retrieve Request |
| 2 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{accessControlPolicyName} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 3 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 2000 (OK) * rqi = (token-string) same as received in request message * pc = Serialized representation of <accessControlPolicy> resource |
| 4 |  | IOP Check | AE indicates successful operation |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.1.2Authorization using accessControlPolicy privileges

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 05 |
| **Objective:** | | | AE accesses <AE> resource using its accessControlPolicyIDs attribute |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0001 [1] clause 9.6.2.0 |
|  | | | |
| **Pre-test conditions:** | | | * CSEBase resource has been created in registrar CSE with name {CSEBaseName} * AE has created an <AE> resource on registrar CSE with name {AE} * accessControlPolicy resource has been created in registrar CSE under <AE> resource with name {accessControlPolicyName} * accessControlPolicyIDs attribute of {AE} is set to resource id of {accessControlPolicyName} * privileges attribute of {accessControlPolicyName} contains the following access control tuple:   acor = AE-ID  acop = 34 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send an AE Retrieve Request |
| 2 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 3 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 2000 (OK) * rqi = (token-string) same as received in request message * pc = Serialized representation of <accessControlPolicy> resource |
| 4 |  | IOP Check | AE indicates successful operation |
| 5 |  | Stimulus | AE is requested to send an AE Delete Request |
| 6 | Mca | PRO Check Primitive | * op = 4 (Delete) * to = {CSEBaseName}/{AE} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 7 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 4103 (ACCESS\_DENIED) * rqi = (token-string) same as received in request message   pc = empty |
| 8 |  | IOP Check | Check if possible that the <AE> resource has not been removed in registrar CSE. |
| 9 |  | IOP Check | AE indicates unsuccessful operation (Delete error - no privilege) |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.1.3Authorization using default access privileges (owner is configured)

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 06 |
| **Objective:** | | | AE accesses <AE> resource using default access privileges |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0001 [1] clause 9.6.2.0 |
|  | | | |
| **Pre-test conditions:** | | | * CSEBase resource has been created in registrar CSE with name {CSEBaseName} * AE has created an <AE> resource on registrar CSE with name {AE} * <container> resource has been created in registrar CSE under <AE> resource with name {containerName} * accessControlPolicyIDs attribute of {containerName} is *NULL* * owner attribute of {containerName} = AE-ID |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send a container Retrieve Request |
| 2 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 3 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 2000 (OK) * rqi = (token-string) same as received in request message * pc = Serialized representation of <accessControlPolicy> resource |
| 4 |  | IOP Check | AE indicates successful operation |
| 5 |  | Stimulus | AE2 is requested to send a container Retrieve Request |
| 6 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE2-ID * rqi = (token-string) * pc = empty |
| 7 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 4103 (ACCESS\_DENIED) * rqi = (token-string) same as received in request message * pc = empty |
| 8 |  | IOP Check | AE indicates unsuccessful operation (Retrieve error - no privilege) |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.1.4Authorization using default access privileges (owner is not configured)

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 07 |
| **Objective:** | | | AE accesses <AE> resource using default access privileges |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0001 [1] clause 9.6.2.0 |
|  | | | |
| **Pre-test conditions:** | | | * CSEBase resource has been created in registrar CSE with name {CSEBaseName} * AE has created an <AE> resource on registrar CSE with name {AE} * <container> resource has been created in registrar CSE under <AE> resource with name {containerName} * accessControlPolicyIDs attribute of {containerName} is *NULL* * owner attribute of {containerName} is not set * creator attribute of {containerName} = AE-ID |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send a ContainerRetrieve Request |
| 2 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 3 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 2000 (OK) * rqi = (token-string) same as received in request message * pc = Serialized representation of <accessControlPolicy> resource |
| 4 |  | IOP Check | AE indicates successful operation |
| 5 |  | Stimulus | AE2 is requested to send a Container Retrieve Request |
| 6 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE2-ID * rqi = (token-string) * pc = empty |
| 7 | Mca | PRO Check Primitive | Registrar CSE sends response containing:   * rsc = 4103 (ACCESS\_DENIED) * rqi = (token-string) same as received in request message * pc = empty |
| 8 |  | IOP Check | AE indicates unsuccessful operation (Retrieve error - no privilege) |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.1.5Direct Dynamic Authorization

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 08 |
| **Objective:** | | | AE accesses <AE> resource using Direct Dynamic Authorization |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [1] clause 7.3.2.2 |
|  | | | |
| **Pre-test conditions:** | | | * CSEBase resource has been created in registrar CSE with name {CSEBaseName} * AE has created an <AE> resource on registrar CSE with name {AE} * <container> resource has been created in registrar CSE under <AE> resource with name {containerName} * Arbitrary set of <accessControlPolicy> resources are linked to the {containerName} |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send a Container Retrieve Request |
| 2 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 3 |  | IOP Check | Check if possible that Tokens or Token-Ids have not been included in the request |
| 4 |  | IOP Check | Check if possible that CSE selected a DAS Server based on accessControlRules linked to the requested resource |
| 5 | Mca | PRO Check Primitive | Registrar CSE sends a Notify request to the DAS server:   * op = 6 (Notify) * pc:   securityInfo: Direct Dynamic Authorization  Originator = AE-ID  Originator Resource Type = 3 (Container)  Operation = 2 (Retrieve) |
| 6 |  | IOP Check | Check that if the DAS Server issued token(s), they conform to the Token structure (TS-0003, clause 7.3.2.4) |
| 7 | Mca | PRO Check Primitive | The DAS server responds to the Registrar CSE:   * op = 6 (Notify response) * pc:   securityInfo: Direct Dynamic Authorization  (optional) token(s): authorization token(s)  (optional) dynamicACPInfo: information for creating accessControlPolicy dynamicaly |
| 8 |  | IOP Check | Check that if token(s) present in response content, the token is validated in the Registrar CSE successfully (TS-0003, clause 7.3.2.5) |
| 9 |  | IOP Check | Check that if dynamicACPInfo present in response content, the Registrar CSE created <accessControlPolicy> resource matching the dynamicACPInfo. |
| 10 | Mca | PRO Check Primitive | If access is granted, the Registrar CSE responds to the AE:   * rsc = 2000 (OK) * rqi = (token-string) same as received in request message * pc = Serialized representation of <container> resource   If access is not granted, the Registrar CSE responds to the AE:   * rsc = 4103 (ACCESS\_DENIED) * rqi = (token-string) same as received in request message * pc = empty |
| 11 |  | IOP Check | If access is granted, AE indicates successful operation, otherwise AE indicates unsuccessful operation (Retrieve error - no privilege) |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.1.6 Indirect Dynamic Authorization

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 09 |
| **Objective:** | | | AE accesses <AE> resource using Indirect Dynamic Authorization |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [1] clause 7.3.2.3 |
|  | | | |
| **Pre-test conditions:** | | | * CSEBase resource has been created in registrar CSE with name {CSEBaseName} * AE has created an <AE> resource on registrar CSE with name {AE} * <container> resource has been created in registrar CSE under <AE> resource with name {containerName} * Arbitrary set of <accessControlPolicy> resources are linked to the {containerName} |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE is requested to send a Container Retrieve Request |
| 2 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE-ID * rqi = (token-string) * pc = empty |
| 3 | Mca | PRO Check Primitive | * rsc = 4103 (ACCESS\_DENIED) * rqi = (token-string) same as received in request message * tqf: DAS Server PoA * pc = empty |
| 4 |  | IOP Check | AE indicates unsuccessful operation (Retrieve error - no privilege) |
| 5 |  | Stimulus | AE is requested to send a token request to the DAS using original request data. *AuthorSignIndicator* parameteris optional. |
| 6 |  | IOP Check | Check that if the DAS Server issued token(s), they conform to the Token structure (TS-0003, clause 7.3.2.4) |
| 7 |  | Stimulus | AE is requested to send a Container Retrieve Request with additional token(s) information |
| 8 | Mca | PRO Check Primitive | * op = 2 (Retrieve) * to = {CSEBaseName}/{AE}/{containerName} * fr = AE-ID * rqi = (token-string) * (optional) tkns: token(s) if ESData-protected Token(s) are provided * (optional) tids: token Id(s) if ESData-protected Token(s) are not provided * pc = empty |
| 9 | Mca | PRO Check Primitive | If the request in step 7 includes token Id(s), the Registrar CSE sends a Notify request to the DAS Server:   * op = 6 (Notify) * securityInfo Type: Indirect Dynamic Authorization * pc:   tids: token Id(s) |
| 10 | Mca | PRO Check Primitive | The DAS server responds to the Registrar CSE:   * op = 6 (Notify response) * pc:   securityInfo: Indirect Dynamic Authorization  token(s): authorization token(s) corresponding token Id(s) |
| 12 |  | IOP Check | Check that the token(s) are validated in the Registrar CSE successfully (TS-0003, clause 7.3.2.5) |
| 13 |  | IOP Check | If access is granted, AE indicates successful operation, otherwise AE indicates unsuccessful operation (Retrieve error - no privilege) |
| 14 | Mca | PRO Check Primitive | If access is granted, the Registrar CSE responds to the AE:   * rsc = 2000 (OK) * ltids: Local-Token-ID(s) * tkns: Token(s) * rqi = (token-string) same as received in request message * pc = Serialized representation of <container> resource   If access is not granted, the Registrar CSE responds to the AE:   * rsc = 4103 (ACCESS\_DENIED) * rqi = (token-string) same as received in request message   pc = empty |
| 15 |  | IOP Check | If access is granted, AE indicates successful operation, otherwise AE indicates unsuccessful operation (Retrieve error - no privilege) |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

### 8.4.2 Key provisioning management

#### 8.4.2.1 MEF Handshake Procedure using certificates

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_10 |
| **Objective:** | | | A MEF Handshake procedure establishes a mutually authenticated TLS session for protecting the communication between an MEF Client and MEF using pre-provisioned certificates. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.2 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client and MEF have been provisioned with certificates and Cipher Suite = TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | MEF Client and MEF establish the TLS or DTLS session using the certificate-based TLS handshake |
| 2 |  | IOP Check | Check that MEF Handshake is successful  Check that the MEF's certificate is verified against the set of provisioned MEF certificate trust anchors (as described in TS-0003 [12]) |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.2 MEF Handshake Procedure using Master Credentials

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 11 |
| **Objective:** | | | A MEF Handshake procedure establishes a mutually authenticated TLS or DTLS session for protecting the communication between an MEF Client and MEF using pre-provisioned Master Credentials. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.2 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client and MEF have been provisioned with Kpm = 123456, KpmID = psk\_identity, and Cipher Suites = TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256, TLS\_PSK\_WITH\_AES\_128\_CCM\_8 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | MEF Client and MEF establish the TLS or DTLS session using the certificate-based TLS handshake |
| 2 | Mca | PRO Check TCP/UDP | * psk\_identity = [test@onem2m.com](mailto:test@onem2m.com) * psk = 123456 |
| 3 |  | IOP Check | Check that MEF Handshake is successful |
|  |  |  |  |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.3 MEF Client Registration Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 12 |
| **Objective:** | | | The MEF Client registers with the MEF to confirm that it is willing to use the services of the MEF, under the authorization of the administrating stakeholder |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.3 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client, and MEF have been provisioned with the parameters described in TS-0003 [12], clause 8.3.7 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client sends a MEF Client Registration request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF * adminFQDN = FQDN of the administrating stakeholder * expirationTime = time when the registration shall expire |
| 4 |  | IOP Check | Check if possible that MEF has created a MEF Client Registration record |
| 5 | Mca | PRO Check TCP/UDP | The MEF sends a MEF Client Registration response   * MEFClientRegID = Identifier for the new MEF Client Registration * expirationTime = time when the MEF Client Registration record shall expire * MEF Client ID = Identifier of the MEF Client * adminFQDN = FQDN of the administrating stakeholder |
| 6 |  | IOP Check | Check if possible that MEF Client has stored parameters provided by the MEF |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.4 MEF Client Configuration Retrieval Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 13 |
| **Objective:** | | | The MEF Client retrieves MEF Client Configurations provided by the administrating stakeholder to the MEF. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.4 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client has previously performed the MEF Client Registration procedure to create the MEF Client Registration record. * The MEF Client Registration record is not expired. |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client sends a MEF Client Configuration Retrieval request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF, from MEF Instruction Configuration * MEFClientRegID = Identifier for the MEF Client registration record being updated |
| 4 | Mca | PRO Check TCP/UDP | The MEF sends a MEF Client Configuration Retrieval response   * MEFClientCfg = MEF Client Configuration currently associated with the identified MEF Client registration record |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.5 MEF Client Configuration Update Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 14 |
| **Objective:** | | | MEF Client updates the MEF Client registration by any combination of extending the *expirationTime* of the MEF Client Registration record or updating the *labels*. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.5 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client has previously performed the MEF Client Registration procedure to create the MEF Client Registration record. * The MEF Client Registration record is not expired. |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client shall send a MEF Client Registration Update request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF * MEFClientRegID = Identifier for the MEF Client registration record being updated * (optional) expirationTime = time when the MEF Client registration record shall expire * (optional) labels = labels to aid discovery of the MEF Client registration record   NOTE: At least one of expirationTime and labels shall be included. |
| 4 |  | IOP Check | Check if possible that MEF has updated the MEF Client Registration record with the proposed values |
| 5 | Mca | PRO Check TCP/UDP | The MEF sends a MEF Client Registration Update response   * (optional) expirationTime = time when the MEF Client registration record shall expire * (optional) labels = labels to aid discovery of the MEF Client registration record   NOTE: The response only includes *expirationTime* and/or *labels* if those parameters were present in the corresponding request. |
| 6 |  | IOP Check | Check if possible that MEF Client has stored parameters provided by the MEF |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.6 MEF Client De-Registration Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 15 |
| **Objective:** | | | The MEF Client registers with the MEF to confirm that it is willing to use the services of the MEF, under the authorization of the administrating stakeholder |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.6 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client has previously performed the MEF Client Registration procedure to create the MEF Client Registration record. * The MEF Client Registration record is not expired. |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client sends a MEF Client De-Registration request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF * MEFClientRegID = Identifier for the MEF Client Registration record being ended |
| 4 |  | IOP Check | Check if possible that MEF has deleted the information associated with the identified MEF Client Registration record |
| 5 |  | IOP Check | The MEF sends a MEF Client Registration Update response. The MEF Client indicates the success of the operation. |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.7 MEF Key Registration Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 16 |
| **Objective:** | | | Source MEF Client establishes a symmetric key with the MEF which can be retrieved for use by one or more Target MEF Clients |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.7 |
|  | | | |
| **Pre-test conditions:** | | | * The Source MEF Client is provided with (or has otherwise determined) the information in the MEF Key Registration Configuration (TS-0003 [12], clause 8.3.7.3) * The Source MEF Client has performed the MEF Client Registration procedure (TS-0003 [12], clause 8.3.5.2.3) with the MEF for the administrating stakeholder identified in the MEF Key Registration Configuration |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The Source MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client sends a MEF Key Registration request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF * expirationTime = time when the MEF Client Registration shall expire * adminFQDN = Identifier for the administrating stakeholder * SUID = The Security Usage Identifier limiting the security feature in which the symmetric key may be used * (optional) targetIDs = list of identifiers for the initial set of Target MEF Clients authorized to retrieve the symmetric key * (optional) Key Value = output symmetric key value which is self-generated by the Source MEF Client |
| 4 |  | IOP Check | If the MEF Key Registration request included Key Value, check that MEF has stored the value. Otherwise, MEF generates Key Value from the (D)TLS session using TLS Key Export. |
| 5 | Mca | PRO Check TCP/UDP | The MEF sends a MEF Key Registration response   * RelativeKeyID = the relative part of the Key Identifier associated with the Key Registration * expirationTime = time when the MEF Client Registration record shall expire * Source MEF Client ID = Identifier of the Source MEF Client * adminFQDN = FQDN of the administrating stakeholder * SUID = the Security Usage Identifier limiting the security feature in which the symmetric key may be used * targetIDs =list of identifiers for the initial set of Target MEF Clients authorized to retrieve the symmetric key |
| 6 |  | IOP Check | Check if possible that the Source MEF Client and MEF has stored the output symmetric key value and corresponding Key Identifier |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.8 MEF Key Retrieval Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 17 |
| **Objective:** | | | The Target MEF Client to retrieve the Key Value from a MEF corresponding to a RelativeKeyID received by the Target MEF Client |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.8 |
|  | | | |
| **Pre-test conditions:** | | | * The Target MEF Client has performed the MEF Client Credential Configuration with the MEF, including configuration of the MEF Key Retrieval URI * The Source MEF Client has performed the MEF Key Registration procedure with the MEF, resulting in a registered Key Value and assigned RelativeKeyID for a specific administrating stakeholder and Security Usage Identifier * The Target MEF Client received a Key Identifier from the Initiating-MEF Client in a security feature with the SUID which the Source MEF Client provided to the MEF during the MEF Key Registration procedure * The Target MEF Client may expect that it is authorized to obtain the corresponding output symmetric key value. |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client sends a MEF Key Retrieval request |
| 3 | Mca | PRO Check Primitive | * RelativeKeyID = The relative part of the Key Identifier received from the Source MEF Client in a security feature |
| 4 | Mca | PRO Check TCP/UDP | The MEF sends a MEF Key Retrieval response   * expirationTime = time when the Key Registration shall expire * Source MEF Client ID = Identifier of the Source MEF Client * adminFQDN = Identifier for the administrating stakeholder * SUID = the Security Usage Identifier limiting the security feature in which the symmetric key may be used * Key Value = The registered value of the output symmetric key |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.9 MEF Key Registration Update Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 18 |
| **Objective:** | | | MEF Client updates the MEF Client registration by any combination of extending the *expirationTime* of the MEF Client Registration record or updating the *labels*. |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.9 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client has previously performed the MEF Key Registration procedure to create the key registration * The key registration is not expired |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client shall send a MEF Key Registration Update request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF * RelativeKeyID = the relative part of the Key Identifier associated with the Key Registration * (optional) expirationTime = time when the Key Registration shall expire * (optional) labels = labels to aid discovery of the registered key * (optional) targetIDs = proposed list of identifiers for the set of Target MEF Clients authorized to retrieve the symmetric key   NOTE: At least one of expirationTime, labels or targetIDs shall be included. |
| 4 |  | IOP Check | Check if possible that MEF has updated the metadata with the proposed values |
| 5 | Mca | PRO Check TCP/UDP | The MEF sends a MEF Key Registration Update response   * (optional) expirationTime = current time when the key registration shall expire * (optional) labels = Updated list of labels to aid discovery of the Key Registration, if any * (optional) targetIDs = current list of identifiers for the initial set of Target MEF Clients authorized to retrieve the symmetric key   NOTE: The response includes only those parameters that were present in the corresponding request. |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.2.10 MEF Key De-Registration Procedure

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 19 |
| **Objective:** | | | Source MEF Client requests the MEF to stop distributing the registered key |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.3.5.2.10 |
|  | | | |
| **Pre-test conditions:** | | | * The MEF Client has previously performed the MEF Key Registration procedure to create the key registration. * The key registration is not expired. |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | The MEF Client establishes a TLS (or DTLS) connection with the MEF by performing the MEF Handshake procedure |
| 2 |  | Stimulus | The MEF Client sends a MEF Key De-Registration request |
| 3 | Mca | PRO Check TCP/UDP | * MEF-FQDN = FQDN of the MEF * RelativeKeyID = the relative part of the Key Identifier associated with the Key Registration |
| 4 |  | IOP Check | Check if possible that MEF has deleted the information associated with the identified key registration |
| 5 |  | IOP Check | The MEF sends a MEF Client De-Registration response. The MEF client indicates success of the operation. |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

### 8.4.3 End-to-End security management

#### 8.4.3.1 End-to-End Security of Primitives (ESPrim) Architecture

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_ 20 |
| **Objective:** | | | AE sends an arbitrary request primitive inside of ESPrim Object to CSE |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [12], clause 8.4.2 |
|  | | | |
| **Pre-test conditions:** | | | * AE and CSE has established a secure ESPrim connection, so that both are able to extract ESPrim Objects sent from each other * AE has produced an ESPrim Object from the serialization of the arbitrary request primitive |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE sends a NOTIFY Request Message with ESPrim Object |
| 2 | Mca | PRO Check Primitive | * op = 5 (Notify) * to = {CSEBaseName} * from = AE-ID * rqi = (token-string) * pc: {seci: {sit = “esprimObject “, epo: serialized ESPrim Object }} |
| 3 |  | IOP Check | Check if possible that the CSE successfully extracted the inner request primitive.  Check if possible that the CSE successfully processed the inner request primitive. |
| 4 | Mca | PRO Check Primitive | The CSE sends a NOTIFY response to the AE   * op = 5 (Notify) * to = AE-ID * from = CSE-ID * rqi = (token-string) * pc: {seci: {sit = “esprimObject “, epo: serialized ESPrim Object }} |
| 5 |  | IOP Check | Check that the AE successfully extracted the inner response primitive.  Check that the AE successfully processed the inner response primitive. |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

#### 8.4.3.2 End-to-End Certificate-based Key Establishment (ESCertKE)

| **Interoperability Test Description** | | | |
| --- | --- | --- | --- |
| **Identifier:** | | | TD\_M2M\_SE\_xx |
| **Objective:** | | | AE establishes a connection with the Registrar CSE using pairwiseE2EKey |
| **Configuration:** | | | M2M\_CFG\_01 |
| **References:** | | | oneM2M TS-0003 [**Error! Reference source not found.**], clause 8.7.2 |
|  | | | |
| **Pre-test conditions:** | | | * Both the Registrar CSE and AE support ESCertKE and are provisioned with private key and certificates. Both entities are configured with the information needed for the authentication and identification. * Cipher Suite = TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 |
| **Test Sequence** | | | |
| **Step** | **RP** | **Type** | **Description** |
| 1 |  | Stimulus | AE sends an ESCertKE Message 1 in Notify request |
| 2 | Mca | PRO Check Primitive | * op = 5 (Notify) * to = {CSEBaseName} * from = AE-ID * rqi = (token-string) * pc: {seci: {sit = “escertkeMessage“,eckm: ESCertKE Message 1 }}   ESCertKE Message 1 includes TLS a Client Hello handshake message:   * Handshake Type = 0x01 (Client Hello) * Cipher Suite: TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2 |
| 3 | Mca | PRO Check Primitive | The Registrar CSE sends an ESCertKE Message 2 in Notify response   * op = 5 (Notify) * to = AE-ID * from = CSE-ID * rqi = (token-string) * pc: {seci: {sit = “escertkeMessage“,eckm: ESCertKE Message 2 }}   ESCertKE Message 2 includes Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done messages  Server Hello handshake message   * Handshake Type = 0x02 (Server Hello) * Cipher Suite: TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 * Version: TLS v1.2   Certificate handshake message   * Handshake Type = 0x0b (Server Certificate) * Certificate: the Registrar CSE certificate   Server Key Exchange handshake message   * Handshake Type = 0x0c (Server Key Exchange) * Public key: ECDHE generated key   Certificate Request handshake message   * Handshake Type = 0x0d (Certificate Request)   Server Hello Done handshake message   * Handshake Type = 0x0e (Server Hello Done) |
| 4 |  | IOP Check | The TLS client on AE checks if the certificate of the Server is valid |
| 5 |  | Stimulus | AE sends an ESCertKE Message 3 in Notify request |
| 6 | Mca | PRO Check Primitive | * op = 5 (Notify) * to = {CSEBaseName} * from = AE-ID * rqi = (token-string) * pc: {seci: {sit = “escertkeMessage“,eckm: ESCertKE Message 3 }}   ESCertKE Message 3 includes Certificate, Client Key exchange, Certificate Verify, Change Cipher Spec, Finished messages  Certificate handshake message   * Handshake Type = 0x0b (Client Certificate) * Certificate: AE certificate   Client Key Exchange message   * Handshake Type = 0x10 (Client Key Exchange) * Public key: ECDHE generated key   Certificate Verify message   * Handshake Type = 0x0f (Certificate Verify)   Change Cipher Spec message   * Content type = 0x14 (Change Cipher Spec)   Finished handshake message   * Handshake Type = 0x14 (Client Finished) |
| 7 |  | IOP Check | The TLS server on CSE checks if the certificate of the Client is valid |
| 8 | Mca | PRO Check Primitive | The Registrar CSE sends an ESCertKE Message 2 in Notify response   * op = 5 (Notify) * to = AE-ID * from = CSE-ID * rqi = (token-string)   pc: {seci: {sit = “escertkeMessage“,eckm: ESCertKE Message 4 }}  ESCertKE Message 4 includes Change Cipher Spec, and Finished messages  Server Change Cipher Spec message   * Content type = 0x14 (Change Cipher Spec)   Server Finished message   * Handshake Type = 0x14 (Client Finished) |
| 9 |  | IOP Check | Check that The TLS client authenticated the Server by validating Verify Data |
| 10 |  | IOP Check | Check that AE and the Registrar CSE has generated and cached a pairwiseE2EKey |
| IOP Verdict | |  | |
| PRO Verdict | |  | |

### -----------------------End of change 1-------------------------------------------