PSIA Common Security (CSEC) Model
V2.0 Rev 1

System WG
<table>
<thead>
<tr>
<th>Revision History</th>
<th>Description</th>
<th>Date</th>
<th>By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>Initial Draft</td>
<td>July 9, 2010</td>
<td>James Wang / Roger Richter</td>
</tr>
<tr>
<td>Rev 1</td>
<td>Changes based on SWG input: 1) csec.xsd: a) breakout symbolic permissions to separate schema, b) change to allow new type of permission (ExplicitPermissionDescriptor) 2) csecPermissionDictionary.xsd: rename all permission symbolic names using path-like-hierarchy and add new permissions suggested by SWG. 3) Add Resource-Requirement Table (Section 4) 4) Add supportedPermissions Resource to advertise Permission Dictionary 5) Updated Resource Hierarchy diagram to reflect new Resource (supportedPermissions). 6) general cleanup</td>
<td>August 20, 2010</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 2</td>
<td>1) Add &lt;dictionaryName&gt; to csecPermissionDictionary.xsd 2) Add &lt;permissionBitOffset&gt; to csecPermissionDictionary.xsd 3) CSEC.xsd: a) add &lt;userDescription&gt; string, b) add &lt;fullName&gt;, c) add &lt;UserCODES&gt;, d) add &lt;subDeviceAuthorizationLevel&gt;, e) Changed all REST &lt;id&gt;’s to comply with &quot;psiaCommonTypes.xsd&quot;, using &quot;LocalID&quot; type and 1-based indices vs 0-based indices.</td>
<td>September 21, 2010</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 3</td>
<td>Add pending features + others suggested by System, Access, &amp; Intrusion WGs: 1) Add &quot;Owner GUID&quot; QS required to take ownership of device 2) Add &quot;deviceOwnership/scope&quot; resource (scoping of &quot;ownership&quot;) 3) Add &quot;deviceOwnership/status&quot; resource 4) Add simple MDS Digest for Ownership MAC 5) Add &quot;loginPortAffinity&quot; to &quot;PermissionDescriptor&quot; to allow Permissions to only be enabled/allowed if the user request came in on a specific port 6) Add text requiring &quot;admin&quot; Group and User 7) Add text requiring &quot;viewonly&quot; Group and User 8) Add ability to enable/disable Users and Groups 9) Replace &lt;UserCODES&gt; with &lt;UserCODEList&gt; 10) updated REST Resource Hierarchy Diagram</td>
<td>February 11, 2011</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 3b</td>
<td>Major cleanup of text and add more explanation of Users/Groups.</td>
<td>November 11, 2010</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 4c</td>
<td>Feedback from SWG: 1) Fix reference links</td>
<td>June 7, 2011</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 4e</td>
<td>Feedback from ACWG: 1) Add &quot;ACWGPermissionInfoList&quot; to &lt;CSECPermissionGroup&gt; to achieve encapsulation of ACWG Permissions into CSEC 2) Remove ACWG related elements from &lt;DeviceScopeRestriction&gt;. 3) Merge/Add some ACWG Types to psiaCommonTypes.xsd 4) Removed text describing Level-Based Permission 5) Add text regarding ACWG Permission encapsulation 6) Add ACWGCommonTypes.xsd</td>
<td>August 17, 2011</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 4f</td>
<td>1) Add &quot;UID&quot; to TimeScheduleInfo and HolidayInfo</td>
<td>August 16, 2011</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 4g</td>
<td>1) Removed schema text and replace with link to psia schema repository</td>
<td>September 1, 2011</td>
<td>James Wang</td>
</tr>
<tr>
<td>Rev 4h</td>
<td>Feedback from ACWG + other improvements: 1) Make KeyManager optional (Sec 4.1). 2) Allow for multiple device certificates 3) Change CA Service to allow management of &quot;root cert package&quot;</td>
<td>April 30, 2012</td>
<td>James Wang</td>
</tr>
</tbody>
</table>

Version 1.0 TBD: 1) Add More descriptive text throughout 2) Describe example VMS – CSEC Device flows 3) Add More MAC-algorithms for deviceOwnership
<table>
<thead>
<tr>
<th>Version 1.1</th>
<th>Changed CSEC to now be functionally subdivided into 'Profiles' per input and feedback on Service Model v2.0 and its effects on PSIA common specs.</th>
<th>August 12, 2012</th>
<th>Roger Richter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.1, Rev. 03</td>
<td>Modified Section 2 to further define the function of Profiles within CSEC. Added the new 'Core' profile, plus created tables to cover fundamental meaning of Basic, Full and Core profiles. Updated all requirements tables in Section 4 to reflect the new Core profile requirements. Modified Section 7.6, /PSIA/CSEC/AAA resources, to reflect the new read-only requirements of the Core profile.</td>
<td>November 6, 2012</td>
<td>Roger Richter</td>
</tr>
<tr>
<td>Version 1.1, Revision 0.4a</td>
<td>Added more detail to the CORE profile resources being read-only in the resource tables. Removed wording saying Core AAA resources MUST disallow PUT, POST and DELETE.</td>
<td>January 17, 2013</td>
<td>Roger Richter</td>
</tr>
<tr>
<td>Version 1.2, Revision 0.1</td>
<td>Obsoleted MD5 as an acceptable HMAC algorithm. SHA-256 is now the standard for all PSIA systems, and devices, except those that cannot support SHA-256. In those cases SHA-1 is the minimum standard.</td>
<td>April 5, 2014</td>
<td>Roger Richter</td>
</tr>
<tr>
<td>Version 1.2, R0.2</td>
<td>In Section 7.2 added support for RFC 6265 usage of &quot;Max-Age&quot; expiry designation, in addition to (the already present) &quot;Expires&quot; header designation. Added the new &quot;PLAI&quot; profile requirements in Section 4.</td>
<td>April 22, 2014</td>
<td>Roger Richter</td>
</tr>
<tr>
<td>Version 2.0, R0</td>
<td>Updates per Topologies, general cleanup</td>
<td>Feb 18, 2015</td>
<td>Jeffrey Longo</td>
</tr>
<tr>
<td>Version 2.0 R 1</td>
<td>Updates per group meeting 2/26/15</td>
<td>Feb 26, 2015</td>
<td>Jeffrey Longo</td>
</tr>
</tbody>
</table>
Copyright PSIA

Disclaimer
THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Without limitation, PSIA disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and PSIA disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.
No license, express or implied, by estoppel or otherwise, to any PSIA or PSIA member intellectual property rights is granted herein.

Except that a license is hereby granted by PSIA to copy and reproduce this specification for internal use only.

Contact the Physical Security Interoperability Alliance at info@psialliance.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.
Figure 2: Example Groups

7.6.1 /PSIA/CSEC/AAA/users ......................................................... 47
6.6.1.1 /PSIA/CSEC/AAA/users/<id> ........................................... 49
6.6.1.2 /PSIA/CSEC/AAA/users/<GUID> ....................................... 50
6.6.1.3 Required Special Users and Groups ..................................... 50
6.6.1.3.1 “admin” ........................................................................... 50
6.6.1.3.2 “viewonly” ...................................................................... 50
6.6.2 /PSIA/CSEC/AAA/permissionGroups ....................................... 50
6.6.2.1 /PSIA/CSEC/AAA/permissionGroups/<id> .............................. 55
6.6.3 /PSIA/CSEC/AAA/supportedPermissions .................................. 55
7.0 RTSP/SRTP Usage ...................................................................... 60
9.0 SRTP Crypto Tables ..................................................................... 62
8.0 CSEC Schema (“csec.xsd”) ......................................................... 66
9.0 CSEC Permission Dictionary Schema (“csecPermissionDictionary.xsd”) ........................................................................ 66
10.0 PSIA Common Types Schema (“psiaCommonTypes.xsd”) .......... 66
11.0 ACWG Common Types Schema (“ACWGCommonTypes.xsd”) ...... 66
1.0 Introduction and Overview

This document specifies the new Common Security Service (CSEC) which is designed to be the successor of the /PSIA/Security Service in IPMD v1/v1.1.

The previous /PSIA/Security Service provides a “trivial” security service allowed the creation of “users” and “adminAccesses” without specifying what the actual permissions are. Since many xMS’s only require administrative access to devices, the previous service was sufficient for that purpose. The CSEC Service specification is created to meet the demands of emerging systems that require a more sophisticated Security Model which define user-ids, permissions, and groups in a richer, more-flexible way. In addition, the single “/PSIA/Security/srtpMasterKey” resource is not capable of managing SRTP/SRTCP sessions for a wide number of channels and clients (i.e. single key must be used for all channels and sessions, with implicit re-keying after lifetime expiration). The CSEC Service specification provides a better define mechanism for SRTP key management.

CSEC also describes other security and authentication related requirements of PSIA.

1.1 CSEC and Functional Requirements

This document does not establish requirements in a stand-alone manner. To determine what is required, refer to a profile within a PSIA protocol specification. This profile will declare what PSIA topology it conforms to, from which you can ascertain which CSEC services, resources, or other mandates are required.

PSIA defines the following topologies:
- Master-Slave over the Internet
- Master-Slave over a LAN
- Peer to Peer over the Internet
- Peer to Peer over a LAN

For more information on PSIA Profiles and Topologies, please refer to the PSIA Service Model, version 3.0 or later.
## 2.0 References

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>PSIA Service Model specification</td>
<td><a href="http://www.psialliance.org/documents/PSI-Service-Model_version_1_0.pdf">http://www.psialliance.org/documents/PSI-Service-Model_version_1_0.pdf</a>, March 17, 2009</td>
</tr>
</tbody>
</table>
3.0 REST Resource Hierarchy

CSEC v1r4 Device Security Service Structure
## 4.0 REST Resource Requirements

### 4.1 /PSIA/CSEC (Root) Services & Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>*</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>deviceOwnership (Required for PLAI, applies to data ownership)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>deviceCertificate</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>certificateAuthority</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KeyManager</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>AAA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.2 /PSIA/CSEC/deviceOwnership Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scope</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Status</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.3 /PSIA/CSEC/certificateAuthority Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.4 /PSIA/CSEC/KeyManager Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;id&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;id&gt;/index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;id&gt;/description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;id&gt;/caCertificate</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.4.1 /PSIA/CSEC/KeyManager/directMKIKeyList Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/mime</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.4.2 /PSIA/CSEC/KeyManager/schemes Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NegotiatedMKIList</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MIKEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.4.2.1 /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;MKI&gt;/mime</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.4.2.2 /PSIA/CSEC/KeyManager/schemes/MIKEY Resources

<table>
<thead>
<tr>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>indexr</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.5 /PSIA/CSEC/AAA Resources

<table>
<thead>
<tr>
<th>Command</th>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 4.5.1 /PSIA/CSEC/AAA/users Resources

<table>
<thead>
<tr>
<th>Command</th>
<th>M-S Internet</th>
<th>M-S LAN</th>
<th>P-P Internet</th>
<th>P-P LAN</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>indexr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RO</td>
<td>RO</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>permissionGroups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

#### 4.5.2 /PSIA/CSEC/AAA/permissionGroups Resources
4.6 /PSIA/System Resources (Service Model v2.0+)

<table>
<thead>
<tr>
<th>Command</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>indexr</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>RO</td>
<td>RO</td>
<td>&lt;id&gt;</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

5.0 Audit Log

The CSEC Service does not contain an Audit-Log resource. Instead, the existing /PSIA/System/logging resource from IPMD will be leveraged to perform local AAA logging. The messages should be logged using “Facility” code 10 (“security/authorization messages”) [see RFC 3164, page 9] embedded as the “PRI” value within the <message> value member of <LogMessage>.

From RFC3164:

“The full format of a syslog message seen on the wire has three discernable parts. The first part is called the PRI, the second part is the HEADER, and the third part is the MSG. The total length of the packet MUST be 1024 bytes or less.”

[PRI]

“The Priority value is calculated by first multiplying the Facility number by 8 and then adding the numerical value of the Severity. For example, a kernel message (Facility=0) with a Severity of Emergency (Severity=0) would have a Priority value of 0. Also, a "local use 4" message (Facility=20) with a Severity of Notice (Severity=5) would have a Priority value of 165. In the PRI part of a syslog message, these values would be placed between the angle brackets as <0> and <165> respectively. The only time a value of "0" will follow the "<" is for the Priority value of "0". Otherwise, leading "0"s MUST NOT be used.”

[HEADER]
The HEADER part contains a timestamp and an indication of the hostname or IP address of the device. The HEADER part of the syslog packet MUST contain visible (printing) characters. The code set used MUST also be seven-bit ASCII in an eight-bit field like that used in the PRI part. In this code set, the only allowable characters are the ABNF VCHAR values (%d33-126) and spaces (SP value %d32).

The HEADER contains two fields called the TIMESTAMP and the HOSTNAME. The TIMESTAMP will immediately follow the trailing ">" from the PRI part and single space characters MUST follow each of the TIMESTAMP and HOSTNAME fields. HOSTNAME will contain the hostname, as it knows itself. If it does not have a hostname, then it will contain its own IP address. If a device has multiple IP addresses, it has usually been seen to use the IP address from which the message is transmitted. An alternative to this behavior has also been seen. In that case, a device may be configured to send all messages using a single source IP address regardless of the interface from which the message is sent. This will provide a single consistent HOSTNAME for all messages sent from a device.

The TIMESTAMP field is the local time and is in the format of "Mmm dd hh:mm:ss" (without the quote marks) where:

Mmm is the English language abbreviation for the month of the year with the first character in uppercase and the other two characters in lowercase. The following are the only acceptable values:

Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

dd is the day of the month. If the day of the month is less than 10, then it MUST be represented as a space and then the number. For example, the 7th day of August would be represented as "Aug 7", with two spaces between the "g" and the "7".

hh:mm:ss is the local time. The hour (hh) is represented in a 24-hour format. Valid entries are between 00 and 23, inclusive. The minute (mm) and second (ss) entries are between 00 and 59 inclusive.

A single space character MUST follow the TIMESTAMP field.

The HOSTNAME field will contain only the hostname, the IPv4 address, or the IPv6 address of the originator of the message. The preferred value is the hostname. If the hostname is used, the HOSTNAME field MUST contain the hostname of the device as specified in STD 13 [4]. It should be noted that this MUST NOT contain any embedded spaces. The Domain Name MUST NOT be included in the HOSTNAME field. If the IPv4 address is used, it MUST be shown as the dotted decimal notation as used in STD 13 [5]. If an IPv6 address is used, any valid representation used in RFC 2373 [6] MAY be used. A single space character MUST also follow the HOSTNAME field.

The MSG part will fill the remainder of the syslog packet. This will
usually contain some additional information of the process that generated the message, and then the text of the message. There is no ending delimiter to this part…”

“The MSG part has two fields known as the TAG field and the CONTENT field. The value in the TAG field will be the name of the program or process that generated the message. The CONTENT contains the details of the message. This has traditionally been a freeform message that gives some detailed information of the event. The TAG is a string of ABNF alphanumeric characters that MUST NOT exceed 32 characters. Any non-alphanumeric character will terminate the TAG field and will be assumed to be the starting character of the CONTENT field. Most commonly, the first character of the CONTENT field that signifies the conclusion of the TAG field has been seen to be the left square bracket character ("["), a colon character (":"), or a space character. This is explained in more detail in Section 5.3.”

In short the log entry is of the format: PRI HEADER MSG
Further division into sub-fields: PRI TIMESTAMP HOSTNAME TAG CONTENT

<table>
<thead>
<tr>
<th>Facility</th>
<th>Severity</th>
<th>PRI</th>
<th>Example Log Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1 - Alert</td>
<td>81</td>
<td>&lt;81&gt; Aug 3 20:11:15 my.machine.org aaaproc: intrusion detected</td>
</tr>
<tr>
<td>10</td>
<td>2 - Critical</td>
<td>82</td>
<td>&lt;82&gt; Aug 3 10:10:15 my.machine.org aaaproc: too many login failures</td>
</tr>
<tr>
<td>10</td>
<td>3 - Error</td>
<td>83</td>
<td>&lt;83&gt; Aug 2 22:15:30 my.machine.org aaaproc: database partially corrupted</td>
</tr>
<tr>
<td>10</td>
<td>4 - Warning</td>
<td>84</td>
<td>&lt;84&gt; Aug 2 22:10:15 my.machine.org aaaproc: database nearly full</td>
</tr>
<tr>
<td>10</td>
<td>5 - Notice</td>
<td>85</td>
<td>&lt;85&gt; Aug 2 21:05:08 my.machine.org aaaproc: unauthorized operation</td>
</tr>
<tr>
<td>10</td>
<td>6 - Informational</td>
<td>86</td>
<td>&lt;86&gt; Aug 2 20:10:15 my.machine.org aaaproc: routine backups started</td>
</tr>
<tr>
<td>10</td>
<td>7 - Debug</td>
<td>87</td>
<td>&lt;87&gt; Aug 1 09:00:00 my.machine.org aaaproc: debug test code 357</td>
</tr>
</tbody>
</table>

In future specifications, it is possible to create an AAA-only log locally, as well as a Proxy-Client to forward log messages to external AAA server using the server’s specific auditing-message protocol.
6.0 HTTPS and Authentication

All CSEC compliant PSIA nodes are required to meet the following requirements. Please note that the requirements are ‘topology dependent’. As such, implementers should take note of the requirements as they are described in the ensuing tables.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>M-S Internet</th>
<th>P-P Internet</th>
<th>M-S LAN</th>
<th>P-PLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Authentication</td>
<td>All PSIA Nodes shall provide Digest-based Authentication for all HTTP sessions upon challenge; OR…they shall provide HTTPS support as defined below. No BASIC authentication support is allowed over unencrypted HTTP.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTPS, and Authentication</td>
<td>All Nodes shall provide support for HTTPS sessions upon contact/request, via port 443 unless otherwise configured. All nodes shall provide TLS (1.0 or greater) security support for all HTTPS sessions. Additionally, Basic authentication, or greater, shall be supplied to authenticate all HTTPS sessions.</td>
<td>✔</td>
<td>✔</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

*= HTTP or HTTPS is required in a LAN topology

Resource Details

6.1 /PSIA/CSEC

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC</th>
<th>Type</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods Query String(s) Inbound Data Return Result</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td>Common Security (CSEC) Service root.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.2 /PSIA/CSEC/deviceOwnership

“If a cookie is used to store some information, such as an authentication token, that the server can validate without reliance on session state, cookies are perfectly RESTful…”[“REST Anti-Patterns”].
Function

Resource used to acquire administrative “ownership” of a device in a Master-Slave topology or data in a peer – peer topology.

This capability is only granted to a User that belongs to a Permission Group that contains the Symbolic Permission, or custom equivalent of, “/Configure/Security” (i.e. Root of Security, which equates to all of the Security Service).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>OwnerGUID, ExpireTime OR MaxAge</td>
<td>None</td>
<td>&lt;CSECOwnershipCookie&gt; or “HTTP 409 Conflict”</td>
</tr>
<tr>
<td>PUT POST DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>Notes</td>
<td>ResetOwnership</td>
<td>&lt;CSECOwnershipCookie&gt; or None</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
</tbody>
</table>

To acquire device-ownership, an administrative node will attempt, for example:

GET /PSIA/CSEC/deviceOwnership?OwnerGUID=8b7a0a80-5283-486f-a4ec-40dc04aaa373

If the request succeeds, ownership is granted via the &lt;CSECOwnershipCookie&gt;, in HTTP payload, which contains information used for all subsequent administrative requests.

The &lt;CSECOwnershipCookie&gt; XML contains a cookie string value which holds two HTTP “cookies” [RFC 2965.6265]. These two HTTP cookies are also optionally given in the HTTP response header as:

Set-Cookie2: owner-code=”generated-string-values”; expires=”date-time”; path=”/PSIA”; Secure; Version=”1”; issuer-signature=”generated-signature”

Or, in the cases where max-age is used:

Set-Cookie2: owner-code=”generated-string-values”; max-age=”integer”; path=”/PSIA”; Secure; Version=”1”; issuer-signature=”generated-signature”

On subsequent administrative requests, the cookies must be given in HTTP header (with the addition of a 3rd Identification Cookie appended):

Cookie: $Version=1; owner-code=”generated-string-values”; $Path=”/PSIA”; issuer-signature=”generated-signature”; OwnerGUID=GUID

The target device will use the issuer-signature and owner-code to verify ownership.
Example XML snippets:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<CSECOwnershipCookie version="1.0" xmlns="urn:psialliance-org">
  <ownerCode>3A84D6F8</ownerCode>
  <expires>Sun, 01-Jan-2012 13:00:00 GMT</expires>
  <path>/PSIA</path>
  <cookieVersion>1</cookieVersion>
  <!-- MD5(3A84D6F8:8b7a0a80-5283-486f-a4ec-40dc04aaa373:admin:admin:secret) -->
  <issuerSignature>79b81884ee34ab85207f96d986de23e3</issuerSignature>
</CSECOwnershipCookie>
```

Or, using the MaxAge parameter (with an expiry of 8 hours/28800 seconds):

```xml
<?xml version="1.0" encoding="utf-8" ?>
<CSECOwnershipCookie version="1.0" xmlns="urn:psialliance-org">
  <ownerCode>3A84D6F8</ownerCode>
  <maxage>28800</maxage>
  <path>/PSIA</path>
  <cookieVersion>1</cookieVersion>
  <!-- MD5(3A84D6F8:8b7a0a80-5283-486f-a4ec-40dc04aaa373:admin:admin:secret) -->
  <issuerSignature>79b81884ee34ab85207f96d986de23e3</issuerSignature>
</CSECOwnershipCookie>
```

Message Authentication Code (MAC) Algorithm

The **issuer-signature** will be a MAC of the **owner-code** value. The target device must apply the algorithm, with prescribed inputs, to the **owner-code** value to determine validity of the cookies. If the cookies are valid, then ownership is verified (i.e. the calculated Digest matches the **issuer-signature** value).
Note that SOURCE binders that are not present in the cookie itself protect it from sniffer-based (copy) attacks.
Ownership (cookie) Expiration

NOTE: RFC 6265 has superseded RFC 2965. It allows both the use of the absolute “Expires” header, and the alternate use of the newer “Max-Age” header. “Expires” is expressed in ‘dateTime’ format whereas “MaxAge” is expressed in seconds via an integer value. The “MaxAge” value is a delta from the current time whereas the (older but still acceptable) “Expires” specifies an absolute dateTime for expiry. Please reference RFC 6265 for more details.

The default duration of ownership (and cookie validity) is device and implementation specific. If the requesting administrative node does not specify one of the “ExpireTime”, or “Max-Age” QS values, then the device will automatically assign one. It is possible for the implementation to set the “expires” time to a near infinite time in future (i.e. effectively permanent ownership) or a shorter time, requiring a “refresh” of the ownership before the expiration time via a subsequent GET request. The more likely scenario (to avoid a deadlock situation) is that the device will set the “ExpireTime”, or “Max-Age”, value to sometime in the near future (seconds or minutes).

6.2.1 Resetting (Clearing) Device Ownership

To ability to reset Device Ownership state is provided to allow a management system to free a device from locked-out condition.

To clear the ownership state, a management node would use:
DELETE /PSIA/CSEC/deviceOwnership?ResetOwnership=true

This function will only succeed if the management node issued the request while logged into Device as a User belonging to an administrative Group that contained the symbolic Permission:
/ResetDeviceOwnership

For extra safety, the <PermissionDescriptor> that contains this symbolic permission could set the optional “loginPortAffinity” value to be a physically attached, serial terminal (e.g. “ttyS 000”).

6.2.2 Ownership Message Authentication Codes (MAC)

Currently the only MAC Algorithm defined is a simple MD5 Digest.

6.2.2.1 Simple Digest MAC – User ID Locked

This is a very simple approach provided as a reference in order to allow this Service to function minimally. Future (stronger) crypto approaches may be added in the future. CSEC can be somewhat agnostic to the crypto algorithms, since the generation and verification of the MAC code...
is done entirely within the Device implementation. However, exotic (unpublished) algorithms will result in incompatibility between different vendors’ systems and devices.

**owner-code** = BASE16 string representation of a generated random number (this is the public plain-text used to generated the cipher-text).


**OwnerGUID** = string representation of host’s PSIA Identity value (node GUID).

**user-id** = administrative client user ID.

**user-pw** = administrative client’s password or HA1 value.

**secret** = locally generated secret value (must be saved by Device for subsequent ownership verification).

**NOTE 1:**
A nonce and incrementing counter to prevent replay attack is unnecessary assuming **HTTP DIGEST authentication is employed**. The ownership cookie represents an ownership test applied in addition to DIGEST authentication. Without DIGEST authentication or SSL/TLS, an attacker with physical access to “sniff” in-use cookies can employ a spoofing attack. Security of the **user-pw** is critical for ensuring both valid authentication and ownership.

**NOTE 2:**
IP Address is not used here to avoid possible address change during DHCP address lease renew.

### 6.2.2.2 Simple Digest MAC – User ID Locked – Ownership Group Sharing

Since this form of Ownership Cookie is locked to an administrative User ID (with “Root of Security” privilege), any other PSIA node that logs-in with this User ID, while also presenting the Ownership Cookie, will be granted Ownership rights. In that sense the Cookie is transferable to another PSIA management node that also possesses knowledge of the User Password.

When the Cookie is transferred to another management node, the OwnerGUID value will stay the same. It continues to represent the unique ID of the original owner. The Device itself should log information regarding logins with ownership.
6.2.3 /PSIA/CSEC/deviceOwnership/scope

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/deviceOwnership/scope</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Resource used to manage the “ownership” scope definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td>GET</td>
<td>N/A</td>
<td>None</td>
<td>&lt;CSECOwnershipScope&gt;</td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>&lt;CSECOwnershipScope&gt;</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
</tbody>
</table>

This resource is used to control what “ownership” means, within the Device, in terms of Permission (and, thus, implied resources) that are to be controlled by the “owner”. The <CSECOwnershipScope> XML contains a <PermissionDescriptorList>, which contains an exhaustive list of the Permissions to be granted to the “owner” exclusively.

At any time where there is no recognized “owner” within the Device (i.e. the device is unclaimed by any qualified administrative nodes), then the normal User (PermissionGroup) permissions are enforced fully; however, if the device is owned by an administrative node, then any other Users’ may be denied their normally allowed (i.e. configured) Permissions, if these Permission fall within the scope of the ownership Permissions.

6.2.4 /PSIA/CSEC/deviceOwnership/status

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/deviceOwnership/status</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Resource used get the current “ownership” status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td>GET</td>
<td>N/A</td>
<td>None</td>
<td>&lt;CSECOwnershipStatus&gt;</td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
</tbody>
</table>

Notes
6.3 /PSIA/CSEC/deviceCertificates

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/deviceCertificates</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Resource used to manage the <code>&lt;DeviceCertificateList&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td><code>&lt;DeviceCertificateList&gt;</code></td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td><code>&lt;DeviceCertificateList&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>None</td>
<td><code>&lt;DeviceCertificateList&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
<tr>
<td>Notes</td>
<td>Example XML:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<DeviceCertificateList version="1.0" xmlns="urn:psialliance-org">
  <DeviceCertificate version="1.0">
    <id>1</id>
    <application>default</application>
    <CertificateDesc version="1.0">
      <CertificateFormat>PEM</CertificateFormat>
      <CertificateText>
        -----BEGIN CERTIFICATE-----
        MIIDBjCCAe4CCQX05m0b053QzANBgkqhkiG9w0BAQQQFADBMBMQwQQVYDQQGEweJB
        VTETEMBA1UEBhMKB29tZS1TdfG0ZTEhMBGBGA1UEChMyS50ZXJuZQq2L1Z10
        cyBQdHkgTHKMB4XDT4MTlwNzEwMjMyMl0XDT4MTlwNTEwMjMl0jowRTETMAKg
        A1UEBhMCQVUxEzARBgNVBAsVTETMBEGA1UECBMKS2FkZ2l0cyBQdHkg
        v regards
        I
        S
        T
        0
        1
        2
        3
        4
        5
        6
        7
        8
        9
        A
        B
        C
        D
        E
        F
        -----END CERTIFICATE-----
      </CertificateText>
    </CertificateDesc>
    <DeviceCertificate version="1.0">
      <id>2</id>
      <loginPortAffinity>eth 000</loginPortAffinity>
      <tcpPort>8554</tcpPort>
    </DeviceCertificate>
  </DeviceCertificate>
</DeviceCertificateList>
```
6.3.1 /PSIA/CSEC/deviceCertificates/<id>

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/deviceCertificates/&lt;id&gt;</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Resource used to manage a particular &lt;DeviceCertificate&gt;.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>None</td>
<td>&lt;DeviceCertificate&gt;</td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td>&lt;DeviceCertificate&gt;</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>None</td>
<td>None</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
</tbody>
</table>

Notes: See previous section for XML example.

6.3.2 /PSIA/CSEC/deviceCertificates/<id>/devCertificate

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/deviceCertificates/&lt;id&gt;/devCertificate</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>This Resource is used to load/update the Device’s SSL/TLS Server certificate in a direct or “raw” fashion.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>None</td>
<td>X.509v3/PEM (returns public/cert only)</td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td>X.509v3/PEM (public/cert &amp; private keys)</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
</tbody>
</table>

Notes: PEM: http://www.ietf.org/rfc/rfc1422.txt

Expected types declared in HTTP header(s):
“Content-Type: application/x-pem-file”

Others possibly supported in future:
“Content-Type: application/x-509-ca-cert” (DER)
“Content-Type: application/x-509-user-cert” (Netscape/Firefox-DER)
“Content-Type: application/x-pkcs12” (PKCS#12)
“Content-Type: application/pkix-cert” (DER) and “Content-Type: application/pkix-crl” (DER of Certificate Revocation List), see http://www.ietf.org/rfc/rfc2585.txt.
6.3.3 Certificate Formats

**PEM**

The PEM file can contain all of private keys (RSA and DSA), public keys (RSA and DSA) and (x509) certificates. It is the default format for OpenSSL. It stores data as Base64 encoded ASN1-DER format (surrounded by ascii headers) suitable for text mode transfers. Though there is a MIME type defined for this format, many applications will take PEM and DER as “application/x-x509-ca-cert”.

Example PEM file (contains certificate and private key):

```plaintext
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAwONaLOP7Edaqg1RUqKSDXzVMF3zBuBjHLElhN0o5L5sL4ieH6m5SCGCsV6y32zhqg51vTVaqeZ+xhcSBNZRu2c0mGhedRBWv/VA0jzp/xdMz5HaC
ErP3e9w1pkitVas8BF5zrgjO1nfCuGFCAi1fPavnutPq2K7jWMy2+g460Lnf6f1n
di4aTfTlRqAgXtUU6FpPfJPS35pkCkBT6508JSxxt/xf0eeZHCUS+U1wmGZyV64UB
W2q31f6bGAPihU3pdIo10uMPT/GUX+X4NzZJ4/lsnNaiwBdja4yhXKxejHeh
gqX3Bh5sV1HbZLi1x+414b9ge9w/XLioT6v5w3rdw1DAQBAObIAACFfLutmgQFbCRN
HAGJNMmmyz0vCUvnXTFyeEdXV67qxrYHLOe6LqIPxQ1oMonzO2kYMMvWooFAP
trOnSSzL+qaTYyDq27TKgo4uWhlZKyt8y7mdeExuaMSGKmpjla+tB51V+LxN
x1DEz4ZveFM3o2lycswt4qM5y404Yr3310E1rJfEn61lIdXiXyMhHHw8RkK
SadjYfboqc69f22Bolekgr5m5mJxyQymQoWAw5za5fLsynFdrJrFsb6z46G6
+L5GRyRVxfvxnVHynHPvCHeP40eiPv680wy2XPxV5KI8CidXAOARswSoGJa
N3nncu/EgCEYa6Y2mrM3JQEPFIALd8/L9Z1GBdsQiB2WSaxz1p9r/LdLF20H+oPits
to083om92kvpMrvD3SGxYDRZQ56t1FxYGCzJMBRicsZaHBo0bJv019n56LJyF
Bmju9yGvCsTgsrt3Bd0FryYp72CvWhHq1xVFpDc0J2o0r2uGkhnMNkgkGyEA03M6
WxPhs1dx3y6eVCShVhFbLSOqg810qiTe+y+pdVQcWN4C0/5eyAXXk1lCG9MmKxXy
Y+XgyzrDhfa20cxH5GSPKREA63bK2W8/pCasALWzKZpTifnSD99vtOt4WfFrG
+aQhDd5vAMfM9T0UqAvhc3vdjFUXrfeC7ntM8CyBGA+7t4ScbRhu70w70R/yKYF
kui3p3n9HFy/1+Mr13kEdz2if8FemZClVQVQf2dxW71jnq0z5vP1F3Fj3Dez7D
98w0uAcxeQvDRKV9g0Yo1oIeKpjeULsnnQefK+yuf0OsThXPXVY7jMfen6L6e6b
qN7t3q8vmXtejM9E3gCQuXBGhGR5W2Bl061slqKxl1KrPAv87BUE1HRCyu3ruA
6d1t7m7jLoEI7v242yZdSyanPqO6xv83T/YjL7tjAEbLM0+h+F80Eig0F1a
i1b8+Ac6uX1jkIvVv+p0NhbpFwP8grF5A5DpRk7pXmopvB88A1JlaI7Zv
BeHaHaorAgCAQ9WvCkt+nIP9fFxYbyYUsvgakl2zIqg66/13GryYaq8p1V5gvNv4v
ec0dW58JSJCPqoa3MNP78dEZeQ90oak+FsjBFzD2UkeRyet2/3qB/j2mN/+hEy
hYN0z1i2yTb63jGxKY6gH1R/r9d1kxAmjCzeZrSaAgywnteJWg=
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIEogIBAAKCAQEAwONaLOP7Edaqg1RUqKSDXzVMF3zBuBjHLElhN0o5L5sL4ieH6m5SCGCsV6y32zhqg51vTVaqeZ+xhcSBNZRu2c0mGhedRBWv/VA0jzp/xdMz5HaC
ErP3e9w1pkitVas8BF5zrgjO1nfCuGFCAi1fPavnutPq2K7jWMy2+g460Lnf6f1n
di4aTfTlRqAgXtUU6FpPfJPS35pkCkBT6508JSxxt/xf0eeZHCUS+U1wmGZyV64UB
W2q31f6bGAPihU3pdIo10uMPT/GUX+X4NzZJ4/lsnNaiwBdja4yhXKxejHeh
gqX3Bh5sV1HbZLi1x+414b9ge9w/XLioT6v5w3rdw1DAQBAObIAACFfLutmgQFbCRN
HAGJNMmmyz0vCUvnXTFyeEdXV67qxrYHLOe6LqIPxQ1oMonzO2kYMMvWooFAP
trOnSSzL+qaTYyDq27TKgo4uWhlZKyt8y7mdeExuaMSGKmpjla+tB51V+LxN
x1DEz4ZveFM3o2lycswt4qM5y404Yr3310E1rJfEn61lIdXiXyMhHHw8RkK
SadjYfboqc69f22Bolekgr5m5mJxyQymQoWAw5za5fLsynFdrJrFsb6z46G6
+L5GRyRVxfvxnVHynHPvCHeP40eiPv680wy2XPxV5KI8CidXAOARswSoGJa
N3nncu/EgCEYa6Y2mrM3JQEPFIALd8/L9Z1GBdsQiB2WSaxz1p9r/LdLF20H+oPits
to083om92kvpMrvD3SGxYDRZQ56t1FxYGCzJMBRicsZaHBo0bJv019n56LJyF
Bmju9yGvCsTgsrt3Bd0FryYp72CvWhHq1xVFpDc0J2o0r2uGkhnMNkgkGyEA03M6
WxPhs1dx3y6eVCShVhFbLSOqg810qiTe+y+pdVQcWN4C0/5eyAXXk1lCG9MmKxXy
Y+XgyzrDhfa20cxH5GSPKREA63bK2W8/pCasALWzKZpTifnSD99vtOt4WfFrG
+aQhDd5vAMfM9T0UqAvhc3vdjFUXrfeC7ntM8CyBGA+7t4ScbRhu70w70R/yKYF
kui3p3n9HFy/1+Mr13kEdz2if8FemZClVQVQf2dxW71jnq0z5vP1F3Fj3Dez7D
98w0uAcxeQvDRKV9g0Yo1oIeKpjeULsnnQefK+yuf0OsThXPXVY7jMfen6L6e6b
qN7t3q8vmXtejM9E3gCQuXBGhGR5W2Bl061slqKxl1KrPAv87BUE1HRCyu3ruA
6d1t7m7jLoEI7v242yZdSyanPqO6xv83T/YjL7tjAEbLM0+h+F80Eig0F1a
i1b8+Ac6uX1jkIvVv+p0NhbpFwP8grF5A5DpRk7pXmopvB88A1JlaI7Zv
BeHaHaorAgCAQ9WvCkt+nIP9fFxYbyYUsvgakl2zIqg66/13GryYaq8p1V5gvNv4v
ec0dW58JSJCPqoa3MNP78dEZeQ90oak+FsjBFzD2UkeRyet2/3qB/j2mN/+hEy
hYN0z1i2yTb63jGxKY6gH1R/r9d1kxAmjCzeZrSaAgywnteJWg=
-----END CERTIFICATE-----
```
DER (Distinguished Encoding Rules)

This is another common binary file format. It is a strict form of BER (Basic Encoding Rules). It can contain all of private keys, public keys and certificates. It is stored according to the ASN1-DER format. It is header-less - PEM is text header wrapped DER. It is the default format for many browsers.

CER (Canonical Encoding Rules)

This is another common binary file format. It is another strict form of BER used primarily to encode public key certificates (i.e. “Identity Certificates”). CA Root certificate packages are often distributed as a collection (zip) of PEMs or CERS.

PKCS#12 (aka PFX, or *.p12)

This is another common binary file format. It can contain all of private keys, public keys and certificates. It is supported by OpenSSL. It is the preferred format for MS IIS. MS IE can take DER as “application/x-x509-ca-cert”.

Certificate Format Compatibility

Ideally all CSEC devices and management software can agree on a single certificate format.

However, since that is not the case, and, since universal support for all formats is also not likely to be possible, a device is allowed to support one or more of the listed formats and reject other formats with the following caveat: If a binary format is supported for at the “raw” certificate resource (e.g., PUT /PSIA/CSEC/deviceCertificates/<id>/devCertificate), the public key information must be available in the BASE64 encoded text in the XML response at the parent resource (e.g. GET /PSIA/CSEC/deviceCertificates/<id>). Note that, of course, retrieving the certificate at the “raw” certificate resource (e.g. GET /PSIA/CSEC/deviceCertificates/<id>/devCertificate) should return the public certificate information in the same format that was given with the PUT, with the format also indicated in the returned mime type.
6.4 /PSIA/CSEC/certificateAuthority

| Function | Resource used to manage the <CertificateAuthorityList>.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
</tr>
<tr>
<td>POST</td>
<td>None</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Notes

Example XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<CertificateAuthorityList version="1.0" xmlns="urn:psialliance-org">
  <CertificateAuthority version="1.0">
    <id>1</id>
    <DistinguishedName>
      <RDN>
        <RDNLongLabel>OrganizationName</RDNLongLabel>
        <RDNValue>VeriSign Trust Network</RDNValue>
      </RDN>
      <RDN>
        <RDNLongLabel>OrganizationalUnitName</RDNLongLabel>
        <RDNValue>VeriSign, Inc.</RDNValue>
      </RDN>
      <RDN>
        <RDNLongLabel>OrganizationalUnitName</RDNLongLabel>
        <RDNValue>VeriSign International Server CA - Class 3</RDNValue>
      </RDN>
      <RDN>
        <RDNLongLabel>OrganizationalUnitName</RDNLongLabel>
        <RDNValue>www.verisign.com/CPS Incorp.by Reg. LIABILITY LTD.(c)97</RDNValue>
      </RDN>
      <CommonName><RDNLongLabel>VeriSign</RDNLongLabel>
        <RDNValue>VeriSign</RDNValue>
      </CommonName>
    </DistinguishedName>
    <crlDistributionPoint>http://crl.verisign.com/pca1.crl</crlDistributionPoint>
    <crlDistributionPoint>http://crl.verisign.com/pca2.crl</crlDistributionPoint>
    <RootPackages version="1.0">
      <RootCertPackage version="1.0">
        <id>1</id>
        <path>Universal</path>
        <CertificateDesc version="1.0">
```
-----BEGIN CERTIFICATE-----
MIIeTkAgICA6GwAwIBAgIQQBrEZCGzEyEDDrvkEhrFHTANBgkqhkiG9w0BAQsFADBv
TEMAKAgIBMCVMVFesAVbNVBA6MDl2cmlTaWduLJcbhM6Mr8sHQYDVQQQL
ExZzWzJpOU1nbiUCnVzdCBDBXZ3b3j0MTowQAYDVQQLEzE0YykMHjAwOCBWXJJP
U2lnbiqmgW5j1lAIteZc1IbXdb3JpemVkIHVzzBhcmxSNtCnWgYDVQQDExBy9W
ZXJpOU1nbiVBbm12ZjxYjWgwUm9vDBDZXJ0aWZpY2F0aW9uIE1dGhvcml6eTAe
Fw0oMdAwMDMwBDBwFw02NzeEmYDEMyMzU5NT1aMIG9M0saWQCDQQQGZEEvJUZr
MBUHA1UECHMWNyVgVNP2Z4asiEluu4X4hZAdBgNVBAAsTFlcm1TaWduIFRYdXN0
IE15dHdcmsxojABgNVBA5eTMSmjkSAyM4DA4IFZcm1TaWduLJcbhM6Mr8sQg9mY
IGF1dGhvcml6ZWQ4dXlG9ubHkxoDA2BgNVAMTRlZcm1TaWduIFVuaXZlcnNh
bcBSBz90IENiNcnPzmljXPrpb24gQXV0aG9y5XRMiIBjANWtkgkqkiG9owBAQAF
AA4ACgQAMc4AMCAQ8AMIIBCgKCAQEAx2E3XrEBNNniixWb/1/i4jeCmCokdeQMnW5lZ1iF
9UvKhbWnbnA9UA4Ig6c6XHaXGVMk2sk8sQKNeogk4k1XK9C1fKNQPMJ1MSgEWA
H62MFMFhWIFFEXBFVpa+r3HPDeE5Y2A7CsOWTwCzmnhcrea3ekEzeEO4VMQnGnH
L17g9f4duW/h2JKXeb36x45HVMeEF6nFaucsLiU0cry6x5XcZQj/5Q8kTkjiFJN
/JRMeYctTTI1TZN18AFMQXQYxtrXhPma5hqqG27T91uguHW7wqkMvku1D5fjtR
Y9jmV2ZGUvG59ycrtFw9KrWtaOljQIDAQABo4GyMGW1GvMA8Ad1Ew
EwEB/wQFMAMBfa8d9wVhYDR0PAQP8/BAQDADgEMGSSCCjxQAGUFwEBMGEWEwXFA6os
WTBXMFHlW1tYl/p2dpj9Ahm8bw8WYFkwAdhEFI/1/0xqGRIa0g8PsgQ5G5B5s
exkuMCWiuZ2hDHA6ly9eb2dvLmZcm1AaUdumBVCv282222LmmB0G1Ad4
DqGQcBS2d/psswEufuxXVuomMnyYO2CHGTSBNGykhiG9w0BAQAFAA4ACgQAMc4AMCAQ8
aniCQX8MvLzC2ytwuj0yli3cLkl1UXBY/kmyjDjEod20pzm12MWPsExE9rRjQj
+zeQXlcaB1vZadRhc1LMGmMazxY4sU3B1alkrEvyh9lH1eqEFmDQmDxLzPCPly+0
4Y8JDDnp/PvKrQvdBdEx56N9Srky9ahfyYtXp/jqgdFcrGJ2BtMQp2sXQrDrb2+
BxHw1ixdS5zwq+2X4+/vqGqzvdtdq46etwXDpjaP+PwGz2Y6r2aq91H8
1RQ0fc2VNNnS3BzgXucfr2YydHy75qeuxGM6Y1v/D/wIGv0vBZIGcF4kmJO3
?M2CYFE45+xMkDB--------------END CERTIFICATE-------------
</CertificateDesc>
</RootCertPackage>

<RootCertPackage version="1.0">
<id>2</id>
</RootCertPackage>

<path>G1=G5</path>
</RootCertPackageDesc version="1.0">
</CertificateDesc>
</CertificateFormat>

-----BEGIN CERTIFICATE-----
MIICCTCAA6GwAwIBAgIQQBrEZCGzEyEDDrvkEhrFHTANBgkqhkiG9w0BAQsFADBv
TEMAKAgIBMCVMVFesAVbNVBA6MDl2cmlTaWduLJcbhM6Mr8sHQYDVQQQL
ExZzWzJpOU1nbiUCnVzdCBDBXZ3b3j0MTowQAYDVQQLEzE0YykMHjAwOCBWXJJP
U2lnbiqmgW5j1lAIteZc1IbXdb3JpemVkIHVzzBhcmxSNtCnWgYDVQQDExBy9W
ZXJpOU1nbiVBbm12ZjxYjWgwUm9vDBDZXJ0aWZpY2F0aW9uIE1dGhvcml6eTAe
Fw0oMdAwMDMwBDBwFw02NzeEmYDEMyMzU5NT1aMIG9M0saWQCDQQQGZEEvJUZr
MBUHA1UECHMWNyVgVNP2Z4asiEluu4X4hZAdBgNVBAAsTFlcm1TaWduIFRYdXN0
IE15dHdcmsxojABgNVBA5eTMSmjkSAyM4DA4IFZcm1TaWduLJcbhM6Mr8sQg9mY
IGF1dGhvcml6ZWQ4dXlG9ubHkxoDA2BgNVAMTRlZcm1TaWduIFVuaXZlcnNh
bcBSBz90IENiNcnPzmljXPrpb24gQXV0aG9y5XRMiIBjANWtkgkqkiG9owBAQAF
AA4ACgQAMc4AMCAQ8AMIIBCgKCAQEAx2E3XrEBNNniixWb/1/i4jeCmCokdeQMnW5lZ1iF
9UvKhbWnbnA9UA4Ig6c6XHaXGVMk2sk8sQKNeogk4k1XK9C1fKNQPMJ1MSgEWA
H62MFMFhWIFFEXBFVpa+r3HPDeE5Y2A7CsOWTwCzmnhcrea3ekEzeEO4VMQnGnH
L17g9f4duW/h2JKXeb36x45HVMeEF6nFaucsLiU0cry6x5XcZQj/5Q8kTkjiFJN
/JRMeYctTTI1TZN18AFMQXQYxtrXhPma5hqqG27T91uguHW7wqkMvku1D5fjtR
Y9jmV2ZGUvG59ycrtFw9KrWtaOljQIDAQABo4GyMGW1GvMA8Ad1Ew
EwEB/wQFMAMBfa8d9wVhYDR0PAQP8/BAQDADgEMGSSCCjxQAGUFwEBMGEWEwXFA6os
WTBXMFHlW1tYl/p2dpj9Ahm8bw8WYFkwAdhEFI/1/0xqGRIa0g8PsgQ5G5B5s
exkuMCWiuZ2hDHA6ly9eb2dvLmZcm1AaUdumBVCv282222LmmB0G1Ad4
DqGQcBS2d/psswEufuxXVuomMnyYO2CHGTSBNGykhiG9w0BAQAFAA4ACgQAMc4AMCAQ8
aniCQX8MvLzC2ytwuj0yli3cLkl1UXBY/kmyjDjEod20pzm12MWPsExE9rRjQj
+zeQXlcaB1vZadRhc1LMGmMazxY4sU3B1alkrEvyh9lH1eqEFmDQmDxLzPCPly+0
4Y8JDDnp/PvKrQvdBdEx56N9Srky9ahfyYtXp/jqgdFcrGJ2BtMQp2sXQrDrb2+
BxHw1ixdS5zwq+2X4+/vqGqzvdtdq46etwXDpjaP+PwGz2Y6r2aq91H8
1RQ0fc2VNNnS3BzgXucfr2YydHy75qeuxGM6Y1v/D/wIGv0vBZIGcF4kmJO3
?M2CYFE45+xMkDB--------------END CERTIFICATE-------------
<RDN>
  <RDNLongLabel>CommonName</RDNLongLabel>
  <RDNValue>Thawte Server CA/emailAddress=server-certs@thawte.com</RDNValue>
</RDN>

Note that a <CertificateAuthority> can be created without initially specifying the <RootPackages>. Once the root packages are accumulated in a large continuous block (e.g. PEMs), they can be updated into the device using a single “PUT /PSIA/CSEC/certificateAuthority/<id>/rootPackages”.

After the PUT is used to update the ‘rootPackages’ resource, a subsequent “GET /PSIA/CSEC/certificateAuthority/<id>” will return the <CertificateAuthority> XML which contains the updated root package information. It is up to the device to decide the format of the returned information (i.e. in the simplest form, it can just return the continuous block of PEMs that was sent with the PUT, encapsulated in a <RootPackages> with a single <RootCertPackage>).

6.4.1 /PSIA/CSEC/certificateAuthority/<id>

<table>
<thead>
<tr>
<th>URI Function</th>
<th>/PSIA/CSEC/certificateAuthority/&lt;id&gt;</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET Query String(s)</td>
<td>None</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td>PUT Inbound Data</td>
<td>None</td>
<td>&lt;CertificateAuthority&gt;</td>
<td>None</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE Inbound Data</td>
<td>None</td>
<td>None</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
</tbody>
</table>

Notes See previous section for XML example.
6.4.2 /PSIA/CSEC/certificateAuthority/<id>/rootPackages

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/certificateAuthority/&lt;id&gt;/rootPackages</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Resource used to manage a particular &lt;CertificateAuthority&gt;’s Root Certificate packages (series of Base64 CER or PEM) in a “raw” fashion.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td>Block of all Base64CERs/PEMs</td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td>Block of all Base64CERs/PEMs</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code &gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>None</td>
<td>None</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
</tbody>
</table>

Notes

6.5 /PSIA/CSEC/KeyManager

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/KeyManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Service</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes</td>
<td></td>
<td></td>
<td>Common Security (CSEC) Service’s KeyManager Service</td>
</tr>
</tbody>
</table>

The KeyManager Service exists to manage key management schemes and protocols supported by CSEC, though only one scheme is currently defined (MIKEY).

As an alternative to the key negotiation methods, the keys can be directly set using the /PSIA/CSEC/KeyManager/directMKIKeyList Resource.

The keys are defined as a set of “master” and “salt” values. The intent is to use these keys with SRTP’s Key Derivation Function (KDF), as defined in RFC 3711, section 7.1. Of course, the keys can be used with other security protocols in the future, but, at this time, SRTP is the only protocol defined.

6.5.1 /PSIA/CSEC/KeyManager/directMKIKeyList

<table>
<thead>
<tr>
<th>URI</th>
<th>[HTTPS ONLY] /PSIA/CSEC/KeyManager/directMKIKeyList</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Resource</td>
</tr>
</tbody>
</table>

| Function                             | On a secured TLS/SSL session: This Resource is used to setup/create secret (symmetric) master keys to be used in future (streaming) sessions (e.g. SRTP/SRTCP). As such, it bypasses the normal KeyManager’s key negotiation/agreement schemes. This resource serves the same function as, though it is not identical to, the IPMD(v1.1)’s /System/Security/srtpMasterKey. |

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td>&lt;MKIKeyList&gt;</td>
</tr>
</tbody>
</table>
PUT | None | <MKIKeyList> | <ResponseStatus>
POST | N/A | <MKIKey> | <ResponseStatus>
DELETE | N/A | N/A | <ResponseStatus w/error code>

Notes: Note that, for POST, the newly created <MKI> value will be returned in the <ID> tag of the <ResponseStatus> XML.

Example XML:

```xml
<MKIKeyList version="1.0" xmlns="urn:psialliance-org">
  <MKIKey version="1.0">
    <!-- Restrict to 32bit integer: -->
    <MKI>10</MKI>
    <MKIMediaContext>
      <!-- OPTIONAL: -->
      <StreamingChannelID>0</StreamingChannelID>
      <!-- OPTIONAL: -->
      <StreamingMediaType>video</StreamingMediaType>
      <!-- REQUIRED, if re-keying: -->
      <rtsSessionID>2605004428</rtsSessionID>
      <!-- OPTIONAL (only needed if re-keying different MKI for each RTP Session): -->
      <ssrcID>1D623AC5</ssrcID>
    </MKIMediaContext>
    <!-- Base64 of '1234567890abcdef1234567890abcdef': -->
    <masterKey>MTIzNDU2Nzg5MzFiY2RlZjEyMzQ1Njc4OTBhYmNkZWIy</masterKey>
    <!-- Base64 of '3b04803de51ee7c96423ab5b78d2': -->
    <masterSalt>M2IwNDgwM2RlNTFlZTdjOTY0MjNhYjViNzhhMg==</masterSalt>
    <Cipher>AES_CM_128_HMAC_SHA1_80</Cipher>
    <!-- OPTIONAL - decimal units of pkt-count, must be 0 or power of 2: -->
    <srtpKeyDerivationRate>0</srtpKeyDerivationRate>
    <!-- OPTIONAL - decimal units of pkt-count (max/def for srtp is 2^48 = 2147483648): -->
    <srtpKeyLifetime>2147483648</srtpKeyLifetime>
    <!-- OPTIONAL - decimal units of pkt-count, must be 0 or power of 2: -->
    <srtpKeyDerivationRate>0</srtpKeyDerivationRate>
    <!-- OPTIONAL - decimal units of pkt-count (max/def for srtp is 2^31 = 2147483648): -->
    <srtpKeyLifetime>2147483648</srtpKeyLifetime>
    <!-- OPTIONAL - suggest next MKI when re-keying due to max-key-Lifetimes reached -->
    <keyRotationPreferenceMKI>11</keyRotationPreferenceMKI>
  </MKIKey>
  <MKIKey version="1.0">
    <!-- Restrict to 32bit integer: -->
    <MKI>11</MKI>
    <MKIMediaContext>
      <!-- OPTIONAL: -->
      <StreamingChannelID>0</StreamingChannelID>
      <!-- OPTIONAL: -->
      <StreamingMediaType>video</StreamingMediaType>
      <!-- REQUIRED, if re-keying: -->
      <rtsSessionID>2605004428</rtsSessionID>
      <!-- OPTIONAL (only needed if re-keying different MKI for each RTP Session): -->
      <ssrcID>1D623AC5</ssrcID>
    </MKIMediaContext>
  </MKIKey>
</MKIKeyList>
```
6.5.2 /PSIA/CSEC/KeyManager/directMKIKeyList/<MKI>

<table>
<thead>
<tr>
<th>URI</th>
<th>Function</th>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>/PSIA/CSEC/KeyManager/directMKIKeyList/&lt;MKI&gt;</td>
<td>On a secured TLS/SSL session: This Resource is used access a specific MKI context.</td>
<td>GET PUT</td>
<td>None</td>
<td>None</td>
<td>&lt;MKIKey&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>POST</td>
<td>N/A</td>
<td>&lt;MKIKey&gt;</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELETE</td>
<td>None</td>
<td>None</td>
<td>&lt;ResponseStatus&gt;</td>
</tr>
</tbody>
</table>

Notes: See previous section for example XML.

6.5.3 /PSIA/CSEC/KeyManager/directMKIKeyList/<MKI>/mime

<table>
<thead>
<tr>
<th>URI</th>
<th>Function</th>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>/PSIA/CSEC/KeyManager/directMKIKeyList/&lt;MKI&gt;/mime</td>
<td>On a secured TLS/SSL session: This Resource is used get the key and salt (concatentated) without XML encapsulation.</td>
<td>GET PUT POST DELETE</td>
<td>None</td>
<td>None</td>
<td>BASE64 of key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
</tbody>
</table>

Notes
6.5.4 /PSIA/CSEC/KeyManager/schemes

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/KeyManager/schemes</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
</tr>
<tr>
<td>Notes</td>
<td>Common Security (CSEC) Service’s key management schemes resource</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The children of this resource are indexed to discover what schemes (i.e. key management protocols) are supported by CSEC. The only currently supported scheme is MIKEY (see RFC 3830).

In the future, other schemes (e.g. ZRTP, GDOI, DTLS) may also be advertised, though these schemes may operate freely out-of-scope with PSIA protocols.

6.5.5 /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList

<table>
<thead>
<tr>
<th>URI</th>
<th>HTTPS RECOMMENDED</th>
<th>/PSIA/CSEC/KeyManager/schemes/negotiatedMKIList</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>This Resource is used get a list of negotiated MKI values.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
<td>Inbound Data</td>
<td>Return Result</td>
<td></td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td>&lt;NegotiatedMKIList&gt;</td>
<td></td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
<td></td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<NegotiatedMKIList version="1.0" xmlns="urn:psialliance-org">
  <NegotiatedMKI version="1.0">
    <MKI>1</MKI>
  </NegotiatedMKI>
  <NegotiatedMKI version="1.0">
    <MKI>2</MKI>
  </NegotiatedMKI>
  <NegotiatedMKI version="1.0">
    <MKI>3</MKI>
  </NegotiatedMKI>
  <NegotiatedMKI version="1.0">
    <MKI>4</MKI>
  </NegotiatedMKI>
  <NegotiatedMKI version="1.0">
    <MKI>5</MKI>
  </NegotiatedMKI>
</NegotiatedMKIList>
```
### 6.5.6 /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList/<MKI>

<table>
<thead>
<tr>
<th>URI</th>
<th>[HTTPS RECOMMENDED] /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList/&lt;MKI&gt;</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>This Resource used to get a particular MKI value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td><strong>Query String(s)</strong></td>
<td><strong>Inbound Data</strong></td>
<td><strong>Return Result</strong></td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td>&lt;NegotiatedMKI&gt;</td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td><strong>Notes</strong></td>
<td>See previous section for example XML.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6.5.7 /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList/<MKI>/mime

<table>
<thead>
<tr>
<th>URI</th>
<th>[HTTPS RECOMMENDED] /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList/&lt;MKI&gt;/mime</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>On a secured TLS/SSL session: This Resource is used get the key and salt (concatentated) without XML encapsulation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td><strong>Query String(s)</strong></td>
<td><strong>Inbound Data</strong></td>
<td><strong>Return Result</strong></td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td>BASE64 of key</td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td><strong>Notes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6.5.8 /PSIA/CSEC/KeyManager/schemes/MIKEY

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/KeyManager/schemes/MIKEY</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>This Resource is used to query the advertised MIKEY (see RFC 3830) resource properties.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td><strong>Query String(s)</strong></td>
<td><strong>Inbound Data</strong></td>
<td><strong>Return Result</strong></td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td>&lt;MIKEYProperties&gt;</td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td>&lt;ResponseStatus w/error code&gt;</td>
</tr>
<tr>
<td><strong>Notes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example XML:
The MIKEY protocol allows for two parties to exchange messages to establish a cryptographic context (keys) which is then given to another security protocol for its use. Using MIKEY terminology, there are 2 major types of keys: Traffic Encrypting Key (TEK), and TEK Generation Key (TGK). The MIKEY protocol also allows for update of an existing cryptographic context. This is a required feature to handle re-keying/key-rotation in cases of long-lived (e.g. live media streaming) sessions which use SRTP, since the master-key has a maximum lifetime (2^48 packets for SRTP, 2^31 for SRTCP).

The TEK is derived from the TGK. The TGK can also be used to derive other keys such as “authentication key”, “encryption key”, and “salting key”.

The TEK is communicated in the “Key data sub-payload”, which should also contain the MKI/SPI in the “KV Data” portion at the end. The resultant MKI(s) should be published in the “/PSIA/CSEC/KeyManager/schemes/negotiatedMKIList”.

For SRTP/SRTCP to function, only the TEK and, possibly, the “salting key” are required. Though the TEK is nominally intended by MIKEY to be used for traffic encryption, SRTP will, instead, use it as an input to its Key Derivation Function (KDF).

From RFC 3711:

```
| packet index |-+ master | session encr_key |
| | ext | key | session auth_key |
| | key_mgmt | | deriv |
| | (optional) | | session salt_key |
| | rekey | | |
| | master | |
| | |
| | salt | |
```

Figure 5: SRTP key derivation.

At least one initial key derivation SHALL be performed by SRTP, i.e., the first key derivation is REQUIRED. Further applications of the key derivation MAY be performed, according to the "key_derivation_rate" value in the cryptographic context. The key derivation function SHALL initially be invoked before the first packet and then, when r > 0, a key derivation is performed whenever
index mod r equals zero. This can be thought of as "refreshing" the session keys. The value of "key_derivation_rate" MUST be kept fixed for the lifetime of the associated master key.

6.5.9 /PSIA/CSEC/KeyManager/schemes/MIKEY/tunnel

<table>
<thead>
<tr>
<th>URI</th>
<th>[HTTPS ONLY] /PSIA/CSEC/KeyManager/schemes/MIKEY/tunnel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Resource</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>On a secured TLS/SSL session: This Resource is used to setup/create Crypto Session (CS) contexts using the MIKEY protocol (see RFC 3830) tunneled over the HTTP-REST.</td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td>Query String(s)</td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Notes**

With early implementations of MIKEY, the messages were typically transported within SDP/SIP. The SDP exchanges, carried the MIKEY message as a BASE64 encoded string within the SDP’s media attribute, “key-mgmt”, e.g.:

```
"a=key-mgmt:mikey AQQFgAAATbcCAAAAAAHK/AAAAAAAAAAAAAAAAAoAx9bH1P3ztk LAAAAJwABAQEBEAIABAQMBFAQBdUAAAYBAAcBAQgBAQjBAQoBAQoSAQsBCwBAAcQrp3V4S04/yprsxz2nytCQMC BpMwggaPMIEEd6ADAqE CAgkA8+z1SaxBJE4wDQYJKoZIhvcNAQEFBQAwgYxscCzAJB"
```

There are also proposed standards for transporting key management protocols, such as MIKEY, over RTSP using a combination of SDP and new RTSP headers (see RFC 4567).

GET & POST methods treated identically (POST support is for cases where the client platform’s programming interfaces disallows payloads on GET’s).

Example XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<MIKEYMessage version="1.0" xmlns="urn:psi:alliance-org">
  <MIKEYMediaContext>
    <!-- OPTIONAL: -->
    <StreamingChannelID>1</StreamingChannelID>
    <!-- OPTIONAL: -->
    <StreamingMediaType>video</StreamingMediaType>
    <!-- REQUIRED, if re-keying: -->
    <rtspSessionID>2605004428</rtspSessionID>
    <!-- OPTIONAL (only needed if re-keying different MKI for each RTP Session): -->
    <ssrcID>
      1D623AC5
    </ssrcID>
  </MIKEYMediaContext>
```
The <MKIMediaContext> is used to bind the resultant Crypto Context (MKI) to an existing streaming session for the purpose of re-keying during key rotation (i.e. after the lifetime has expired).

Within <MKIMediaContext>, the <rtspSessionID> should contain copy of string-value from the RTSP “Session:” header for an ongoing RTSP streaming session. This RTSP header is present following RTSP SETUP of the stream and continues to be present for request/response messages that control the stream.

The “Session” string is described as follows in RFC 2326, alphanumeric string representation of a random value (typically a decimal value string):

3.4 Session Identifiers

Session identifiers are opaque strings of arbitrary length. Linear white space must be URL-escaped. A session identifier MUST be chosen randomly and MUST be at least eight octets long to make guessing it more difficult. (See Section 16.)

session-id = 1*( ALPHA | DIGIT | safe )

Within the <MKIMediaContext>, the <ssrcID> should contain a copy of the “ssrc=XXXXXXXX” hex-string from the “Transport:” header. This header value only present in the response message to a RTSP SETUP request.

The “ssrc” string is described as follows in RFC 2326, as an 8 character hex-string (i.e. BASE16):

ssrc:
The ssrc parameter indicates the RTP SSRC [24, Sec. 3] value that should be (request) or will be (response) used by the media server. This parameter is only valid for unicast transmission. It identifies the synchronization source to be associated with the media stream.

Transport = "Transport" ::
1#transport-spec
transport-spec = transport-protocol/profile[/lower-transport]*parameter
transport-protocol = "RTP"
profile = "AVP"
lower-transport = "TCP" | "UDP"
parameter = { "unicast" | "multicast" } |
| ; "destination" [ =" address ] |
| ; "interleaved" =" channel [ =" channel ] |
| ; "append" |
| ; "ttl" =" ttl |
| ; "layers" =" 1*DIGIT |
| ; "port" =" port [ =" port ] |
| ; "client_port" =" port [ =" port ] |
| ; "server_port" =" port [ =" port ] |
| ; "ssrc" =" ssrc
Example SETUP response message:

```
RTSP/1.0 200 OK
CSeq: 320
Session: 2605004428
Transport: RTP/AVP;unicast;client_port=3724-3725;source=10.2.100.59;server_port=6970-6971;ssrc=1D623AC5
```
6.6 /PSIA/CSEC/AAA

Model Summary

CSEC’s User and Permission/Entitlement Model can roughly be summarized as a Resource-based RBAC (Role Based Access Control) System. The Roles are described by Permission Groups. Users belong to Groups. The Resources that are managed are primarily PSIA-REST Resources. Physical resources, for which there is no representation via the PSIA-REST Resource model, are managed as device-specific “Restrictions”.

Permissions

The default assumption is that all Users are not allowed to execute actions and not entitled to access any resource until granted by it Group definition. For simplicity, there are no negative permissions defined (negating requires removal of a positive permission).

The inherent Permission design of CSEC is based on symbolic or abstract resources (e.g. “Video” or “Fire Zone”) that are specific to the security industry. The actions allowed against these resources are also symbolic (e.g. “Stream the Video” or “Arm the Fire Zone”). CSEC uses the URI path syntax to codify these symbolic permissions into text strings. It is also (optionally) possible to assign binary (i.e. bit) values, within a permission mask, to the symbolic permission, which would allow for testing and manipulation via binary math operations.

An alternate Permission definition is allowed via "ExplicitPermissionDescriptor", which allows for the definition of a Permission as CRUDIE (Create, Read, Update, Delete, Index, Export) against a REST Resource (xs:anyURI).

Groups

A Permission Group collects Permissions within its "PermissionDescriptorList". A Permission Group is identified by its local REST ID ("id") or its global ID ("groupGUID"). The Group is also the container for “Restrictions”.

Device Scope Restrictions

The “CSECPermissionGroup” optionally contains a "DeviceScopeRestrictionList". A scope restriction allows for containing the operating scope of the Group. The two types of restrictions that can be defined are the “local device” or a “global device identified by a GUID”. Within a device’s scope, the restriction can further be defined using symbolic restriction names (e.g. “Video ports 1-8 only”).

Each “DeviceScopeRestriction” item within the “DeviceScopeRestrictionList” can also opaquely encapsulate the permission information as defined by the Area Control Working Group (ACWG) in the element “ACWGPermissionInfoList”. These ACWG permissions manage access to physical resources (e.g. specific zones or doors) outside the scope of the defined PSIA-REST Resources.
The “DeviceScopeRestriction” can either be bound to the “local device” or a global GUID, meaning that a central management node can construct a Group with a list of “Restrictions”, each referring to different device on the network. When configuring a particular device, the central node has the option of sending the unaltered Group definition, which contain information pertaining to other devices, or sending a modified Group definition, which only contains information pertaining to the target device.

In this Permission Groups Example (defined from the perspective of central management):

- Group 1 represents the “Super User” for this security domain, as far as REST-based Permissions are concerned; however Group 1 does not carry ACWG (panel-specific physical permissions).
- Group 2 can only view “live” video from the 2 cameras.
- Group 3 can only view “live” or recorded media from the RaCM device.
- Group 4 can view any REST-based settings in the security domain.
• Group 5 represents the administrators for the Access Panels only. Members of this group possess all REST-based permissions and the highest ACWG authority level on the Access Panels only.

Note that the ACWG Permissions are carried in CSEC opaquely as “Restrictions”. As such, it is not possible to define a “Super Group” which can administer all devices while also possessing the highest ACWG authority level for each ACWG device. Users requiring such permissions must belong to both Group 1 and Group 5.

When programming these Group definitions into the target devices, they can be sent unaltered from the central server’s version if the target device is aware of its own Device GUID. However, since that is unlikely to be the case, the management node must translate the Group definition to “local” device scope.

/PSIA/CSEC and Profile Requirements

CSEC Profiles determine the functional operation of the CSEC AAA resources. All CSEC AAA resources are read-only for ‘Core’ profile implementations. All other profiles comply with the full definition of each resource. The tables below specifically detail the HTTP Methods allowed per each AAA resource, per profile.

<table>
<thead>
<tr>
<th>CORE Profile:</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/PSIA/CSEC/AAA/users</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/AAA/users/&lt;id&gt;</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/AAA/users/&lt;GUID&gt;</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/permissionGroups</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/permissionGroups/&lt;id&gt;</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/supportedPermissions</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BASIC and FULL Profiles:</th>
<th>Resource</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/PSIA/CSEC/AAA/users</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/AAA/users/&lt;id&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/AAA/users/&lt;GUID&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/permissionGroups</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/permissionGroups/&lt;id&gt;</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/PSIA/CSEC/supportedPermissions</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 7.6.1 /PSIA/CSEC/AAA/users

**URI**

[HTTPS ONLY]

/PSIA/CSEC/AAA/users

**Type**

Resource

**Function**

Resource used to manage the `<CSECUserList>`.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td><code>&lt;CSECUserList&gt;</code></td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td><code>&lt;CSECUserList&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>None</td>
<td><code>&lt;CSECUser&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>DELETE</td>
<td>N/A</td>
<td>N/A</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
</tbody>
</table>

**Notes**

Core profile nodes return ‘405 Method Not Allowed’ for all PUT, POST, and DELETE methods against this resource. Basic and Full nodes return ‘405 Method Not Allowed’ for DELETE operations.

Example XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<CSECUserList version="1.0" xmlns="urn:psialliance-org">
  <CSECUser version="1.0">
    <id>1</id>
    <userEnabled>true</userEnabled>
    <userDescription>administrator</userDescription>
    <fullName>John Doe</fullName>
    <userGUID>{cbe45517-4df1-4c5a-8290-43d883d099e2}</userGUID>
    <UserLogin>
      <username>admin</username>
      <password>*</password>
    </UserLogin>
    <UserCODEList>
      <UserCODE>
        <id>1</id>
        <CODE>191155</CODE>
        <CODEClass>INTRUSION</CODEClass>
      </UserCODE>
      <UserCODE>
        <id>2</id>
        <CODE>236722</CODE>
        <CODEClass>ACCESS</CODEClass>
      </UserCODE>
      <UserCODE>
        <id>3</id>
        <CODE>77777</CODE>
        <CODEClass>FIRE</CODEClass>
      </UserCODE>
    </UserCODEList>
    <UserPermissionGroupId>1</UserPermissionGroupId>
  </CSECUser>
</CSECUserList>
```
<CSECUser version="1.0">
  <id>2</id>
  <userEnabled>true</userEnabled>
  <userDescription>guard</userDescription>
  <fullName>Jack Black</fullName>
  <userGUID>{16d5c215-505b-48f1-a2de-557c2f43ea8e}</userGUID>
  <UserLogin>
    <username>guard123</username>
    <password>*</password>
  </UserLogin>
  <UserCodeList>
    <UserCode>
      <id>1</id>
      <CODE>123456</CODE>
      <CODEClass>INTRUSION</CODEClass>
    </UserCode>
    <UserCode>
      <id>2</id>
      <CODE>333333</CODE>
      <CODEClass>ACCESS</CODEClass>
    </UserCode>
  </UserCodeList>
  <UserPermissionGroupId>2</UserPermissionGroupId>
</CSECUser>

<CSECUser version="1.0">
  <id>3</id>
  <userEnabled>true</userEnabled>
  <userDescription>view only</userDescription>
  <UserLogin>
    <username>viewonly</username>
    <password>*</password>
  </UserLogin>
  <UserPermissionGroupId>3</UserPermissionGroupId>
</CSECUser>

<CSECUser version="1.0">
  <id>4</id>
  <userEnabled>true</userEnabled>
  <userDescription>DURESS</userDescription>
  <UserCodeList>
    <UserCode>
      <id>1</id>
      <CODE>999999</CODE>
      <CODEClass>INTRUSION</CODEClass>
    </UserCode>
    <UserCode>
      <id>2</id>
      <CODE>999999</CODE>
      <CODEClass>ACCESS</CODEClass>
    </UserCode>
    <UserCode>
      <id>3</id>
      <CODE>999999</CODE>
      <CODEClass>FIRE</CODEClass>
    </UserCode>
  </UserCodeList>
</CSECUser>
6.6.1.1/PSIA/CSEC/AAA/users/<id>

<table>
<thead>
<tr>
<th>Function</th>
<th>Resource used to manage a particular &lt;CSECUser&gt; identified by its local REST ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>Query String(s)</td>
</tr>
<tr>
<td>GET</td>
<td>None</td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
</tr>
<tr>
<td>DELETE</td>
<td>None</td>
</tr>
</tbody>
</table>

Notes: See previous section for example XML. Core profile nodes return ‘405 Method Not Allowed’ for all PUT, POST, and DELETE methods against this resource. Basic and Full profile nodes return ‘405 Method Not Allowed’ for POST operations.
6.6.1.2/PSIA/CSEC/AAA/users/<GUID>

<table>
<thead>
<tr>
<th>URI</th>
<th>HTTPS ONLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>/PSIA/CSEC/AAA/users/&lt;GUID&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Function**: Resource used to manage a particular `<CSECUser>` identified by its User GUID.

**Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td><code>&lt;CSECUser&gt;</code></td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td><code>&lt;CSECUser&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
<tr>
<td>DELETE</td>
<td>None</td>
<td>None</td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
</tbody>
</table>

**Notes**: See previous section for example XML. Core profile nodes return ‘405 Method Not Allowed’ for all PUT, POST, and DELETE methods against this resource. Basic and Full nodes return ‘405 Method Not Allowed’ for POST operations.

6.6.1.3 Required Special Users and Groups

6.6.1.3.1 “admin”

For usability and interoperability purposes, a user named “admin” (along with membership to a Group with matching name, “admin”) is REQUIRED to exist within the CSEC Device. The “admin” Group MUST be granted Permissions to fully configure the CSEC Device.

6.6.1.3.2 “viewonly”

For usability and interoperability purposes, a user named “viewonly” (along with matching membership to a Group named “viewonly”) is REQUIRED to exist within the CSEC Device. The “viewonly” Group MUST be granted Permissions to ONLY view/stream media (i.e. MUST have only “read-only” capabilities and MUST not have any “write” capability within the CSEC Device).

6.6.2 /PSIA/CSEC/AAA/permissionGroups

<table>
<thead>
<tr>
<th>URI</th>
<th>/PSIA/CSEC/AAA/permissionGroups</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
</table>

**Function**: Resource used to manage the `<CSECPermissionGroupList>`.

**Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td><code>&lt;CSECPermissionGroupList&gt;</code></td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td><code>&lt;CSECPermissionGroupList&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>None</td>
<td><code>&lt;CSECPermissionGroup&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
</tbody>
</table>
Core profile nodes return ‘405 Method Not Allowed’ for all PUT, POST, and DELETE methods against this resource. Basic and Full nodes return ‘405 Method Not Allowed’ for DELETE operations.

Example XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<CSECPermissionGroupList version="1.0" xmlns="urn:psialliance-org">
  <CSECPermissionGroup version="1.0">
    <id>1</id>
    <groupEnabled>true</groupEnabled>
    <groupDescription>administrators</groupDescription>
    <PermissionDescriptorList version="1.0">
      <PermissionDescriptor>
        <id>1</id>
        <SymbolicPermissionDescriptor>
          <PermissionDictionaryDescriptor>
            <dictionaryName>PSIA-CSEC-D1</dictionaryName>
            <!-- SymbolicPermission '/' means it has ALL PERMISSIONS -->
            <SymbolicPermission>/</SymbolicPermission>
          </PermissionDictionaryDescriptor>
        </SymbolicPermissionDescriptor>
      </PermissionDescriptor>
    </PermissionDescriptorList>
  </CSECPermissionGroup>
  <CSECPermissionGroup version="1.0">
    <id>2</id>
    <groupEnabled>true</groupEnabled>
    <groupDescription>guards A</groupDescription>
    <PermissionDescriptorList version="1.0">
      <PermissionDescriptor>
        <id>1</id>
        <SymbolicPermissionDescriptor>
          <PermissionDictionaryDescriptor>
            <dictionaryName>PSIA-CSEC-D1</dictionaryName>
            <SymbolicPermission>/View/Statistics</SymbolicPermission>
          </PermissionDictionaryDescriptor>
        </SymbolicPermissionDescriptor>
      </PermissionDescriptor>
      <PermissionDescriptor>
        <id>2</id>
        <SymbolicPermissionDescriptor>
          <PermissionDictionaryDescriptor>
            <dictionaryName>PSIA-CSEC-D1</dictionaryName>
            <SymbolicPermission>/View/Logs</SymbolicPermission>
          </PermissionDictionaryDescriptor>
        </SymbolicPermissionDescriptor>
      </PermissionDescriptor>
    </PermissionDescriptorList>
  </CSECPermissionGroup>
</CSECPermissionGroupList>
```
<PermissionDescriptor>
  <id>3</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/Stream/Live/Media/Video</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>4</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/Stream/Recorded/Media/Video</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>5</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/Search/Media/Video</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>6</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/Search/Metadata/Analytics/Video</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

</PermissionDescriptorList>
</CSECPermissionGroup>

<CSECPermissionGroup version="1.0">
  <id>3</id>
  <groupEnabled>true</groupEnabled>
  <groupDescription>viewers</groupDescription>
  <PermissionDescriptorList version="1.0">
    <PermissionDescriptor>
      <id>1</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/Stream/Live/Media/Video</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
  </PermissionDescriptorList>
</CSECPermissionGroup>
<PermissionDescriptor>
  <id>2</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/Stream/Recorded/Media/Video</SymbolicPermission>
    </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
</PermissionDescriptor>

</PermissionDescriptorList>
<!-- RESTRICITION: local ports 9-16 only for this Group -->
<DeviceScopeRestrictionList>
  <DeviceScopeRestriction>
    <id>1</id>
    <restrictionDescription>Restrict access to Local Video Ports 9-16</restrictionDescription>
    <type>Local</type>
    <subDevVideoInPortRestriction>9-16</subDevVideoInPortRestriction>
    <subDevStreamChannelRestriction>9-16</subDevStreamChannelRestriction>
  </DeviceScopeRestriction>
</DeviceScopeRestrictionList>
</CSECPermissionGroup>

<CSECPermissionGroup version="1.0">
  <id>4</id>
  <groupEnabled>true</groupEnabled>
  <groupDescription>DURESS</groupDescription>
  <PermissionDescriptorList version="1.0">
    <PermissionDescriptor>
      <id>1</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>

  </PermissionDescriptorList>
<!-- no <DeviceScopeRestrictionList> -->
</CSECPermissionGroup>

<CSECPermissionGroup version="1.0">
  <id>5</id>
  <groupEnabled>true</groupEnabled>
  <groupDescription>Access Panel Group</groupDescription>
  <PermissionDescriptorList version="1.0">
    <PermissionDescriptor>
      <id>1</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
  </PermissionDescriptorList>
</CSECPermissionGroup>
<DeviceScopeRestrictionList>
  <DeviceScopeRestriction>
    <id>1</id>
    <restrictionDescription>Panel Portal 5</restrictionDescription>
    <type>Local</type>
    <ACWGPPermissionInfoList>
      <PermissionInfo>
        <ID>1</ID>
        <PrivilegeList>
          <Privilege>
            <Allow>
              <AuthorityLevel>1</AuthorityLevel>
              <PortalIDList>
                <PortalID>
                  <ID>1</ID>
                </PortalID>
                <ID>2</ID>
                <ID>3</ID>
              </PortalIDList>
            </Allow>
          </Privilege>
        </PrivilegeList>
        </PermissionInfo>
      </ACWGPPermissionInfoList>
    </DeviceScopeRestriction>
  </DeviceScopeRestrictionList>
</CSECPermissionGroupList>
6.6.2.1/PSIA/CSEC/AAA/permissionGroups/<id>

<table>
<thead>
<tr>
<th>Function</th>
<th>URI</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource used to manage a particular <code>&lt;CSECPermissionGroup&gt;</code></td>
<td>/PSIA/CSEC/AAA/permissionGroups/&lt;id&gt;</td>
<td>Resource</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td><code>&lt;CSECPermissionGroup&gt;</code></td>
</tr>
<tr>
<td>PUT</td>
<td>None</td>
<td><code>&lt;CSECPermissionGroup&gt;</code></td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
<tr>
<td>DELETE</td>
<td>None</td>
<td>None</td>
<td><code>&lt;ResponseStatus&gt;</code></td>
</tr>
</tbody>
</table>

| Notes | See previous section for example XML. Core profile nodes return ‘405 Method Not Allowed’ for all PUT, POST, and DELETE methods against this resource. Basic and Full nodes return ‘405 Method Not Allowed’ for POST operations. |

6.6.3 /PSIA/CSEC/AAA/supportedPermissions

<table>
<thead>
<tr>
<th>Function</th>
<th>URI</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource used get a <code>&lt;PermissionDescriptorList&gt;</code> of the supported permissions in the Device.</td>
<td>/PSIA/CSEC/AAA/supportedPermissions</td>
<td>Resource</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Query String(s)</th>
<th>Inbound Data</th>
<th>Return Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>None</td>
<td>None</td>
<td><code>&lt;PermissionDescriptorList&gt;</code></td>
</tr>
<tr>
<td>PUT</td>
<td>N/A</td>
<td>N/A</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
<tr>
<td>POST</td>
<td>N/A</td>
<td>N/A</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
<tr>
<td>DELETE</td>
<td>None</td>
<td>None</td>
<td><code>&lt;ResponseStatus w/error code&gt;</code></td>
</tr>
</tbody>
</table>

| Notes | Core profile nodes return ‘405 Method Not Allowed’ for all PUT, POST, and DELETE methods against this resource. Basic and Full nodes return ‘405 Method Not Allowed’ for PUT and POST operations. |

Example XML (this is a partial list for exemplary purposes only, but actual Device would return a complete list of all supported Symbolic Permissions and Explicit Permissions in its local Dictionary):

```xml
<?xml version="1.0" encoding="utf-8" ?>
<PermissionDescriptorList version="1.0" xmlns="urn:psialliance-org">
    <PermissionDescriptor>
        <id>1</id>
        <SymbolicPermissionDescriptor>
            <PermissionDictionaryDescriptor>
                <dictionaryName>PSIA-CSEC-D1</dictionaryName>
                <SymbolicPermission>/</SymbolicPermission>
            </PermissionDictionaryDescriptor>
        </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
        <id>2</id>
```

PSIA CSEC v2 - 55 -
<SymbolicPermissionDescriptor>
  <PermissionDictionaryDescriptor>
    <dictionaryName>PSIA-CSEC-D1</dictionaryName>
    <SymbolicPermission>/View</SymbolicPermission>
  </PermissionDictionaryDescriptor>
  <SymbolicPermissionDescriptor>
    <id>3</id>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/Statistics</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
  <PermissionDescriptor>
    <id>4</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/Logs</SymbolicPermission>
      </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
  </PermissionDescriptor>
  <PermissionDescriptor>
    <id>5</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/System</SymbolicPermission>
      </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
  </PermissionDescriptor>
  <PermissionDescriptor>
    <id>6</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/System/NetworkSettings</SymbolicPermission>
      </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
  </PermissionDescriptor>
  <PermissionDescriptor>
    <id>7</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/System/IOSettings</SymbolicPermission>
      </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
  </PermissionDescriptor>
</SymbolicPermissionDescriptor>
Copyright PSIA

<id>8</id>
<SymbolicPermissionDescriptor>
<PermissionDictionaryDescriptor>
<dictionaryName>PSIA-CSEC-D1</dictionaryName>
<SymbolicPermission>/View/System/StorageSettings</SymbolicPermission>
</PermissionDictionaryDescriptor>
</SymbolicPermissionDescriptor>
<PermissionDescriptor>
<id>9</id>
<SymbolicPermissionDescriptor>
<PermissionDictionaryDescriptor>
<dictionaryName>PSIA-CSEC-D1</dictionaryName>
<SymbolicPermission>/View/System/SerialPortSettings</SymbolicPermission>
</PermissionDictionaryDescriptor>
</SymbolicPermissionDescriptor>
<PermissionDescriptor>
<id>10</id>
<SymbolicPermissionDescriptor>
<PermissionDictionaryDescriptor>
<dictionaryName>PSIA-CSEC-D1</dictionaryName>
<SymbolicPermission>/View/System/Diagnostics</SymbolicPermission>
</PermissionDictionaryDescriptor>
</SymbolicPermissionDescriptor>
<PermissionDescriptor>
<id>11</id>
<SymbolicPermissionDescriptor>
<PermissionDictionaryDescriptor>
<dictionaryName>PSIA-CSEC-D1</dictionaryName>
<SymbolicPermission>/View/System/TimeSettings</SymbolicPermission>
</PermissionDictionaryDescriptor>
</SymbolicPermissionDescriptor>
<PermissionDescriptor>
<id>12</id>
<SymbolicPermissionDescriptor>
<PermissionDictionaryDescriptor>
<dictionaryName>PSIA-CSEC-D1</dictionaryName>
<SymbolicPermission>/View/VideoDevice</SymbolicPermission>
</PermissionDictionaryDescriptor>
</SymbolicPermissionDescriptor>
<PermissionDescriptor>
<id>13</id>
<SymbolicPermissionDescriptor>
<PermissionDictionaryDescriptor>
<dictionaryName>PSIA-CSEC-D1</dictionaryName>
<SymbolicPermission>/View/VideoDevice/Overlays</SymbolicPermission>
</PermissionDictionaryDescriptor>
</SymbolicPermissionDescriptor>
<PermissionDescriptor>
  <id>14</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/VideoDevice/Privacy Masks</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>15</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/VideoDevice/Other Settings</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>16</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/AudioDevice</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>17</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/AudioDevice/Microphone</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>18</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/AudioDevice/Other Settings</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>19</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/Codec</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>
<PermissionDescriptor>
  <id>20</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/Codec/Video</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>21</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/Codec/Audio</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>22</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/Codec/Other</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>23</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/PTZ</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>24</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/PTZ/PersistentPresets</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>

<PermissionDescriptor>
  <id>25</id>
  <SymbolicPermissionDescriptor>
    <PermissionDictionaryDescriptor>
      <dictionaryName>PSIA-CSEC-D1</dictionaryName>
      <SymbolicPermission>/View/PTZ/PersistentPresets</SymbolicPermission>
    </PermissionDictionaryDescriptor>
  </SymbolicPermissionDescriptor>
</PermissionDescriptor>
7.0 RTSP/SRTP Usage

To setup a secured data stream from the target PSIA Device, the MKI to be used must be created.

**MKI Creation**

To create a MKI, one of two mechanisms must be used:

A. On a TLS session, create a new MKI directly via (see section 6.5.1):
   POST /PSIA/CSEC/KeyManager/directMKIKeyList

B. On a TLS session, negotiate a new MKI via one of the supported key management schemes. Currently, only MIKEY is supported, which can be discovered via (see section 6.5.8):
   GET /PSIA/CSEC/KeyManager/schemes/index
   Perform the MIKEY PSK, PKE, or DH exchange via (see section 6.5.9):
   GET /PSIA/CSEC/KeyManager/schemes/MIKEY/tunnel
   If successful, the keys and MKI are returned in the “Key data sub-payload” or “DH data payload”.

**STREAMING**

After MKI crypto context has been created (and optionally bound to a particular streaming channel), the client must issue a RTSP DESCRIBE to the target device to get an SDP which describes the available data streams.

A special Query String will be used to request description of secured streams if available, using a particular MKI context, e.g.:

rtsp://10.2.100.62/PSIA/Streaming/channels/0?MKI=10

**SDP on secured RTSP**
Copyright PSIA

It is possible to describe all necessary security information (context) via RFC RFC 4568, however, this requires a secured (e.g. TLS) channel for RTSP. If a secured RTSP channel is available, a SDP similar to the following is returned in response to the RTSP DESCRIBE request:

```
v=0
o-- 1239724272167027 1239724272167027 IN IP4 10.2.100.62
s=Media Presentation
e=NONE
c=IN IP4 0.0.0.0
b=AS:50064
t=0 0
a=control:rtsp://10.2.100.62/PSIA/Streaming/channels/0
a=range:npt=0.000000-
m=video 0 RTP/AVP 96
b=AS:50000
a=control:1 AES_CM_128_HMAC_SHA1_80
inline:dORmdmcVCpeEc3QGiLNpVLFJhQX1cfHAWJSOj|2^48|10:4
a=framerate:30.0
a=control:rtsp://10.2.100.62/PSIA/Streaming/channels/0/video
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1; profile-level-id=420029; sprop-parameter-sets=Z0IAKeKQFAe2AtwEBAaQeJEV,aM48gA==
m=audio 0 RTP/AVP 0
b=AS:64
k=uri:https://10.2.100.62/PSIA/CSEC/KeyManager/directMKIKeyList/10/mime
a=control:rtsp://10.2.100.62/PSIA/Streaming/channels/0/audio
```

**SDP on unsecured RTSP**

If a secured RTSP channel is not available, then a standard SDP response may be used to convey enough information to indicate the available MKI, using the “k=” session description protocol (see RFC 2327, page 17), resulting in a SDP similar to the following:

```
v=0
o-- 1239724272167027 1239724272167027 IN IP4 10.2.100.62
s=Media Presentation
e=NONE
c=IN IP4 0.0.0.0
b=AS:50064
t=0 0
a=control:rtsp://10.2.100.62/PSIA/Streaming/channels/0
a=range:npt=0.000000-
m=video 0 RTP/AVP 96
b=AS:50000
k=uri:https://10.2.100.62/PSIA/CSEC/KeyManager/directMKIKeyList/10/mime
a=framerate:30.0
a=control:rtsp://10.2.100.62/PSIA/Streaming/channels/0/video
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1; profile-level-id=420029; sprop-parameter-sets=Z0IAKeKQFAe2AtwEBAaQeJEV,aM48gA==
m=audio 0 RTP/AVP 0
b=AS:64
k=uri:https://10.2.100.62/PSIA/CSEC/KeyManager/directMKIKeyList/10/mime
a=control:rtsp://10.2.100.62/PSIA/Streaming/channels/0/audio
```

In both SDP examples above, the client has enough information to prepare to receive the data streams on the prescribed cryptographic context (MKI 10). At this point, the RTSP SETUP(s) and PLAY can proceed as normal.
The SRTP data streams MUST contain the MKI trailer, which indicate the expected MKI requested by the client, but will also be used to indicate key-rotation to a new MKI, if applicable.

**KEY ROTATION (RE-KEYING)**

Similar to MKI Creation, MKI update mechanisms exist along the same path to change keys (i.e. allow for rotation), since their validity period is finite for SRTP/SRTCP.

A. On a TLS session, update a MKI directly via (see section 6.5.1):
   ```
   PUT /PSIA/CSEC/KeyManager/directMKIKeyList/<MKI>
   ```

B. On a TLS session, use one of the supported key management schemes to update a key-context. Perform the MIKEY PSK, PKE, or DH update-exchange via (see section 6.5.9):
   ```
   GET /PSIA/CSEC/KeyManager/schemes/MIKEY/tunnel
   ```

From RFC 3711:

11.3. Re-keying and access control

Re-keying may occur due to access control (e.g., when a member is removed during a multicast RTP session), or for pure cryptographic reasons (e.g., the key is at the end of its lifetime). When using SRTP default transforms, the master key MUST be replaced before any of the index spaces are exhausted for any of the streams protected by one and the same master key.

How key management re-keys SRTP implementations is out of scope, but it is clear that there are straightforward ways to manage keys for a multicast group. In one-sender multicast, for example, it is typically the responsibility of the sender to determine when a new key is needed. The sender is the one entity that can keep track of when the maximum number of packets has been sent, as receivers may join and leave the session at any time, there may be packet loss and delay etc. In scenarios other than one-sender multicast, other methods can be used. Here, one must take into consideration that key exchange can be a costly operation, taking several seconds for a single exchange. Hence, some time before the master key is exhausted/expires, out-of-band key management is initiated, resulting in a new master key that is shared with the receiver(s).

At this time, there is no method for the server (i.e. target PSIA Device) to force the client to re-negotiate keys, so it will be the responsibility of the client to re-key the MKI before master-key’s lifetime expires.

As an alternative, if “A” is the chosen mechanism, the `<keyRotationPreferenceMKI>` value can be used to prescribe a switch to new `<MKI>` at time of expiration. This would forestall (not eliminate) the need to perform an MKI update.

### 9.0 SRTP Crypto Tables

From RFC 4568:
From “srtp-big-aes-3”:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master key length</td>
<td>192 bits</td>
</tr>
<tr>
<td>Master salt length</td>
<td>112 bits</td>
</tr>
<tr>
<td>Key Derivation Function</td>
<td>AES_192_CM_PRF (Section 3)</td>
</tr>
<tr>
<td>Default key lifetime</td>
<td>2(^{31}) packets</td>
</tr>
<tr>
<td>Cipher (for SRTP and SRTCP)</td>
<td>AES_192_CM (Section 2)</td>
</tr>
<tr>
<td>SRTP authentication function</td>
<td>HMAC_SHA1 (Section 4.2.1 of [RFC3711])</td>
</tr>
<tr>
<td>SRTP authentication key length</td>
<td>160 bits</td>
</tr>
<tr>
<td>SRTP authentication tag length</td>
<td>80 bits</td>
</tr>
<tr>
<td>SRTCP authentication key length</td>
<td>160 bits</td>
</tr>
<tr>
<td>SRTCP authentication tag length</td>
<td>80 bits</td>
</tr>
</tbody>
</table>

Table 1: The AES_192_CM_HMAC_SHA1_80 cryptosuite.
Table 2: The AES_192_CM_HMAC_SHA1_32 cryptosuite.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master key length</td>
<td>256 bits</td>
</tr>
<tr>
<td>Master salt length</td>
<td>112 bits</td>
</tr>
<tr>
<td>Key Derivation Function</td>
<td>AES_256_CM_PRF (Section 3)</td>
</tr>
<tr>
<td>Default key lifetime</td>
<td>$2^{31}$ packets</td>
</tr>
<tr>
<td>Cipher (for SRTP and SRTCP)</td>
<td>AES_256_CM (Section 2)</td>
</tr>
<tr>
<td>SRTP authentication function</td>
<td>HMAC-SHA1 (Section 4.2.1 of</td>
</tr>
<tr>
<td></td>
<td>[RFC3711])</td>
</tr>
<tr>
<td>SRTP authentication key</td>
<td>160 bits</td>
</tr>
<tr>
<td>length</td>
<td></td>
</tr>
<tr>
<td>SRTP authentication tag</td>
<td>80 bits</td>
</tr>
<tr>
<td>length</td>
<td></td>
</tr>
<tr>
<td>SRTCP authentication</td>
<td>HMAC-SHA1 (Section 4.2.1 of</td>
</tr>
<tr>
<td>function</td>
<td>[RFC3711])</td>
</tr>
<tr>
<td>SRTCP authentication key</td>
<td>160 bits</td>
</tr>
<tr>
<td>length</td>
<td></td>
</tr>
<tr>
<td>SRTCP authentication tag</td>
<td>80 bits</td>
</tr>
<tr>
<td>length</td>
<td></td>
</tr>
</tbody>
</table>
### Table 3: The AES_256_CM_HMAC_SHA1_80 cryptosuite.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master key length</td>
<td>256 bits</td>
</tr>
<tr>
<td>Master salt length</td>
<td>112 bits</td>
</tr>
<tr>
<td>Key Derivation Function</td>
<td>AES_256_CM_PRF (Section 3)</td>
</tr>
<tr>
<td>Default key lifetime</td>
<td>2^31 packets</td>
</tr>
<tr>
<td>Cipher (for SRTP and SRTCP)</td>
<td>AES_256_CM (Section 2)</td>
</tr>
<tr>
<td>SRTP authentication function</td>
<td>HMAC-SHA1 (Section 4.2.1 of [RFC3711])</td>
</tr>
<tr>
<td>SRTP authentication key length</td>
<td>160 bits</td>
</tr>
<tr>
<td>SRTP authentication tag length</td>
<td>32 bits</td>
</tr>
<tr>
<td>SRTCP authentication function</td>
<td>HMAC-SHA1 (Section 4.2.1 of [RFC3711])</td>
</tr>
<tr>
<td>SRTCP authentication key length</td>
<td>160 bits</td>
</tr>
<tr>
<td>SRTCP authentication tag length</td>
<td>80 bits</td>
</tr>
</tbody>
</table>

### Table 4: The AES_256_CM_HMAC_SHA1_32 cryptosuite.
8.0 CSEC Schema ("csec.xsd")

The schema file can be found in the PSIA schema repository at:
http://www.psialliance.org/schemas/csec/1.0/csec.xsd

9.0 CSEC Permission Dictionary Schema
("csecPermissionDictionary.xsd")

NOTE: This has no “targetNamespace” for anonymous insertion into CSEC schema. Other possible (future) vendor-specific Dictionaries can be integrated in same fashion with less direct impact on the CSEC schema and “psialliance-org” namespace.

The schema file can be found in the PSIA schema repository at:
http://www.psialliance.org/schemas/csec/1.0/csecPermissionDictionary.xsd

10.0 PSIA Common Types Schema ("psiaCommonTypes.xsd")

The schema file can be found in the PSIA schema repository at:
http://www.psialliance.org/schemas/system/1.2/psiaCommonTypes.xsd

11.0 ACWG Common Types Schema ("ACWGCommonTypes.xsd")

The schema file can be found in the PSIA schema repository at:
http://www.psialliance.org/schemas/system/1.2/ACWGCommonTypes.xsd

---

1 Alternatives: a) Put all Dictionaries into their own individual namespaces, requiring use of explicit “xs:import” in the CSEC schema to change Dictionaries. b) Put all Dictionaries into “psialliance-org” namespace, which would require name-collision avoidance by schema authors and explicit change to CSEC schema to switch Dictionaries. In most scenarios, only one Dictionary can be used easily. Multiple Dictionaries would require changing “SymbolicPermissionDescriptor” definition.