Physical Security Interoperability Alliance
PSIA Specification: Common Security Service Specification
Version 2.0 R1
February 26, 2015

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

System WG

Revision History	Description	Date	By
Version 1.0 Rev 1	Initial Draft	July 9, 2010	James Wang / Roger Richter
Version 1.0 Rev 2	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-Requirements Table (Section 4) 4) Add supportedPermissions Resource to advertise Permission Dictionary 5) Updated Resource Hierarchy diagram to reflect new Resource (supportedPermissions). 6) general cleanup	August 20, 2010	James Wang
Version 1.0 Rev 3	1) Add <dictionaryName> to csecPermissionDictionary.xsd 2) Add <permissionBitOffset> to csecPermissionDictionary.xsd 3) CSEC.XSD: a) add <userDescription> string, b) add <fullName>, c) add <UserCODES>, d) add <subDeviceAuthorizationLevel>, e) Changed all REST <id>'s to comply with "psiaCommonTypes.xsd", using "LocalID" type and 1-based indices vs 0-based indices.	September 21, 2010	James Wang
Version 1.0 Rev 3b	Minor cleanup of text and add more explanation of Users/Groups.	November 11, 2010	James Wang
Version 1.0 Rev 4	Add pending features + others suggested by System, Access, & Intrusion WG's: 1) Add "Owner GUID" QS required to take ownership of device 2) Add "deviceOwnership/scope" resource (scoping of "ownership") 3) Add "deviceOwnership/status" resource 4) Add simple MD5 Digest for Ownership MAC 5) Add "loginPortAffinity" to "PermissionDescriptor" to allow Permissions to only be enabled/allowed if the user request came in on a specific port 6) Add text requiring "admin" Group and User 7) Add text requiring "viewonly" Group and User 8) Add ability to enable/disable Users and Groups 9) Replace <UserCODES> with <UserCODEList> 10) updated REST Resource Hierarchy Diagram	February 11, 2011	James Wang
Version 1.0 Rev 4b	Feedback from SWG: 1) Modify 'deviceOwnership' Resource description to clarify User-ID locked Cookie MAC and uses for Group Sharing 2) Add ownership MAC diagram 3) Add /System/logging as IPMD dependent Resource to Requirements tables	May 27, 2011	James Wang
Version 1.0 Rev 4c	Feedback from SWG: 1) Fix reference links	June 7, 2011	James Wang
Version 1.0 Rev 4d	Feedback From SWG: 1) Add new symbolic permission "/ResetDeviceOwnership" 2) Add QS to REST Resource "/PSIA/CSEC/deviceOwnership" to perform new Reset operation 3) Schema: Fix typo in ="MIKEYExchangeMethod"	July 19, 2011	James Wang
Version 1.0 Rev 4e	Feedback from ACWG: 1) Add "ACWGPermissionInfoList" to <CSECPermissionGroup> to achieve encapsulation of ACWG Permissions into CSEC 2) Remove ACWG related elements from <DeviceScopeRestriction>. 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	August 17, 2011	James Wang
Version 1.0 Rev 4f	1) Add "UID" to TimeScheduleInfo and HolidayInfo	August 16, 2011	James Wang
Version 1.0 Rev 4g	1) Removed schema text and replace with link to psia schema repository	September 1, 2011	James Wang
Version 1.0 Rev 4h	Feedback from ACWG + other improvements: 1) Make KeyManager optional (Sect 4.1). 2) Allow for multiple device certificates 3) Change CA Service to allow management of "root cert package"	April 30, 2012	James Wang
	Version 1.0 TBD: 1) Add More descriptive text throughout 2) Describe example VMS – CSEC Device flows 3) Add More MAC-algorithms for deviceOwnership		

	4) Add Table mapping symbolic permissions to PSIA REST Resources		
Version 1.1 Rev. 01	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.	August 12, 2012	Roger RIchter
Version 1.1, Rev. 03		October 31, 2012	Roger Richter
Version 1.1, Revision 04	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.	November 6, 2012	Roge r RIchter
Version 1.1, Revision 0.4a	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.	January 17, 2013	Roger RIchter
Version 1.2, Revision 0.1	Obsoleted MD5 as and 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.	April 5, 2014	Roger Richter
Version 1.2, R0.2	In Section 7.2 added support for RFC 6265 usage of "Max-Age" expiry designation, in addition to (the already present) "Expires" header designation. Added the new "PLAI" profie requirements in Section 4.	April 22, 2014	Roger Richter
Version 2.0, R 0	Updates per Topologies, general cleanup	Feb 18, 2015	Jeffrey Longo
Version 2.0 R 1	Updates per group meeting 2/26/15	Feb 26, 2015	Jeffrey Longo

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.

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

Table of Contents

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 sophisticate 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.

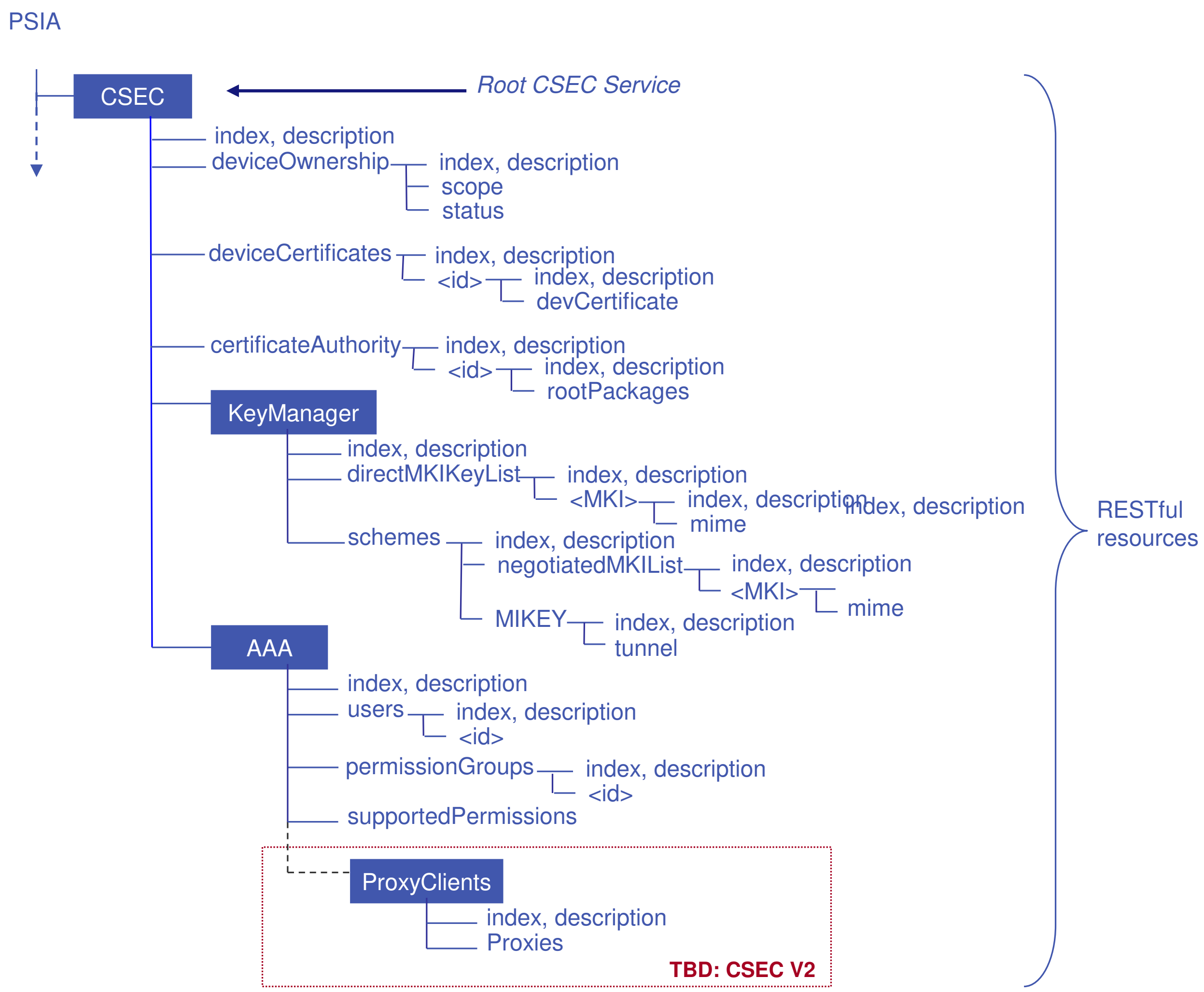
2.0 References

1	Security Requirement document	http://www.psiaforums.org/attachment.php?attachmentid=160&d=1306961788
2	Security Use Case Diagrams	http://www.psiaforums.org/attachment.php?attachmentid=161&d=1306962059 http://www.psiaforums.org/attachment.php?attachmentid=162&d=1306962078
3	[RFC 1945]	"Hypertext Transfer Protocol -- HTTP/1.0", T. Berners-Lee et al, May 1996 http://www.ietf.org/rfc/rfc1945.txt
4	[RFC 2326]	"Real Time Streaming Protocol (RTSP)", H. Schulzrinne et al, April 1998 http://www.ietf.org/rfc/rfc2326
5	[RFC 2327]	"SDP", M. Handley et al, April 1998 http://www.ietf.org/rfc/rfc2327
6	[RFC 2616]	"Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding et al, June 1999 http://www.ietf.org/rfc/rfc2616.txt
7	[RFC 2965]	"HTTP State Management Mechanism", D. Kristol et al, October 2000 http://www.ietf.org/rfc/rfc2965.txt
8	[RFC 3164]	"The BSD Syslog Protocol", C. Lonvick, August 2001 http://www.ietf.org/rfc/rfc3164.txt
9	[RFC 3550]	"RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne et al., July 2003 http://www.ietf.org/rfc/rfc3550.txt
10	[RFC 3711]	"The Secure Real-time Transport Protocol (SRTP)", M. Baugher et al, March 2004 http://www.ietf.org/rfc/rfc3711.txt
11	[RFC 3830]	"MIKEY", J. Arkko et al, August 2004 http://www.ietf.org/rfc/rfc3830.txt
12	[RFC 4571]	"Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", J. Lazzaro, July 2006 http://www.ietf.org/rfc/rfc4571.txt
13	[RFC 4567]	"Key Management Extensions for SDP, and RTSP", J. Arrko et al, July 2006 http://www.ietf.org/rfc/rfc4567.txt
14	[Errata for RFC 4567]	Errata 2247 for RFC 4567 (fixes missing "data" field in "key-mgmt-spec" for new RTSP header) http://www.rfc-editor.org/errata_search.php?rfc=4567
15	[RFC 4568]	"SDP Security Description for Media Streams", F. Andreasen et al., July 2006 http://www.ietf.org/rfc/rfc4568.txt
16	["srtp-big-aes-3"]	"The use of AES-192 and AES-256 in SRTP draft", D. McGrew, March 2010 http://tools.ietf.org/html/draft-ietf-avt-srtp-big-aes-03
17	[RFC 4650]	"HMAC-Authentication Diffie-Hellman for MIKEY", M. Euchner, September 2006 http://www.ietf.org/rfc/rfc4650.txt
18	[Errata for RFC4650]	http://www.rfc-editor.org/errata_search.php?rfc=4650&rec_status=15&presentation=records
19	[RFC 4738]	"MIKEY-RSA-R", D. Ignatic et al., November 2006 http://www.ietf.org/rfc/rfc4738.txt
20	[RFC 5285]	"A General Mechanism for RTP Header Extensions", D. Singer et al, July 2008 http://www.ietf.org/rfc/rfc5285.txt
21	"Key Management Mechanisms"	http://media.techtarget.com/searchVoIP/downloads/Key.Management.Mechanisms.CH7.pdf
22	"REST Anti-Patterns"	http://www.infoq.com/articles/rest-anti-patterns, July 2008
23	PSIA Service Model specification	http://www.psialliance.org/documents/PSI-Service-Model_version_1_0.pdf, March 17, 2009

24	PSIA IP Media Device specification	http://www.psialliance.org/register_form.html?file=SpecPackQ109, February 15, 2010

3.0 REST Resource Hierarchy

CSEC v1r4 Device Security Service Structure



4.0 REST Resource Requirements

4.1 /PSIA/CSEC (Root) Services & Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
✓	✓	*		deviceOwnership (Required for PLAI, applies to data ownership)	✓			✓
				deviceCertificate	✓	✓		
				certificateAuthority	✓	✓	✓	
				KeyManager				
✓	✓			AAA				

4.2 /PSIA/CSEC/deviceOwnership Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
				Scope	✓	✓		
				Status	✓			

4.3 /PSIA/CSEC/certificateAuthority Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				indexr	✓			
				description	✓			
				<id>	✓	✓		✓
				<id>/index	✓			
				<id>/description	✓			
				<id>/caCertificate	✓	✓		

4.4 /PSIA/CSEC/KeyManager Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
				directMKIKeyList	✓	✓	✓	
				Schemes				

4.4.1 /PSIA/CSEC/KeyManager/directMKIKeyList Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
				<MKI>	✓	✓		✓
				<MKI>/index	✓			
				<MKI>/indexr	✓			
				<MKI>/description	✓			
				<MKI>/mime	✓			

4.4.2 /PSIA/CSEC/KeyManager/schemes Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
				NegotiatedMKIList	✓	✓	✓	
				MIKEY				

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

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
				<MKI>	✓	✓		✓
				<MKI>/index	✓			
				<MKI>/indexr	✓			
				<MKI>/description	✓			
				<MKI>/mime	✓			

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

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				tunnel	✓		✓	

4.5 /PSIA/CSEC/AAA Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
✓	✓			Users	✓	✓	✓	
RO	RO			permissionGroups	✓	✓	✓	
✓	✓			supportedPermissions	✓			

4.5.1 /PSIA/CSEC/AAA/users Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POST	DEL
				index	✓			
				indexr	✓			
				description	✓			
✓	✓			<id>	✓	✓		✓

4.5.2 /PSIA/CSEC/AAA/permissionGroups Resources

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POS T	DEL
				index	✓			
				indexr	✓			
				description	✓			
RO	**RO**			<id>	✓	✓		✓

4.6 /PSIA/System Resources (Service Model v2.0+)

M-S Internet	M-S LAN	P-P Internet	P-P LAN	Command	GET	PUT	POS T	DEL
✓	✓			logging	✓	✓		

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.”

[MSG]

“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

Facility	Severity	PRI	Example Log Entry
10	0 - Emergency	80	<80> Aug 3 22:15:20 my.machine.org aaaproc: emergency security breach
10	1 – Alert	81	<81> Aug 3 20:11:15 my.machine.org aaaproc: intrusion detected
10	2 - Critical	82	<82> Aug 3 10:10:15 my.machine.org aaaproc: too many login failures
10	3 – Error	83	<83> Aug 2 22:15:30 my.machine.org aaaproc: database partially corrupted
10	4 - Warning	84	<84> Aug 2 22:10:15 my.machine.org aaaproc: database nearly full
10	5 - Notice	85	<85> Aug 2 21:05:08 my.machine.org aaaproc: unauthorized operation
10	6 - Informational	86	<86> Aug 2 20:10:15 my.machine.org aaaproc: routine backups started
10	7 – Debug	87	<87> Aug 1 09:00:00 my.machine.org aaaproc: debug test code 357

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.

Category	Description	M-S Internet	P-P Internet	M-S LAN	P-P LAN
HTTP Authentication	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.			*	*
HTTPS, and Authentication	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.	✓	✓	*	*

*= HTTP or HTTPS is required in a LAN topology

Resource Details

6.1 /PSIA/CSEC

URI	/PSIA/CSEC		**Type**	Service
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
Notes	Common Security (CSEC) Service root.			

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"].

URI	/PSIA/CSEC/deviceOwnership	**Type**	Resource

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).		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	OwnerGUID, ExpireTime *OR* MaxAge	None	<CSECOwnershipCookie> or "HTTP 409 Conflict"
PUT	N/A	N/A	<ResponseStatus w/error code>
POST	N/A	N/A	<ResponseStatus w/error code>
DELETE	ResetOwnership	<CSECOwnershipCookie> or None	<ResponseStatus>
Notes			

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 <CSECOwnershipCookie>, in HTTP payload, which contains information used for all subsequent administrative requests.

The <CSECOwnershipCookie> 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 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 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).

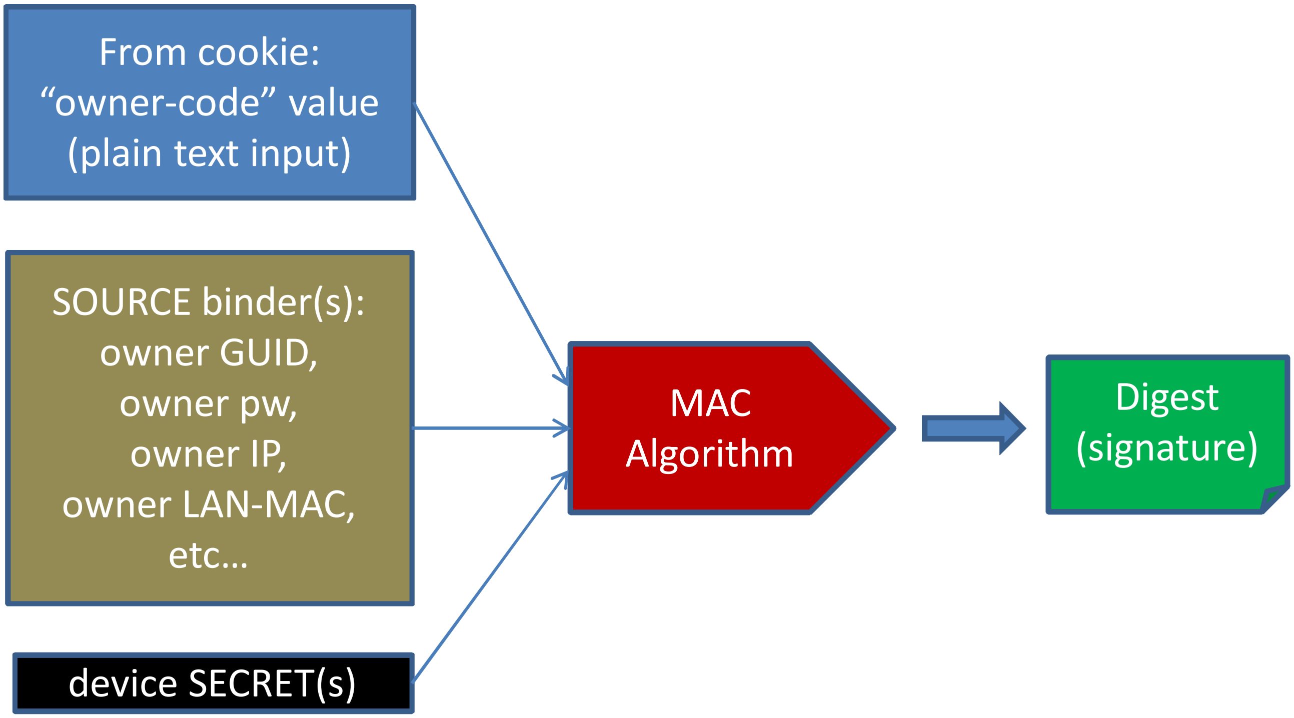


Figure 1: MAC Algorithm Function

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).

issuer-signature = BASE16 string representation of calculation: MD5(**owner-code:OwnerGUID**: **user-id:user-pw**:**secret**).

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

URI	`/PSIA/CSEC/deviceOwnership/scope`		**Type**	Resource
Function	Resource used to manage the "ownership" scope definition			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET	N/A	None	<CSECOwnershipScope>	
PUT	N/A	<CSECOwnershipScope>	<ResponseStatus>	
POST	N/A	N/A	<ResponseStatus w/error code>	
DELETE	N/A	N/A	<ResponseStatus w/error code>	
Notes				

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

URI	`/PSIA/CSEC/deviceOwnership/status`		**Type**	Resource
Function	Resource used get the current "ownership" status			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET	N/A	None	<CSECOwnershipStatus>	
PUT	N/A	N/A	<ResponseStatus w/error code>	
POST	N/A	N/A	<ResponseStatus w/error code>	
DELETE	N/A	N/A	<ResponseStatus w/error code>	
Notes				

6.3 /PSIA/CSEC/deviceCertificates

URI	/PSIA/CSEC/deviceCertificates	Type	Resource
Function	Resource used to manage the <DeviceCertificateList>.		
Methods	Query String(s)	Inbound Data	Return Result
GET	None	None	<DeviceCertificateList>
PUT	None	<DeviceCertificateList>	<ResponseStatus>
POST	None	<DeviceCertificateList>	<ResponseStatus>
DELETE	N/A	N/A	<ResponseStatus w/error code>
Notes			

Example 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-----
                MIIDBjCCAe4CCQCX05m0b053QzANBgkqhkiG9w0BAQQFADBFMQswCQYDVQQGEwJB
                VTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50ZXJuZXQgV2lkZ2l0
                cyBQdHkgTHRkMB4XDTA4MTIwNzEwMjUyMloXDTE4MTIwNTEwMjUyMlowRTELMAkG
                A1UEBhMCQVUxEzARBgNVBAgTClNvbWUtU3RhdGUxITAfBgNVBAoTGEludGVybmV0
                IFdpZGdpdHMgUHR5IEx0ZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
                AMDjWizj+xHXoKo0bkCkg187x5hTGXwbn44RC4TY6OSrC+Inh4TEhgnElest9oc4
                akeZb01YmkHs/sYXOfDWUbvjlIBIaHUQVv7wDo6c/8XTM+R2ghIT9xPVtaZIrVbH
                kvAea64CaNQXwhmotXz2r9Z7rT6tio+7zMtuPoOOtC6J3+pZ9XYuGkyEbQKl7VFO
                kRaTyT9+T5Al20yuuTvByUscbf17X6HsxwlOflCMJmRmMeOlAVs4NyHyumxj0oh1
                N6XYk6JdLjD7fxlCvseDcmSePyJrDQInEcDAXY2uMsBylyXoR4FwbYbFSB2y9tcf
                uIm7IPXlTP14qE1erVtyK3cCAwEAATANBgkqhkiG9w0BAQQFAAOCAQEAW4yZdqpB
                oIdiuXRosr86Sg9FiMg/cn+2OwQ0QIaA8ZBwKsc+wIIHEgXCS8J6316BGQeUvMD+
                plNe0r4GWzzmlDMdobeQ5arPRB89qd9skE6pAMdLg3FyyfEjz3A0VpskolW5VBMr
                P5R7uJ1FLgH12RyAjZCWYcCRqEMOffqvyMCH6oAjyDmQOA5IssRKX/HsHntSH/HW
                W7slTcP45ty1b44Nq22/ubYk0CJRQgqKOIQ3cLgPomN1jNFQbAbfVTaK1DpEysrQ
                5V8a8gNW+3sVZmV6d1Mj3pN2Le62wUKuV2g6BNU7iiwcoY8HI68aRxz2hVMS+t5f
                SEGI4JSxV56lYg==
                -----END CERTIFICATE-----
            </CertificateText>
        </CertificateDesc>
    </DeviceCertificate>
    <DeviceCertificate version="1.0">
        <id>2</id>
        <loginPortAffinity>eth 000</loginPortAffinity>
        <tcpPort>8554</tcpPort>
        <application>RTSPS</application>
        <CertificateDesc version="1.0">
            <CertificateFormat>PEM</CertificateFormat>
            <CertificateText>
                -----BEGIN CERTIFICATE-----
                MIIDdTCCAt6gAwIBAgIJAOlmKWNOCMEGMA0GCSqGSIb3DQEBBQUAMIGEMQswCQYD
                VQQGEwJVUzELMAkGA1UECBMCVFgxDzANBgNVBAcTBkF1c3RpbjEMMAoGA1UEChMD
                R0VTMREwDwYDVQQLEwhTZWN1cml0eTETMBEGA1UEAxMKSmFtZXMgV2FuZzEhMB8G
                CSqGSIb3DQEJARYSamFtZXMud2FuZzJAZ2UuY29tMB4XDTA5MDkyMjE3MTAwOVoX
                DTEwMDkyMjE3MTAwOVowgYQxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJUWDEPMA0G
                A1UEBxMGQXVzdGluMQwwCgYDVQQKEwNHRVMxETAPBgNVBAsTCFNlY3VyaXR5MRMw
                EQYDVQQDEwpKYW1lcyBXYW5nMSEwHwYJKoZIhvcNAQkBFhJqYW1lcy53YW5nMkBn
                ZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANYJqH/N44H3L+gp8/Qq
                PKFPngsYtZEk4CCL50lEMVWuupZFbrmnY/8aO9m8gGS+2RtP9YgCkRM1qUIoUHli
                3wd7K+UkbDm1U8wuP5ET6c41xxMMzM+fUQzZIz9HSFdvAsOd2UyMkkanRJQcHBD5
                Zq04c80f+lWr84S3MVTT07/VAgMBAAGjgewwgekwHQYDVR0OBBYEFEME1Mh91x3p
                RufxtXnuGgvSCGISMIG5BgNVHSMEgbEwga6AFEME1Mh91x3pRufxtXnuGgvSCGIS
                oYGKpIGHMIGEMQswCQYDVQQGEwJVUzELMAkGA1UECBMCVFgxDzANBgNVBAcTBkF1
                c3RpbjEMMAoGA1UEChMDR0VTMREwDwYDVQQLEwhTZWN1cml0eTETMBEGA1UEAxMK
                SmFtZXMgV2FuZzEhMB8GCSqGSIb3DQEJARYSamFtZXMud2FuZzJAZ2UuY29tggkA
                6WYpY04IwQYwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOBgQAXc7PEkRvY
                GZjisdH0GwN5XP0jcds+pXQ0k0mxlEPx0tFdINxZpI94CP24HigpbhWe09w66hWO
                pg3X6tk1v5mybxTx1qtAg+h3QTa+uUVoJAjeDYtgL0lWFRev/pFHhXQlapyaRemj
                OhJLInS+02YzpGd8PuhQR3D1wAjkFCplAw==
                -----END CERTIFICATE-----
            </CertificateText>
        </CertificateDesc>
    </DeviceCertificate>
    <DeviceCertificate version="1.0">
        <id>3</id>
        <loginPortAffinity>eth 001</loginPortAffinity>
        <tcpPort>989</tcpPort>
        <tcpPort>990</tcpPort>
        <application>FTPS</application>
        <CertificateDesc version="1.0">
            <CertificateFormat>B64PKCS12</CertificateFormat>
            <CertificateText>

MIIF8gIBAzCABgkqhkiG9w0BBwGggASCBakwggWlMIAGCSqGSIb3DQEHBqCAMIIDlAIBADCCA40GCSqGSIb3DQEHATAcBgoqhkiG
9w0BDAEGMA4ECNN4m59QAXIbAgIIAICCA2D6tsqcc6KwI07M6Yf6r7DVKaRY3WbQAMcMPj8U2r5p3HklI16MIgVTHizOc36i6jKi
2qC7Ee2rdJG7/TcDs0ukhSk9P6jjVt6U0IANKiMEADKX+lkMwyTpc7NKMikmUf1Nkn6UIIliEObmMSHcOFr/qQ4sKsUZ2nGb1Bgo
Cx7PQoE/Zc73NpEkyd1LWch+VJxkYOX5B022BG4pXVZa1EF/QZ+XZ4AIkbQfBqv2lR0HpQzWIrGkjzBA7nplX5eS0IXvQSd1ksDz
7z5cufl4TlULpr9CsKMB/yVg7rNfUpE/n3mvxJmAI0bosbgIU2Oi7qfHE9ORgBcdshIKPa0kHbQNKEmoK7yBR6nKsACxySUdyRMX
JaUCXhL8xrLD9QgzUH0708CCKU+8Bjfbbbi0dlVde4P36ECpHMJIuKmvshfyq0JsHvfNf9IiYPEuS2ku8bcFrGyHbQ6Trc+/l03A
qlAqRmLYs2fz5WhRr00aorgLR5AB9LJUqFkZI1Okzr83oc2/ttgVtTqqSA8GvQPkTkWmzHEE0I9B5M73ursIfmijGnaXn8wlu/VC
WWM2v5uiniG7Y3ROJdgV4Fm2wekBZINUdSRZ+r/NSIJU/Jns5PQMLh8KfAdCuPmvxuH9uSlklaBQAG77FJI+R+HneUa5W700G1kZ
UIFo3a3YbmG3NLPlNVaSi2d/zH0+S/UxiSJYMNf5iPNwlsco+2+KFWNMcJOw473N6z+u8RmCOaKUruNb9aKWE6YtkqXHZlY9JTgT
Vj8/WyL2ASRTbRdS5E5LyLi3pYHijHoBWotNiG43XnUSPYZ/cLGNk5TcWEV8VOHkD93hChgya9vubNJst/MhDSlFfwLgwHwYy7HZ
nps/OIUUywuWAHcI3bmYNixpNqrfxbXhMub5Axo5C0hV2KMpzbBfxEfJbTl1JC1xIfxcbmeJ/Dn3f4t6JX/HOay9vfgDpOVKtqGn
aE7p8SzYrroBNw4b39mZTdKjyrOz9nIOhFHPQKtNmMzuG/le7jzG7BMrJvPVvwvvcyiUOg57cephPJzEk2BenpHlawIdhBFMBXxA
xIr3w+axADyL9da7S5jMlORnRTjewICjSDT/qhBjAgixkkv4Yulynw5Kl2n3pbNy9cpXQeKW/AElGQsoUWArlA0AAAAAMIAGCSqG
SIb3DQEHAaCABIIB4zCCAd8wggHbBgsqhkiG9w0BDAoBAqCCAXYwggFyMBwGCiqGSIb3DQEMAQMwDgQIS1LBKtbOn2gCAggABIIB
UDJimTj0KCK1h0aFTm+yKkpG/FLl1M8NpGZLWgzEpUwVN5xGdL4vMe+KRHgruRwwSWf6udDGLSFUQzFZD4dlhRqXafOmqtYiB2z0
yS/pxx03A8/piEMli6WRFPG9Nn9x7XgVFoU67y+g8W8qi0R7X4sMZSyr+tYoJIkwiJmg7xFakuoDue0q6e9jQOcVa8uBOG/hgEEM
IYI6SJYsg91oh1K5n3eFK8xHmPuUPR/MtaYofqzdq+KDiYVLLEYrKL6GH6/aI+3vRxSzihp/PKj3WhkK0tK8MIIJx9KpUXhJbLZX
uJBLlRurnHAYzH0pGzaJX2LIRmgsn62agVZcbHHc5bVjrFMg6HQLqZ9zyRaoSYWEwhrOzUbwi0YIjy81wU5iXX7+PAZXRsBmatCx
8Xf6BXDvLzFj1VDqjfeDbWlBQqocerbw6T/FScph9zLisjjJATFSMCMGCSqGSIb3DQEJFTEWBBQcnAA3C0oHBpz6JM8ySx/wrroO
```

```
mjArBgkqhkiG9w0BCRQxHh4cAE0AeQAgAEMAZQByAHQAaQBmAGkAYwBhAHQAZQAAAAAAAAAAAMC0wITAJBgUrDgMCGgUABBR8vQfw
nHcyj9kUZbJI7b2YRsUY5AQIKU86swHdipI=
            </CertificateText>
        </CertificateDesc>
    </DeviceCertificate>
</DeviceCertificateList>
```

6.3.1 /PSIA/CSEC/deviceCertificates/<id>

URI	`/PSIA/CSEC/deviceCertificates/<id>`		**Type**	Resource
Function	Resource used to manage a particular <DeviceCertificate>.			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET	None	None	<DeviceCertificate>	
PUT	None	< DeviceCertificate >	<ResponseStatus>	
POST	N/A	N/A	<ResponseStatus w/error code >	
DELETE	None	None	<ResponseStatus>	
Notes	See previous section for XML example.			

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

URI	`/PSIA/CSEC/deviceCertificates/<id>/devCertificate`		**Type**	Resource
Function	This Resource is used to load/update the Device's SSL/TLS Server certificate in a direct or "raw" fashion.			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET	None	None	X.509v3/PEM (returns public/cert only)	
PUT	None	X.509v3/PEM (public/cert & private keys)	<ResponseStatus>	
POST	N/A	N/A	<ResponseStatus w/error code>	
DELETE	N/A	N/A	<ResponseStatus w/error code>	
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-x509-ca-cert" (DER) "Content-Type: application/x-x509-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):

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAwONaLOP7EdegqjRuQKSDXzvHmFMZfBufjhELhNjo5KsL4ieH
hMSGCcSV6y32hzhqR5lvTViaQez+xhc58NZRu+OUgEhodRBW/vAOjpz/xdMz5HaC
EhP3E9W1pkitVseS8B5rrgJo1BfCGai1fPav1nutPq2Kj7vMy24+g460Lonf6ln1
di4aTIRtAqXtUU6RFpPJP35PkCXbTK65O8HJSxxt/XtfoezHCU5+UIwmZGYx46UB
Wzg3IfK6bGPSiHU3pdiTol0uMPt/GUK+x4NyZJ4/ImsNAicRwMBdja4ywHKXJehH
gXBthsVIHbL21x+4ibsg9eVM/XioTV6tW3IrdwIDAQABAoIBACFfdLutmkQFBcRN
HAJNNHmmsyr0vcUOVnXTFyYeDXV67qxrYHQlOHe6LqIpKq1Mon7O2kYMnWvooFAP
trOnsS6L+qaTYJdYg2TKjgo4ubw1hZXytyB/mdExuaMSkgMgtpia+tB5lD+V+LxN
x1DesZ+veFMO3Zluyckswt4qM5yVa04YFrt31H0E1rJfIen61lidXIKYmHHWuRxK
SadjFfbcqJ6P9ZF22BOkleg5Fm5NaxJmyQynOWaAkSZa5w1XySFfRjRfsbDr64G6
+LSG8YtRuvfxnvUNhynVPHcpE40eiPo6v8Ho6yZKXpV5klCKciodXAORsswSoGJa
N3nnu/ECgYEA6Yb2rM3QUEPIALdL8f/OzZ1GBSdiQB2WSAxzl9pR/dLF2H+0pitS
to0830mk92ppVmRVD3JGxYDRZQ56tlFXyGaCzJBMRIcsotAhBoNbjV0i9n5bLJYf
BmjU9yvWcgsTt0tr3B0FrtYyp2tCvwHqlxvFpFdUCj2oRw2uGpkhmNkCgYEA03M6
WxFhsix3y6eVCVvShfbLBSOqp8l0qiTEty+dgVQcWN4CO/5eyaZXKxlCG9KMmKxy
Yx+YgxZrDhfaZ0cxhHGPRKEAxM3IKwT2C8/wCaSiLWXZZpTifnSD99vtOt4wEfrG
+AghNd5kamFiM9tU0AyvhJc2vdJFuXrfeC7ntM8CgYBGDA+t4cZcbRhu7ow/OKYF
kulP3nJgHP/Y+LMrl3cEldZ2jEfZmCElVNQvfd2XwTl7injhOzvzPiKRF3jDez7D
g8w0JAxceddvttJRK9GoY4l7OoeKpjUELSnEQkf+yUfOsTbXPXVY7jMfeNL6jE6b
qN7t3qv8rmXtejMBE3G6cQKBgGR5W2BMiRSlxqKx1cKlrApV87BUe1HRCyuR3xuA
d6Item7Lx1oEi7vb242yKdSYnpApWQ06xTh83Y/Ly87JaIEbiM0+h+P8OEIg0F1a
iB+86AcUX1I8KseVy+Np0HbpfwP8GrFfA5DaRPK7pXMopEtby8cAJ1XZZaI1/ZvZ
BebHAoGAcQU9WvCkT+nIp9FpXfBybYUsvgkaizMIqp66/l3GYgYAq8p1VLGvN4v5
ec0dW58SJrCpqsM3NP78DtEzQf9OOsk+FsjBFzDU2RkeUreyt2/nQBj/2mN/+hEy
hYN0Zii2yTb63jGxKY6gH1R/r9dL8kXaJmcZrfSa3AgywnteJWg=
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIDBjCCAe4CCQCX05m0b053QzANBgkqhkiG9w0BAQQFADBFMQswCQYDVQQGEwJB
VTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50ZXJuZXQgV2lkZ2l0
cyBQdHkgTHRkMB4XDTA4MTIwNzEwMjUyMloXDTE4MTIwNTEwMjUyMlowRTELMAkG
A1UEBhMCQVUxEzARBgNVBAgTClNvbWUtU3RhdGUxITAfBgNVBAoTGEludGVybmV0
IFdpZGdpdHMgUHR5IEx0ZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMDjWizj+xHXoKo0bkCkg187x5hTGXwbn44RC4TY6OSrC+Inh4TEhgnElest9oc4
akeZb01YmkHs/sYXOfDWUbvjlIBIaHUQVv7wDo6c/8XTM+R2ghIT9xPVtaZIrVbH
kvAea64CaNQXwhmotXz2r9Z7rT6tio+7zMtuPoOOtC6J3+pZ9XYuGkyEbQKl7VFO
kRaTyT9+T5Al20yuuTvByUscbf17X6HsxwlOflCMJmRmMeOlAVs4NyHyumxj0oh1
N6XYk6JdLjD7fxlCvseDcmSePyJrDQInEcDAXY2uMsBylyXoR4FwbYbFSB2y9tcf
uIm7IPXlTP14qE1erVtyK3cCAwEAATANBgkqhkiG9w0BAQQFAAOCAQEAW4yZdqpB
```

```
oIdiuXRosr86Sg9FiMg/cn+2OwQ0QIaA8ZBwKsc+wIIHEgXCS8J6316BGQeUvMD+
plNe0r4GWzzmlDMdobeQ5arPRB89qd9skE6pAMdLg3FyyfEjz3A0VpskolW5VBMr
P5R7uJ1FLgH12RyAjZCWYcCRqEMOffqvyMCH6oAjyDmQOA5IssRKX/HsHntSH/HW
W7slTcP45ty1b44Nq22/ubYk0CJRQgqKOIQ3cLgPomN1jNFQbAbfVTaK1DpEysrQ
5V8a8gNW+3sVZmV6d1Mj3pN2Le62wUKuV2g6BNU7iiwcoY8HI68aRxz2hVMS+t5f
SEGI4JSxV56lYg==
-----END CERTIFICATE-----
-----BEGIN DH PARAMETERS-----
MEYCQQD+ef8hZ4XbdoyIpJyCTF2UrUEfX6mYDvxuS5O1UNYcslUqlj6JkA11e/yS
6DK8Z86W6mSj5CEk4IjbyEOECXH7AgEC
-----END DH PARAMETERS-----
```

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 likely to be 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

URI	/PSIA/CSEC/certificateAuthority	**Type**	Resource
Function	Resource used to manage the <CertificateAuthorityList>.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<CertificateAuthorityList>
PUT	None	<CertificateAuthorityList>	<ResponseStatus>
POST	None	<CertificateAuthority>	<ResponseStatus>
DELETE	N/A	N/A	<ResponseStatus w/error code>
Notes			

Example 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
VeriSign</RDNValue>
            </RDN>
            <RDN>
                <RDNLongLabel>CommonName</RDNLongLabel>
                <RDNValue>VeriSign</RDNValue>
            </RDN>
        </DistinguishedName>
        <crlDistributionPoint>http://crl.verisign.com/pca1.crl</crlDistributionPoint>
        <crlDistributionPoint>http://crl.verisign.com/pca2.crl</crlDistributionPoint>
        <ocspInfoAccess>http://ocsp.verisign.com</ocspInfoAccess>
        <RootPackages version="1.0">
            <RootCertPackage version="1.0">
                <id>1</id>
                <path>Universal</path>
                <CertificateDesc version="1.0">
                <CertificateFormat>PEM</CertificateFormat>
                <CertificateText>
                -----BEGIN CERTIFICATE-----
                MIIEuTCCA6GgAwIBAgIQQBrEZCGzEyEDDrvkEhrFHTANBgkqhkiG9w0BAQsFADCB
                vTELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
                ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTowOAYDVQQLEzEoYykgMjAwOCBWZXJp
                U2lnbiwgSW5jLiAtIEZvciBhdXRob3JpemVkIHVzZSBvbmx5MTgwNgYDVQQDEy9W
                ZXJpU2lnbiBVbml2ZXJzYWwgUm9vdCBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAe
                Fw0wODA0MDIwMDAwMDBaFw0zNzEyMDEyMzU5NTlaMIG9MQswCQYDVQQGEwJVUzEX
                MBUGA1UEChMOVmVyaVNpZ24sIEluYy4xHzAdBgNVBAsTFlZlcmlTaWduIFRydXN0
                IE5ldHdvcmsxOjA4BgNVBAsTMShjKSAyMDA4IFZlcmlTaWduLCBJbmMuIC0gRm9y
                IGF1dGhvcml6ZWQgdXNlIG9ubHkxODA2BgNVBAMTL1ZlcmlTaWduIFVuaXZlcnNh
                bCBSb290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEF
                AAOCAQ8AMIIBCgKCAQEAx2E3XrEBNNti1xWb/1hajCMj1mCOkdeQmIN65lgZOIzF
                9uVkhbSicfvtvbnazU0AtMgtc6XHaXGVHzk8skQHnOgO+k1KxCHfKWGPMiJhgsWH
                H26MfF8WIFFE0XBPV+rjHOPMee5Y2A7Cs0WTwCznmhcrewA3ekEzeOEz4vMQGn+H
                LL729fdC4uW/h2KJXwBL38Xd5HVEMkE6HnFuacsLdUYI0crSK5XQz/u5QGtkjFdN
                /BMReYTtXlT2NJ8IAfMQJQYXStrxHXpma5hgZqTZ79IugvHw7wnqRMkVauIDbjPT
                rJ9VAMf2CGqUuV/c4DPxhGD5WycRtPwW8rtWaoAljQIDAQABo4GyMIGvMA8GA1Ud
                EwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMG0GCCsGAQUFBwEMBGEwX6FdoFsw
                WTBXMFUWCWltYWdlL2dpZjAhMB8wBwYFKw4DAhoEFI/l0xqGrI2Oa8PPgGrUSBgs
                exkuMCUWI2h0dHA6Ly9sb2dvLnZlcmlzaWduLmNvbS92c2xvZ28uZ2lmMB0GA1Ud
                DgQWBBS2d/ppSEefUxLVwuoHMnYH0ZcHGTANBgkqhkiG9w0BAQsFAAOCAQEASvj4
                sAPmLGd75JR3Y8xuTPl9Dg3cyLk1uXBPY/ok+myDjEedO2Pzmvl2MpWRsXe8rJq+
                seQxIcaBlVZaDrHC1LGmWazxY8u4TB1ZkErvkBYoH1quEPuBUDgMbMzxPcP1Y+Oz
                4yHJJDnp/RVmRvQbEdBNc6N9Rvk97ahfYtTxP/jgdFcrGJ2BtMQo2pSXpXDrrB2+
                BxHw1dvd5Yzw1TKwg+ZX4o+/vqGqvz0dtdQ46tewXDpPaj+PwGZsY6rp2aQW9IHR
                lRQOfc2VNNnSj3BzgXucfr2YYdhFh5iQxeuGMMY1v/D/w1WIg0vvBZIGcfK4mJO3
                7M2CYfE45k+XmCpajQ==
                -----END CERTIFICATE-----
                </CertificateText>
            </CertificateDesc>
        </RootCertPackage>
        <RootCertPackage version="1.0">
            <id>2</id>
            <path>G1-G5</path>
            <CertificateDesc version="1.0">
                <CertificateFormat>PEM</CertificateFormat>
                <CertificateText>
                    -----BEGIN CERTIFICATE-----
                    MIICPTCCAaYCEQDNun9W8N/kvFT+IqyzcqpVMA0GCSqGSIb3DQEBAgUAMF8xCzAJ
                    BgNVBAYTAlVTMRcwFQYDVQQKEw5WZXJpU2lnbiwgSW5jLjE3MDUGA1UECxMuQ2xh
                    c3MgMSBQdWJsaWMgUHJpbWFyeSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw05
                    NjAxMjkwMDAwMDBaFw0yODA4MDEyMzU5NTlaMF8xCzAJBgNVBAYTAlVTMRcwFQYD
                    VQQKEw5WZXJpU2lnbiwgSW5jLjE3MDUGA1UECxMuQ2xhc3MgMSBQdWJsaWMgUHJp
                    bWFyeSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTCBnzANBgkqhkiG9w0BAQEFAAOB
                    jQAwgYkCgYEA5Rm/baNWYS2ZSHH2Z965jeu3noaACpEO+jglr0aIguVzqKCbJF0N
                    H8xlbgyw0FaEGIeaBpsQoXPftFg5a27B9hXVqKg/qhIGjTGsf7A01480Z4gJzRQR
                    4k5FVmkfeAKA2txHkSm7NsljXMXg1y2He6G3MrB7MLoqLzGq7qNn2tsCAwEAATAN
                    BgkqhkiG9w0BAQIFAAOBgQBMP7iLxmjf7kMzDl3ppssHhE16M/+SG/Q2rdiVIjZo
                    EWx8QszznC7EBz8UsA9P/5CSdvnivErpj82ggAr3xSnxgiJduLHdgSOjeyUVRjB5
                    FvjqBUuUfx3CHMjjt/QQQDwTw18fU+hI5Ia0e6E1sHslurjTjqs/OJ0ANACY89Fx
                    lA==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCEC0b/EoXjaOR6+f/9YtFvgswDQYJKoZIhvcNAQECBQAwXzELMAkG
A1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFz
cyAyIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTk2
MDEyOTAwMDAwMFoXDTI4MDgwMTIzNTk1OVowXzELMAkGA1UEBhMCVVMxFzAVBgNV
BAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAyIFB1YmxpYyBQcmlt
YXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GN
ADCBiQKBgQC2WoujDWojg4BrzzmH9CETMwZMJaLtVRKXxaeAufqDwSCg+i8VDXyh
YGt+eSz6Bg86rvYbb7HS/y8oUl+DfUvEerf4Zh+AVPy3wo5ZShRXRtGak75BkQO7
FYCTXOvnzAhsPz6zSvz/S2wj1VCCJkQZjiPDceoZJEcEnnW/yKYAHwIDAQABMA0G
CSqGSIb3DQEBAgUAA4GBAIobK/o5wXTXXtgZZKJYSi034DNHD6zt96rbHuSLBlxg
J8pFUs4W7z8GZOeUaHxgMxURaa+dYo2jA1Rrpr7l7gUYYAS/QoD90KioHgE796Nc
r6Pc5iaAIzy4RHT3Cq5Ji2F4zCS/iIqnDupzGUH9TQPwiNHleI2lKk/2lw0Xd8rY
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/Er4wDQYJKoZIhvcNAQEFBQAwXzELMAkG
A1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFz
cyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTk2
MDEyOTAwMDAwMFoXDTI4MDgwMjIzNTk1OVowXzELMAkGA1UEBhMCVVMxFzAVBgNV
BAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAzIFB1YmxpYyBQcmlt
YXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GN
ADCBiQKBgQDJXFme8huKARS0EN8EQNvjV69qRUCPhAwL0TPZ2RHP7gJYHyX3KqhE
BarsAx94f56TuZoAqiN91qyFomNFx3InzPRMxnVx0jnvT0Lwdd8KkMaOIG+YD/is
I19wKTakyYbnsZogy1Olhec9vn2a/iRFM9x2Fe0PonFkTGUugWhFpwIDAQABMA0G
CSqGSIb3DQEBBQUAA4GBABByUqkFFBkyCEHwxWsKzH4PIRnN5GfcX6kb5sroc50i
2JhucwNhkcV8sEVAbkSdjbCxlnRhLQ2pRdKkkirWmnWXbj9T/UWZYB2oK0z5XqcJ
2HUw19JlYD1n1khVdWk/kfVIC0dpImmClr7JyDiGSnoscxlIaU5rfGW/D/xwzoiQ
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDhDCCAwqgAwIBAgIQL4D+I4wOIg9IZxIokYesszAKBggqhkjOPQQDAzCByjEL
MAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQLExZW
ZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTowOAYDVQQLEzEoYykgMjAwNyBWZXJpU2ln
biwgSW5jLiAtIEZvciBhdXRob3JpemVkIHVzZSBvbmx5MUUwQwYDVQQDEzxWZXJp
U2lnbiBDbGFzcyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9y
aXR5IC0gRzQwHhcNMDcxMTA1MDAwMDAwWhcNMzgwMTE4MjM1OTU5WjCByjELMAkG
A1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQLExZWZXJp
U2lnbiBUcnVzdCBOZXR3b3JrMTowOAYDVQQLEzEoYykgMjAwNyBWZXJpU2lnbiwg
SW5jLiAtIEZvciBhdXRob3JpemVkIHVzZSBvbmx5MUUwQwYDVQQDEzxWZXJpU2ln
biBDbGFzcyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5
IC0gRzQwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAASnVnp8Utpkmw4tXNherJI9/gHm
GUo9FANL+mAnINmDiWn6VMaaGF5VKmTeBvaNSjutEDxlPZCIBIngMGGzrl0Bp3ve
fLK+ymVhAIau2o970ImtTR1ZmkGxvEeA3J5iw/mjgbIwga8wDwYDVR0TAQH/BAUw
AwEB/zAOBgNVHQ8BAf8EBAMCAQYwbQYIKwYBBQUHAQwEYTBfoV2gWzBZMFcwVRYJ
aW1hZ2UvZ2lmMCEwHzAHBgUrDgMCGgQUj+XTGoasjY5rw8+AatRIGCx7GS4wJRYj
aHR0cDovL2xvZ28udmVyaXNpZ24uY29tL3ZzbG9nby5naWYwHQYDVR0OBBYEFLMW
kf3upm7ktS5Jj4d4gYDs5bG1MAoGCCqGSM49BAMDA2gAMGUCMGYhDBgmYFo4e1ZC
4Kf8NoRRkSAsdk1DPcQdhCPQrNZ8NQbOzWm9kA3bbEhCHQ6qQgIxAJw9SDkjOVga
FRJZap7v1VmyHVIsmXHNxynfGyphe3HR3vPA5Q06Sqotp9iGKt0uEA==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIE0zCCA7ugAwIBAgIQGNrRniZ96LtKIVjNzGs7SjANBgkqhkiG9w0BAQUFADCB
yjELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
```

```
                    ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTowOAYDVQQLEzEoYykgMjAwNiBWZXJp
                    U2lnbiwgSW5jLiAtIEZvciBhdXRob3JpemVkIHVzZSBvbmx5MUUwQwYDVQQDEzxW
                    ZXJpU2lnbiBDbGFzcyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0
                    aG9yaXR5IC0gRzUwHhcNMDYxMTA4MDAwMDAwWhcNMzYwNzE2MjM1OTU5WjCByjEL
                    MAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQLExZW
                    ZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTowOAYDVQQLEzEoYykgMjAwNiBWZXJpU2ln
                    biwgSW5jLiAtIEZvciBhdXRob3JpemVkIHVzZSBvbmx5MUUwQwYDVQQDEzxWZXJp
                    U2lnbiBDbGFzcyAzIFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9y
                    aXR5IC0gRzUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCvJAgIKXo1
                    nmAMqudLO07cfLw8RRy7K+D+KQL5VwijZIUVJ/XxrcgxiV0i6CqqpkKzj/i5Vbex
                    t0uz/o9+B1fs70PbZmIVYc9gDaTY3vjgw2IIPVQT60nKWVSFJuUrjxuf6/WhkcIz
                    SdhDY2pSS9KP6HBRTdGJaXvHcPaz3BJ023tdS1bTlr8Vd6Gw9KIl8q8ckmcY5fQG
                    BO+QueQA5N06tRn/Arr0PO7gi+s3i+z016zy9vA9r911kTMZHRxAy3QkGSGT2RT+
                    rCpSx4/VBEnkjWNHiDxpg8v+R70rfk/Fla4OndTRQ8Bnc+MUCH7lP59zuDMKz10/
                    NIeWiu5T6CUVAgMBAAGjgbIwga8wDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8E
                    BAMCAQYwbQYIKwYBBQUHAQwEYTBfoV2gWzBZMFcwVRYJaW1hZ2UvZ2lmMCEwHzAH
                    BgUrDgMCGgQUj+XTGoasjY5rw8+AatRIGCx7GS4wJRYjaHR0cDovL2xvZ28udmVy
                    aXNpZ24uY29tL3ZzbG9nby5naWYwHQYDVR0OBBYEFH/TZafC3ey78DAJ80M5+gKv
                    MzEzMA0GCSqGSIb3DQEBBQUAA4IBAQCTJEowX2LP2BqYLz3q3JktvXf2pXkiOOzE
                    p6B4Eq1iDkVwZMXnl2YtmAl+X6/WzChl8gGqCBpH3vn5fJJaCGkgDdk+bW48DW7Y
                    5gaRQBi5+MHt39tBquCWIMnNZBU4gcmU7qKEKQsTb47bDN0lAtukixlE0kF6BWlK
                    WE9gyn6CagsCqiUXObXbf+eEZSqVir2G3l6BFoMtEMze/aiCKm0oHw0LxOXnGiYZ
                    4fQRbxC1lfznQgUy286dUV4otp6F01vvpX1FQHKOtw5rDgb7MzVIcbidJ4vEZV8N
                    hnacRHr2lVz2XTIIM6RUthg/aFzyQkqFOFSDX9HoLPKsEdao7WNq
                    -----END CERTIFICATE-----
                </CertificateText>
            </CertificateDesc>
        </RootCertPackage>
    </RootPackages>
</CertificateAuthority>
<CertificateAuthority version="1.0">
    <id>2</id>
    <DistinguishedName>
        <RDN>
            <RDNLongLabel>CountryName</RDNLongLabel>
            <RDNValue>ZA</RDNValue>
        </RDN>
        <RDN>
            <RDNLongLabel>StateOrProvinceName</RDNLongLabel>
            <RDNValue>Western Cape</RDNValue>
        </RDN>
        <RDN>
            <RDNLongLabel>LocalityName</RDNLongLabel>
            <RDNValue>Cape Town</RDNValue>
        </RDN>
        <RDN>
            <RDNLongLabel>OrganizationalName</RDNLongLabel>
            <RDNValue>Thawte Consulting cc</RDNValue>
        </RDN>
        <RDN>
            <RDNLongLabel>OrganizationalUnitName</RDNLongLabel>
            <RDNValue>Certification Services Division</RDNValue>
        </RDN>
```

```
            <RDN>
                <RDNLongLabel>CommonName</RDNLongLabel>
                <RDNValue> Thawte Server CA/emailAddress=server-certs@thawte.com </RDNValue>
            </RDN>
        </DistinguishedName>
        <crlDistributionPoint>http://crl.thawte.com/ThawteSGCCA.crl</crlDistributionPoint>
        <ocspInfoAccess>http://ocsp.thawte.com</ocspInfoAccess>
    </CertificateAuthority>
</CertificateAuthorityList>
```

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>

URI	/PSIA/CSEC/certificateAuthority/<id>	**Type**	Resource
Function	Resource used to manage a particular <CertificateAuthority>.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<CertificateAuthority>
PUT	None	<CertificateAuthority>	<ResponseStatus>
POST	N/A	N/A	<ResponseStatus w/error code >
DELETE	None	None	<ResponseStatus>
Notes	See previous section for XML example.		

6.4.2 /PSIA/CSEC/certificateAuthority/<id>/rootPackages

URI	/PSIA/CSEC/certificateAuthority/<id>/rootPackages		Type	Resource
Function	Resource used to manage a particular <CertificateAuthority>'s Root Certificate packages (series of Base64 CER or PEM) in a "raw" fashion.			
Methods	Query String(s)	Inbound Data	Return Result	
GET	None	None	Block of all Base64CERs/PEMs	
PUT	None	Block of all Base64CERs/PEMs	<ResponseStatus>	
POST	N/A	N/A	<ResponseStatus w/error code >	
DELETE	None	None	<ResponseStatus>	
Notes				

6.5 /PSIA/CSEC/KeyManager

URI	/PSIA/CSEC/KeyManager		Type	Service
Methods	Query String(s)	Inbound Data	Return Result	
Notes	Common Security (CSEC) Service's KeyManager Service			

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

URI	***[HTTPS ONLY]*** /PSIA/CSEC/KeyManager/directMKIKeyList		Type	Resource
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.			
Methods	Query String(s)	Inbound Data	Return Result	
GET	None	None	<MKIKeyList>	

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:

```
<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: -->
      <rtspSessionID>2605004428</rtspSessionID>
      <!-- OPTIONAL (only needed if re-keying different MKI for each RTP Session): -->
      <ssrcID>1D623AC5</ssrcID>
    </MKIMediaContext>
    <!-- Base64 of '1234567890abcdef1234567890abcdef': -->
    <masterKey>MTIzNDU2Nzg5MGFiY2RlZjEyMzQ1Njc4OTBhYmNkZWY=</masterKey>
    <!-- Base64 of '3b04803de51ee7c96423ab5b78d2': -->
    <masterSalt>M2IwNDgwM2RlNTFlZTdjOTY0MjNhYjViNzhkMg==</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 = 281474976710656): -->
    <srtpKeyLifetime>281474976710656</srtpKeyLifetime>
    <!-- OPTIONAL - decimal units of pkt-count, must be 0 or power of 2: -->
    <srtcpKeyDerivationRate>0</srtcpKeyDerivationRate>
    <!-- OPTIONAL - decimal units of pkt-count (max/def for srtp is 2^31 = 2147483648): -->
    <srtcpKeyLifetime>2147483648</srtcpKeyLifetime>
    <!-- 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: -->
      <rtspSessionID>2605004428</rtspSessionID>
      <!-- OPTIONAL (only needed if re-keying different MKI for each RTP Session): -->
      <ssrcID>1D623AC5</ssrcID>
    </MKIMediaContext>
    <!-- Base64 of 'abcdef0123456789abcdef0123456789': -->
    <masterKey>YWJjZGVmMDEyMzQ1Njc4OWFiY2RlZjAxMjM0NTY3ODk=</masterKey>
    <!-- Base64 of '1c13457fa5de3dc26f2354ada609': -->
    <masterSalt>MWMxMzQ1N2ZhNWRlM2RjMjZmMjM1NGFkYTYwOQ====</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 = 281474976710656): -->
    <srtpKeyLifetime>281474976710656</srtpKeyLifetime>
    <!-- OPTIONAL - decimal units of pkt-count, must be 0 or power of 2: -->
    <srtcpKeyDerivationRate>0</srtcpKeyDerivationRate>
    <!-- OPTIONAL - decimal units of pkt-count (max/def for srtp is 2^31 = 2147483648): -->
    <srtcpKeyLifetime>2147483648</srtcpKeyLifetime>
  </MKIKey>
</MKIKeyList>
```

6.5.2 /PSIA/CSEC/KeyManager/directMKIKeyList/<MKI>

URI	***[HTTPS ONLY]*** /PSIA/CSEC/KeyManager/directMKIKeyList/<MKI>	**Type**	Resource
Function	**On a secured TLS/SSL session**: This Resource is used access a specific MKI context.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<MKIKey>
PUT	None	<MKIKey>	<ResponseStatus>
POST	N/A	N/A	<ResponseStatus w/error code>
DELETE	None	None	<ResponseStatus>
Notes	See previous section for example XML.		

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

URI	***[HTTPS ONLY]*** /PSIA/CSEC/KeyManager/directMKIKeyList/<MKI>/mime	**Type**	Resource
Function	**On a secured TLS/SSL session**: This Resource is used get the key and salt (concatentated) without XML encapsulation.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	BASE64 of key||salt
PUT	N/A	N/A	<ResponseStatus w/error code>
POST	N/A	N/A	<ResponseStatus w/error code>
DELETE	N/A	N/A	<ResponseStatus w/error code>
Notes			

6.5.4 /PSIA/CSEC/KeyManager/schemes

URI	/PSIA/CSEC/KeyManager/schemes	Type	Resource
Methods	Query String(s)	Inbound Data	Return Result
Notes	Common Security (CSEC) Service's key management schemes resource		

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

URI	*[HTTPS RECOMMENDED]* /PSIA/CSEC/KeyManager/schemes/negotiatedMKIList	Type	Resource
Function	This Resource is used get a list of negotiated MKI values.		
Methods	Query String(s)	Inbound Data	Return Result
GET	None	None	<NegotiatedMKIList>
PUT	N/A	N/A	<ResponseStatus w/error code>
POST	N/A	N/A	<ResponseStatus w/error code>
DELETE	N/A	N/A	<ResponseStatus w/error code>
Notes			

Example 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>

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

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

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

6.5.8 /PSIA/CSEC/KeyManager/schemes/MIKEY

URI	`/PSIA/CSEC/KeyManager/schemes/MIKEY`	**Type**	Resource
Function	This Resource is used to query the advertised MIKEY (see RFC 3830) resource properties.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<MIKEYProperties>
PUT	N/A	N/A	<ResponseStatus w/error code>
POST	N/A	N/A	<ResponseStatus w/error code>
DELETE	N/A	N/A	<ResponseStatus w/error code>
Notes			

Example XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<MIKEYProperties version="1.0" xmlns="urn:psialliance-org">
  <!-- Required: -->
  <MIKEYExchangeMethod>PSK</MIKEYExchangeMethod>
  <!-- OPTIONAL: -->
  <MIKEYExchangeMethod>PKE</MIKEYExchangeMethod>
  <!-- OPTIONAL/RECOMMENDED if Perfect Forward Security (PFS) is required: -->
  <MIKEYExchangeMethod>DH</MIKEYExchangeMethod>
  <!-- This indicates presence of the /tunnel resource: -->
  <MIKEYTransport>PSIA-REST-TUNNEL</MIKEYTransport>
  <!-- OPTIONAL, if RFC4567 is supported: -->
  <MIKEYTransport>RTSP</MIKEYTransport>
</MIKEYProperties>
```

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 ---+
                                         |
                                         v
               +-----------+ master  +--------+ session encr_key
               | ext       | key     |        |---------->
               | key mgmt  |-------->|  key   | session auth_key
               | (optional |         | deriv  |---------->
               | rekey)    |-------->|        | session salt_key
               |           | 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

URI	***[HTTPS ONLY]*** /PSIA/CSEC/KeyManager/schemes/MIKEY/tunnel		**Type**	Resource
Function	**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.			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET	None	<MIKEYMessage>	<MIKEYMessage> or <ResponseStatus w/error code>	
PUT	N/A	N/A	<ResponseStatus w/error code>	
POST	N/A	<MIKEYMessage>	<MIKEYMessage> or <ResponseStatus w/error code>	
DELETE	N/A	N/A	<ResponseStatus w/error code>	
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 AQQFgAAATbcCAAAAAHK/AAAAAAAAAAAAAAAAAAoAx9bH1P3ztk LAAAAJwABAQEBEAIBAQMBFAQBDgUBAAYBAAcBAQgBAQkBAAoBAQsBCgwBAAcQrp33V4S04/yprsxz2nytcQMC BpMwggaPMIIEd6ADAgE CAgkA8+z1SAxBJE4wDQYJKoZIhvcNAQEFBQAwgYsxCzAJB". 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 version="1.0" encoding="utf-8" ?>
<MIKEYMessage version="1.0" xmlns="urn:psialliance-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>
  <Message>

AQQFgAAATbcCAAAAAHK/AAAAAAAAAAAAAAAAAAoAx9bH1P3ztkLAAAAJwABAQEBEAIBAQMBFAQBDgUBAAYBAAcBAQgBAQkBAAoBA
QsBCgwBAAcQrp33V4S04/yprsxz2nytcQMCBpMwggaPMIIEd6ADAgECAgkA8+z1SAxBJE4wDQYJKoZIhvcNAQEFBQAwgYsxCzAJB
  </Message>
</MIKEYMessage>
```

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
```

```
                     |    ";" "mode" = <"> 1\#mode <">
ttl                  =    1*3(DIGIT)
port                 =    1*5(DIGIT)
ssrc                 =    8*8(HEX)
channel              =    1*3(DIGIT)
address              =    host
mode                 =    <"> *Method <"> | Method
```

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

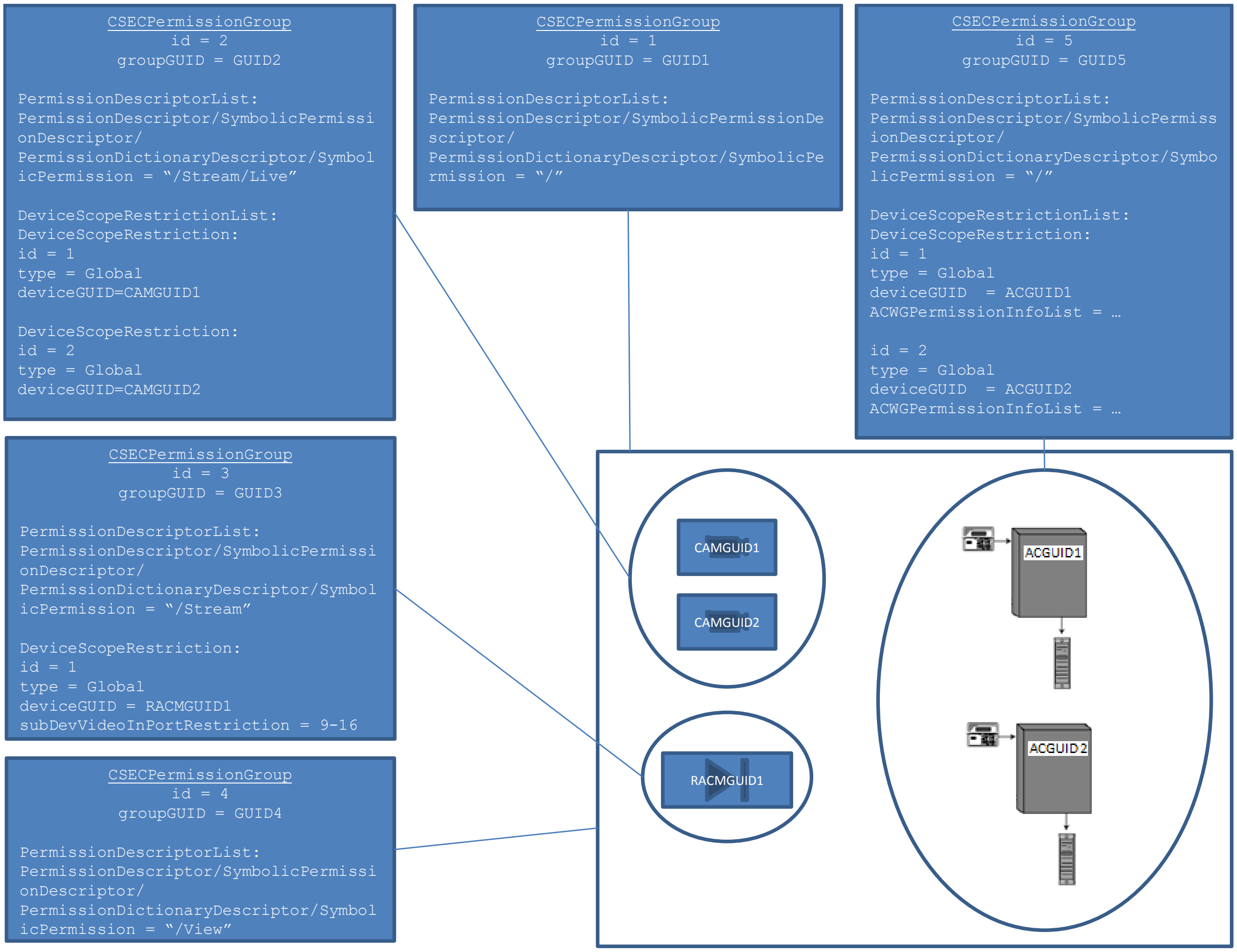
A Permission Group collects Permissions within its "PermissionDescriptorList". A Permission Group is identified by its local REST ID ("id") or its global ID ("groupGUID"). The Group is also the container for "Restrictions".

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.



```
CSECPermissionGroup
id = 2
groupGUID = GUID2

PermissionDescriptorList:
PermissionDescriptor/SymbolicPermissi
onDescriptor/
PermissionDictionaryDescriptor/Symbol
icPermission = "/Stream/Live"

DeviceScopeRestrictionList:
DeviceScopeRestriction:
id = 1
type = Global
deviceGUID=CAMGUID1

DeviceScopeRestriction:
id = 2
type = Global
deviceGUID=CAMGUID2
```

```
CSECPermissionGroup
id = 1
groupGUID = GUID1

PermissionDescriptorList:
PermissionDescriptor/SymbolicPermissionDe
scriptor/
PermissionDictionaryDescriptor/SymbolicPe
rmission = "/"
```

```
CSECPermissionGroup
id = 5
groupGUID = GUID5

PermissionDescriptorList:
PermissionDescriptor/SymbolicPermiss
ionDescriptor/
PermissionDictionaryDescriptor/Symbo
licPermission = "/"

DeviceScopeRestrictionList:
DeviceScopeRestriction:
id = 1
type = Global
deviceGUID  = ACGUID1
ACWGPermissionInfoList = …

id = 2
type = Global
deviceGUID  = ACGUID2
ACWGPermissionInfoList = …
```

```
CSECPermissionGroup
id = 3
groupGUID = GUID3

PermissionDescriptorList:
PermissionDescriptor/SymbolicPermissi
onDescriptor/
PermissionDictionaryDescriptor/Symbol
icPermission = "/Stream"

DeviceScopeRestriction:
id = 1
type = Global
deviceGUID = RACMGUID1
subDevVideoInPortRestriction = 9-16
```

```
CSECPermissionGroup
id = 4
groupGUID = GUID4

PermissionDescriptorList:
PermissionDescriptor/SymbolicPermissi
onDescriptor/
PermissionDictionaryDescriptor/Symbol
icPermission = "/View"
```

Figure 2: Example Groups

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.

CORE Profile:

Resource	**GET**	**PUT**	**POST**	**DELETE**
/PSIA/CSEC/AAA/users	✓			
/PSIA/CSEC/AAA/users/<*id*>	✓			
/PSIA/CSEC/AAA/users/<*GUID*>	✓			
/PSIA/CSEC/permissionGroups	✓			
/PSIA/CSEC/permissionGroups/<*id*>	✓			
/PSIA/CSEC/supportedPermissions	✓			

BASIC and FULL Profiles:Resource	**GET**	**PUT**	**POST**	**DELETE**
/PSIA/CSEC/AAA/users	✓	✓	✓	
/PSIA/CSEC/AAA/users/<*id*>	✓	✓		✓
/PSIA/CSEC/AAA/users/<*GUID*>	✓	✓		✓
/PSIA/CSEC/permissionGroups	✓	✓	✓	
/PSIA/CSEC/permissionGroups/<*id*>	✓	✓		✓
/PSIA/CSEC/supportedPermissions	✓			

7.6.1 /PSIA/CSEC/AAA/users

URI	**[HTTPS ONLY]** /PSIA/CSEC/AAA/users	**Type**	Resource
Function	Resource used to manage the <CSECUserList>.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<CSECUserList>
PUT	None	<CSECUserList>	<ResponseStatus>
POST	None	<CSECUser>	<ResponseStatus>
DELETE	N/A	N/A	<ResponseStatus w/error code>
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 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>777777</CODE>
        <CODEClass>FIRE</CODEClass>
      </UserCODE>
    </UserCODEList>
    <UserPermissionGroupId>1</UserPermissionGroupId>
  </CSECUser>
  <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>
    <UserPermissionGroupId>4</UserPermissionGroupId>
  </CSECUser>
  <CSECUser version="1.0">
    <id>5</id>
    <userEnabled>true</userEnabled>
    <userDescription>Superviser for Intrusion A only</userDescription>
    <fullName>Frank Able</fullName>
    <userGUID>{3f06288a-4c73-4d63-b8b1-a22817cce2ca}</userGUID>
    <UserCODEList>
      <UserCODE>
        <id>1</id>
        <CODE>654321</CODE>
        <CODEClass>INTRUSION</CODEClass>
      </UserCODE>
    </UserCODEList>
    <UserPermissionGroupId>5</UserPermissionGroupId>
  </CSECUser>
</CSECUserList>
```

6.6.1.1/PSIA/CSEC/AAA/users/<id>

URI	***[HTTPS ONLY]*** /PSIA/CSEC/AAA/users/<id>		**Type**	Resource
Function	Resource used to manage a particular <CSECUser> identified by its local REST ID.			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET	None	None	<CSECUser>	
PUT	None	<CSECUser>	<ResponseStatus>	
POST	N/A	N/A	<ResponseStatus w/error code>	
DELETE	None	None	<ResponseStatus>	
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>

URI	*[HTTPS ONLY]* /PSIA/CSEC/AAA/users/<GUID>		Type	Resource
Function	Resource used to manage a particular <CSECUser> identified by its User GUID.			
Methods	Query String(s)	Inbound Data	Return Result	
GET	None	None	<CSECUser>	
PUT	None	<CSECUser>	<ResponseStatus>	
POST	N/A	N/A	<ResponseStatus w/error code>	
DELETE	None	None	<ResponseStatus>	
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

URI	/PSIA/CSEC/AAA/permissionGroups		Type	Resource
Function	Resource used to manage the <CSECPermissionGroupList>.			
Methods	Query String(s)	Inbound Data	Return Result	
GET	None	None	<CSECPermissionGroupList>	
PUT	None	<CSECPermissionGroupList>	<ResponseStatus>	
POST	None	<CSECPermissionGroup>	<ResponseStatus>	

DELETE	N/A	N/A	<ResponseStatus w/error code>
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 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>
    <!-- no <DeviceScopeRestrictionList> -->
  </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>
      <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>
```

```
      <PermissionDescriptor>
        <id>2</id>
        <SymbolicPermissionDescriptor>
          <PermissionDictionaryDescriptor>
            <dictionaryName>PSIA-CSEC-D1</dictionaryName>
            <SymbolicPermission>/Stream/Recorded/Media/Video</SymbolicPermission>
          </PermissionDictionaryDescriptor>
        </SymbolicPermissionDescriptor>
      </PermissionDescriptor>
    </PermissionDescriptorList>
    <!-- RESTRICTION: 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>

    <!-- RESTRICTION: Panel Permission definition -->
    <DeviceScopeRestrictionList>
      <DeviceScopeRestriction>
        <id>1</id>
        <restrictionDescription>Panel Portal 5</restrictionDescription>
        <type>Local</type>
        <ACWGPermissionInfoList>
          <PermissionInfo>
            <ID>1</ID>
            <PrivilegeList>
              <Privilege>
                <Allow>
                  <AuthorityLevel>1</AuthorityLevel>
                  <PortalIDList>
                    <PortalID>
                      <ID>1</ID>
                      <ID>2</ID>
                      <ID>3</ID>
                    </PortalID>
                  </PortalIDList>
                </Allow>
              </Privilege>
            </PrivilegeList>
          </PermissionInfo>
        </ACWGPermissionInfoList>
      </DeviceScopeRestriction>
    </DeviceScopeRestrictionList>
  </CSECPermissionGroup>

</CSECPermissionGroupList>
```

6.6.2.1/PSIA/CSEC/AAA/permissionGroups/<id>

URI	/PSIA/CSEC/AAA/permissionGroups/<id>	**Type**	Resource
Function	Resource used to manage a particular <CSECPermissionGroup>.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<CSECPermissionGroup>
PUT	None	<CSECPermissionGroup>	<ResponseStatus>
POST	N/A	N/A	<ResponseStatus w/error code >
DELETE	None	None	<ResponseStatus>
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

URI	/PSIA/CSEC/AAA/supportedPermissions	**Type**	Resource
Function	Resource used get a <PermissionDescriptorList> of the supported permissions in the Device.		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET	None	None	<PermissionDescriptorList>
PUT	N/A	N/A	<ResponseStatus w/error code>
POST	N/A	N/A	<ResponseStatus w/error code>
DELETE	None	None	<ResponseStatus w/error code>
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 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>
```

```
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
      <id>3</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/Statistics</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <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>
    <PermissionDescriptor>
      <id>8</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/System/StorageSettings</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
      <id>9</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/System/SerialPortSettings</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
      <id>10</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/System/Diagnostics</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
      <id>11</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/System/TimeSettings</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
      <id>12</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/VideoDevice</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
    <PermissionDescriptor>
      <id>13</id>
      <SymbolicPermissionDescriptor>
        <PermissionDictionaryDescriptor>
          <dictionaryName>PSIA-CSEC-D1</dictionaryName>
          <SymbolicPermission>/View/VideoDevice/Overlays</SymbolicPermission>
        </PermissionDictionaryDescriptor>
      </SymbolicPermissionDescriptor>
    </PermissionDescriptor>
```

```xml
  <PermissionDescriptor>
    <id>14</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/VideoDevice/PrivacyMasks</SymbolicPermission>
      </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
  </PermissionDescriptor>
  <PermissionDescriptor>
    <id>15</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/VideoDevice/OtherSettings</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/OtherSettings</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>
  <PermissionDescriptor>
    <id>26</id>
    <SymbolicPermissionDescriptor>
      <PermissionDictionaryDescriptor>
        <dictionaryName>PSIA-CSEC-D1</dictionaryName>
        <SymbolicPermission>/View/PTZ/TemporaryPresets</SymbolicPermission>
      </PermissionDictionaryDescriptor>
    </SymbolicPermissionDescriptor>
  </PermissionDescriptor>
</PermissionDescriptorList>
```

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

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=crypto:1 AES_CM_128_HMAC_SHA1_80
 inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|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
a=crypto:1 AES_CM_128_HMAC_SHA1_80
 inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^48|10:4
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:

	AES_CM_128_ HMAC_SHA1_80	AES_CM_128_ HMAC_SHA1_32	F8_128_ HMAC_SHA1_80
Master key length	128 bits	128 bits	128 bits
Master salt length	112 bits	112 bits	112 bits
SRTP lifetime	2^48 packets	2^48 packets	2^48 packets
SRTCP lifetime	2^31 packets	2^31 packets	2^31 packets
Cipher	AES Counter Mode	AES Counter Mode	AES F8 Mode
Encryption key	128 bits	128 bits	128 bits
MAC	HMAC-SHA1	HMAC-SHA1	HMAC-SHA1
SRTP auth. tag	80 bits	32 bits	80 bits
SRTCP auth. tag	80 bits	80 bits	80 bits
SRTP auth. key len.	160 bits	160 bits	160 bits
SRTCP auth. key len.	160 bits	160 bits	160 bits

From “srtp-big-aes-3”:

Parameter	Value
Master key length	192 bits
Master salt length	112 bits
Key Derivation Function	AES_192_CM_PRF (Section 3)
Default key lifetime	2^31 packets
Cipher (for SRTP and SRTCP)	AES_192_CM (Section 2)
SRTP authentication function	HMAC-SHA1 (Section 4.2.1 of [RFC3711])
SRTP authentication key length	160 bits
SRTP authentication tag length	80 bits
SRTCP authentication function	HMAC-SHA1 (Section 4.2.1 of [RFC3711])
SRTCP authentication key length	160 bits
SRTCP authentication tag length	80 bits

Table 1: The AES_192_CM_HMAC_SHA1_80 cryptosuite.

Parameter	Value
Master key length	192 bits
Master salt length	112 bits
Key Derivation Function	AES_192_CM_PRF (Section 3)
Default key lifetime	2^31 packets
Cipher (for SRTP and SRTCP)	AES_192_CM (Section 2)
SRTP authentication function	HMAC-SHA1 (Section 4.2.1 of

	[RFC3711])
SRTP authentication key length	160 bits
SRTP authentication tag length	32 bits
SRTCP authentication function	HMAC-SHA1 (Section 4.2.1 of [RFC3711])
SRTCP authentication key length	160 bits
SRTCP authentication tag length	80 bits

Table 2: The AES_192_CM_HMAC_SHA1_32 cryptosuite.

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

Table 3: The AES_256_CM_HMAC_SHA1_80 cryptosuite.

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

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[1].

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.