SMPTE ST 430-6:2010
Revision of
SMPTE 430-6-2008

SMPTE STANDARD

D-Cinema Operations — Auditorium Security Messages for Intra-Theater Communications

Table of Contents

Approved
August 18, 2010

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

SMPTE ST 430-6 was prepared by Technology Committee 21DC.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

1 Scope

The Auditorium Security Message (ASM) specification enables interoperable communication of security-critical information (information necessary to ensure security of D-Cinema content) between devices over an intra-theater exhibition network. The specification uses Transport Layer Security (TLS) for authentication and confidentiality, and Key-Length-Value (KLV) coding for message encoding. It defines a protocol, a general purpose request-response message set and a specific message set for link encryption keying.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

[336M] SMPTE 336M-2007, Data Encoding Protocol Using Key-Length-Value

[Dcert] SMPTE 430-2-2006, D-Cinema Operations — Digital Certificate

[IANA] Internet Assigned Numbers Authority. See www.iana.org/assignments/port-numbers

[KDM] SMPTE 430-1-2006, D-Cinema Operations — Key Delivery Message

[Log] SMPTE 430-5-2008, D-Cinema Packaging — Security Log Event Class and Constraints

[TLS] “The TLS Protocol, Version 1.0” RFC 2246 See www.ietf.org/rfc/rfc2246.txt

[TLS-AES] “AES Cyphersuites for TLS” RFC 3268 See www.ietf.org/rfc/rfc3268.txt

4 Glossary

The following acronyms are used in this specification:

ASM	Auditorium Security Message
AES	Advanced Encryption Standard
BER	Basic Encoding Rules (ASN.1)
CBC	Cipher Block Chaining
IMB	Image Media Block
KLV	Key Length Value
LDB	Link Decryptor Block
LE	Link Encryption
RRP	Request Response Pair
RSA	Rivest Shamir Adleman public key encryption
SHA-1	Secure Hash Algorithm revision 1
SM	Security Manager
SPB	Secure Processing Block
TLS	Transport Layer Security
Uintx	Unsigned x bit integer
UL	Universal Label
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier (ISO 11578)

5 Overview (Informative)

Exhibition security equipment configurations which employ remote Secure Processing Blocks (SPBs) (i.e., SPBs which are remote from that which contains the Security Manager) require a secure method of communicating with such SPBs. The generic model for this is illustrated in Figure 1.

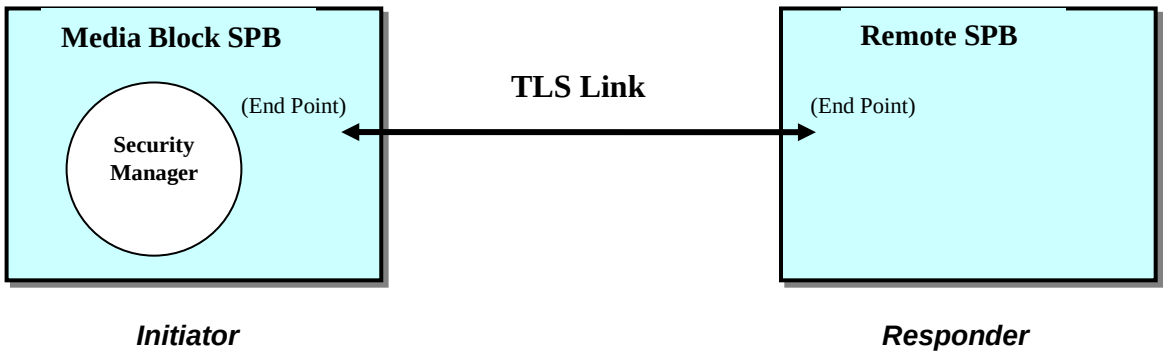

Figure 1 – Auditorium Security Message Model

The communication security protection mechanism needs to provide (1) confidentiality, (2) integrity, (3) authentication and (4) prevention of replay. In addition, the mechanism needs to be inexpensive to implement, and simple to support in secure silicon processors.

Message descriptions are given in terms of the Initiator and Responder (and this specification makes no distinction between messages emanating from the Security Manager vs. the Image Media Block that contains it). As used herein the generic name for a "block" is SPB.

6 Message Security, RRP Structure and General Requirements

The implementation of Auditorium Security Messages (ASM) shall be in the form of a "Request" from the Initiator followed by a "Response" from the Responder (recipient SPB). Each pair of messages is referred to as a Request-Response Pair (RRP).

6.1 Message Security: Transport Layer Security (TLS)

Message security shall be provided by communicating ASMs under Transport Layer Security (TLS) (see [TLS]). During TLS session establishment, the Initiator (which contains the Security Manager) and Responder exchange their X.509 certificates as part of the initial TLS handshake. This exchange shall be supported using D-Cinema compliant certificates as defined in the D-Cinema Digital Certificate specification [DCert].

The TLS protocol shall be TLS 1.0 [TLS] constrained as follows to simplify implementation, facilitate interoperability and ensure predictable processing:

1. 2048-bit RSA using a public exponent value of 65537 shall be the only supported public key algorithm.
2. AES-CBC 128-bit shall be the only supported symmetric cipher (see [TLS-AES]).
3. SHA-1 shall be the only supported hash algorithm.
4. The CipherSuite shall be "TLS_RSA_WITH_AES_128_CBC_SHA" (0x00, 0x2F) (see [TLS-AES]).
5. The TLSCiphertext.length ([TLS] section 6.2.3.) shall be less than or equal to 512 bytes (see Annex C).
6. The Compression Method shall be "null" (no compression).
7. Other than as part of the opening handshake, the ChangeCipherSpec message shall be ignored.
8. When performing TLS handshake mutual authentication, it shall be permissible for the TLS client and server devices to exchange only the respective SPB device leaf certificate.

6.2 Message Structure: Key Length Value (KLV)

Request and Response ASMs shall be Key Length Value (KLV) encoded using Fixed Length Pack encoding according to SMPTE 336M-2001 [336M]. The Fixed Length Universal Label (UL) Key is given in Annex A of this document. As a Fixed Length Pack, each individual item in the Value field comprises only an item value. The KLV Length field shall be a long-form BER value encoded with a fixed length of 4 bytes total.

Example: For a KLV packet having a Value field that is 12 bytes in length, the Length field would be encoded as the following 4 bytes, 0x83 0x00 0x00 0x0C (hexadecimal).

Each ASM Request-Response Pair (RRP) represents two message types and thus KLV UL "value" registration is required twice for each defined RRP (see Annex A).

Note: The recipient of each RRP Request or Response command is implicit by virtue of the TLS socket (which is known at the applications level) that carries the messages.

6.3 General ASM Command Elements

For each message type, the following shall apply:

- The command type is denoted within the opening KLV “Key” field (16 bytes).
- “Length” is a BER-encoded four byte field which describes the length of the message in bytes.
- “Request_ID” shall be an application level tag for the Request, which shall be echoed by the corresponding Response. A non-zero Request_ID value shall be set by the SM, which should select unique values (e.g. a sequencing counter) for each TLS connection it manages. (Request_ID generation means is left to implementers and is out of scope of this specification.)
- Multi-byte integer values shall be sent as big-endian data, meaning most significant byte first.
- “Event ID” shall be a 32-bit value divided into two components:

    o A 13 bit “Event Log” component:
    The Event Log shall be generated in a sequential (increasing) manner in order to match the Associated log records time sequence. Event Log values shall begin at 0x0000 and end at 0x1FFF. Once 0x1FFF is reached, the next Event Log value shall be 0x0000.

    These 13 bits are the 13 least significant bits (LSB) of the 32 bits.

    o A 19 bit “User Defined Value” component:
    The definition of these 19 bits is out of the scope of this document and may vary from one implementation to the other.

    These 19 bits are the 19 most significant bits (MSB) of the 32 bits.

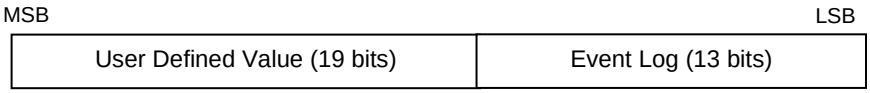

Figure 2 – Event ID Structure

Note: The Event ID as defined above and as used through this document is different from the Event ID defined in SMPTE 430-4, despite the fact that they share the same name. This Event ID is a Uint32, the Event ID in SMPTE 430-4 is a UUID.

General “Response” elements for each Response command are defined as follows:

General Response Elements

Element	Meaning	UInt8 Value
RRP successful	Request successfully processed	0
RRP failed	Responder unable to process Request	1
RRP Invalid	Invalid parameter or command structure	2
ResponderBusy	Responder too busy to process Request	3

Messages defined in this document may contain batches. A batch is a compound data type that is created from combinations of simple data types. It is usually preceded by a name (e.g., an EventIDBatch is an unordered batch of Event ID values):

Batch: A compound type comprising multiple individual elements. The elements are unordered, the type is defined, the count of elements is explicit and the size of each element is fixed and explicit.

xxxBatch: A batch of zero or more individual elements of name "xxx" preceded by a header of 8 bytes. The first 4 bytes of the header define the number of elements to follow and the second 4 bytes define the length of each element, both represented as UInt32.

Item Name	Type	Len	UL	Meaning	Default
Number of Items	Uint32	4	n/a	The number of Items in the Batch	n
Item Length	Uint32	4	n/a	The length of each Item	L
First component of first instance of xxx	...	...		First of one or more components describing element 'xxx' and having a total length of L	...

6.4 General TLS and RRP Requirements for Auditorium Security Messages

This section defines implementation constraints for security assurance, interoperability, RRP contention management and serendipity with other exhibition subsystems which may use network resources shared by these security functions.

1. TLS sessions shall be established by the Initiator following the standard applications level TLS handshake protocol using mutual authentication mode (see [TLS]). Mutual authentication shall exchange both TLS client (Initiator) and server (Responder) D-Cinema compliant certificates.

   Note*:* Certificate utility at each TLS end point is out of scope of this specification; however the purpose of mutual authentication is to enable the Responder (remote SPB) to receive the Initiator's (Image Media Block) certificate to record its thumbprint for logging purposes.

2. RRP protocols shall be synchronous (i.e., each pairing shall be opened and closed before a new RRP is opened between the same two SPBs). To avoid hang-ups, RRP Responder implementations should be designed to support maximum round-trip Request-to-Response latencies as specified in the message definition sections below. Latency shall be measured from the end of the "Request" message receipt to the start of the "Response" message transmission. Responders unable to transmit the Response within the specified limit because of a "busy" condition should close that RRP duple by issuance of a BadRequest Response with the general Response element indicating "busy" per the General Response Elements table in Section 6.3.

   Note: Should the Responder fail to respond (at all) after the specified time limit, the Initiator may consider this a communications failure condition and may, for instance, close and restart the TLS session.

3. SMPTE standardized ASM security messages shall use well-known port 1173, which has been reserved for D-Cinema "security" RRPs by the Internet Assigned Numbers Authority (see [IANA]).

   Note: Non-standardized, or non-security related RRPs may exist to support other functionality; however, such RRPs should use a different port.

7 General Purpose ASM Commands

This section defines ASM commands which support remote SPBs generally (i.e., independently of the specific type of SPB or contained security functions). Table 1 shows these commands together with the names as recorded in the SMPTE UL metadata registries.

Request-Response round trip latency – Per item (2) of Section 6.4, Responder implementations should support a maximum round-trip Request-to-Response latency of 2 seconds for general purpose ASM commands.

Table 1 – General Purpose ASM Command Types

General Purpose ASM Commands	SMPTE Metadata General Purpose ASM Command UL Name
BadRequest Request	Bad Request Response
GetTime_ Request	Time Request
GetTime Response	Time Response
GetEventList Request	Event List Request
GetEventList Response	Event List Response
GetEventID Request	Event ID Request
GetEventID Response	Event ID Response
QuerySPB Request	Secure Processing Block Query Request
QuerySPB Response	Secure Processing Block Query Response
GetProjCert Request	Projector Certificate Request
GetProjCert Response	Projector Certificate Response

7.1 BadRequest Response

Each RRP Response command contains a general “Response” element; however instances may arise where the Responder does not understand the incident Request command that was received. In such case the appropriate Response command would be unknown. The BadRequest Response shall be used when the Recipient cannot otherwise respond with a Response command appropriate to the incident Request. A complete copy of the Request command as received by the Responder is carried in this message (an exception is noted below).

BadRequest Response

Item Name	Type	Len	UL	Meaning
Bad Request Response UL Name	Pack Key	16		Identifies the BadRequest Response
Length	BER Length	4	n/a	Pack length
Request Copy	Text	Var		Copy of Request
Response	Uint8	1		General Response info

- The Request_Copy is a complete copy of the Request command that was not understood.
- The length of Request_Copy can be determined from the Length of the message.
- No copy of the incident Request shall be carried in the BadRequest Response in the event that a Responder is “busy” per the General Response Elements table of Section 6.3. In such case the “Request Copy” field shall be null; that is, of length zero.

7.2 GetTime

The GetTime command returns a snapshot of the Responder's absolute UTC time. The units of Time shall be seconds.

GetTime Request

Item Name	Type	Len	UL	Meaning
Time Request UL Name	Pack Key	16		Identifies the GetTime Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request

GetTime Response

Item Name	Type	Len	UL	Meaning
Time Response UL Name	Pack Key	16		Identifies the GetTime Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of the Request for which this is the Response
Time	Uint64	8		Responder's time
Response	Uint8	1		General Response info

- Time shall be a 64-bit integer representing the number of seconds elapsed since January 1, 1970 00:00:00 UTC. The Time value shall be taken at the instant that the GetTime Response message is queued for transmission to the Initiator.

Note: The Recipient device clock reports absolute time (UTC). The SM will use the GetTime command to determine the difference between true time (the SM's time) and time in the remote SPB, and remove the delta in log reporting. Accuracy requirements are out of scope of this specification, but it is assumed that corrections for clock drift or other offsets (e.g. leap seconds) are unnecessary.

7.3 GetEventList

The GetEventList command identifies a UTC TimeStart and TimeStop period for which the Responder will respond with a list of identified logged event records which it holds.

Note: By appropriate management of start/stop periods, the Initiator is expected to keep track of the collection of log information from a remote SPB and assure no gaps exist across linear time. There are no restrictions placed upon the Initiator regarding the time start/stop times (e.g., book-ending or overlaps) or whether log information has been previously collected. A Responder will simply respond to all Requests. Requirements for when log data is collected or how long it is retained by remote SPBs are out of scope of this specification.

GetEventList Request

Item Name	Type	Len	UL	Meaning
Event List Request UL Name	Pack Key	16		Identifies the GetEventList Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request
TimeStart	Uint32	4		Event list period start
TimeStop	Uint32	4		Event list period stop

- The TimeStart and TimeStop elements define a UTC based time window for which logged event information is being requested.

GetEventList Response

Item Name	Type	Len	UL	Meaning
Event List Response UL Name	Pack Key	16		Identifies the GetEventList Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
EventIDBatch	Event ID Batch	8+4n		An unordered Batch of Event ID values (see table below)
Response	Uint8	1		General Response info

EventIDBatch

Item Name	Type	Len	UL	Meaning	Default
Number of Items	Uint32	4	n/a	The number of Items in the Batch	n
Item Length	Uint32	4	n/a	The length of each Item	4
Event ID	Uint32	4		Unique event identifier(s)	

- The EventIDBatch is a batch of Event IDs. These Event IDs are in no particular order. There shall be only one entry for each Event ID in the EventIDBatch.
- The length of the variable portion of EventIDBatch can be determined by multiplying the Number_of_Items by the Item_length.
- The batch format returns the Event ID(s) for all the events recorded with time stamps within the specified Request time window (including the TimeStart and TimeStop instances).

7.4 GetEventID

The GetEventID command is used to request specific log event records by Event ID (from the list returned in the GetEventList Response). The Responder returns the requested log record.

GetEventID Request

Item Name	Type	Len	UL	Meaning
Event ID Request UL Name	Pack Key	16		Identifies the GetEventID Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request
Event ID	Uint32	4		ID of the requested event

- The Event ID identifies a specific log record.

GetEventID Response

Item Name	Type	Len	UL	Meaning
Event ID Response UL Name	Pack Key	16		Identifies the GetEventID Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
Log Record	Text	Var		Record being delivered
Response	Uint8	1		General Response info

- Log Record is the logged information for the specified event. It is a signed or unsigned XML log data record constructed as defined in [Log] and carried as text within the KLV structure.

- The length of the Log Record item can be determined from the Length of the message.

Note: The utility of the GetEventList and GetEventID messages is to provide a method to identify and move security log data from a remote SPB to the Image Media Block (IMB containing the Security Manager). The TLS connection between Initiator (IMB) and Responder (remote SPB) provides a trusted and secure path for the log record "signing proxy" technique as described in [Log]. Provisioning for log record filtering (as described in [Log]) is unaffected by these RRP messages, and outside the scope of this specification.

7.5 QuerySPB

The QuerySPB command is used to interrogate an SPB as to health and status.

QuerySPB Request

Item Name	Type	Len	UL	Meaning
Secure Processing Block Query Request UL Name	Pack Key	16		Identifies the QuerySPB Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request

QuerySPB Response

Item Name	Type	Len	UL	Meaning
Secure Processing Block Query Response UL Name	Pack Key	16		Identifies the QuerySPB Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
Protocol_Ver	Uint8	1		ASM Protocol in Use
Status	Uint8	1		Status information
Response	Uint8	1		General response info

- Protocol_Ver returns a value indicating the ASM protocol suite being used by the Responder. The initial version value of this field shall be 0x01.
- Status items are defined as follows:

Element	Meaning	Value
Not playing	Recipient is not performing a playout security function	0
Playing	Recipient is performing a playout security function	1
Security Alert	Recipient is reporting a security alert condition	2

Note: Examples of security alert conditions would be an open access panel or failed clock.

7.6 GetProjCert

The GetProjCert command returns the projector SPB certificate from the Link Decryptor Block (LDB) over the LDB's TLS connection with the Security Manager. The certificate returned shall be from the projector (SPB) to which the LDB is currently married ("marriage" is defined in [Log]). This command shall fail if the LDB is not in an actively married state.

GetProjCert Request

Item Name	Type	Len	UL	Description
Projector Certificate Request UL Name	Pack Key	16		Identifies the GetProjCert Request
Length	BER Length	4	n/a	Pack Length
Request ID	Uint32	4		ID of this Request

GetProjCert Response

Item Name	Type	Len	UL	Description
Projector Certificate Response UL Name	Pack Key	16		Identifies GetProjCert Response
Length	BER Length	4	n/a	Pack Length
Request ID	Uint32	4		IDs the Request for which this is the Response
Projector Certificate Data	Byte Array	Var		DER encoded Certificate
Response	Uint8	1		General Response Info

Note: The length of the certificate can be determined from the length of the response.

8 Link Encryption ASM Commands

Table 2 lists the ASM commands defined for support of Link Encryption (LE). Table 1 shows these commands together with the names as recorded in the SMPTE UL metadata registries.

Request-Response round trip latency — Per item (2) of Section 6.4, Responder implementations should support a maximum round-trip Request-to-Response latency of 2 seconds for Link Encryption ASM commands.

Responder key buffer — The Link Encryption Responder endpoint shall have a buffer for at least 16 Keys and Key_IDs.

Note*:* Other messages (or message sets) may be defined in the future (e.g., to support keying of remote SPBs).

Table 2 – Link Encryption ASM Command Types

Link Encryption ASM Commands	SMPTE Metadata Link Encryption ASM Command UL Name
LEKeyLoad Request	Link Encryption Key Load Request
LEKeyLoad Response	Link Encryption Key Load Response
LEKeyQueryID Request	Link Encryption Key Query ID Request
LEKeyQueryID Response	Link Encryption Key Query ID Response
LEKeyQueryAll Request	Link Encryption Key Query All Request
LEKeyQueryAll Response	Link Encryption Key Query All Response
LEKeyPurgeID Request	Link Encryption Purge ID Request
LEKeyPurgeID Response	Link Encryption Purge ID Response
LEKeyPurgeAll Request	Link Encryption Purge All Request
LEKeyPurgeAll Response	Link Encryption Purge All Response

8.1 LEKeyLoad

The LEKeyLoad command delivers LE keys to a Link Decryptor Block (LDB). The use of the KLV batch facility enables the command to carry more than one key.

Note: Synchronization of the current decryption key to be used is coordinated through the use of an in-band (i.e., essence-borne) metadata indicator not defined in this document.

LEKeyLoad Request

Item Name	Type	Len	UL	Meaning
Link Encryption Key Load Request UL Name	Pack Key	16		Identifies the LEKeyLoad Request command
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request
LEKeyBatch	LE Key Batch	8+(32*n)		An unordered batch of LE Key ID to Key, Expire Time and Attribute Data mappings (see table below)

LEKeyBatch

Item Name	Type	Len	UL	Meaning	Default
Number of Items	Uint32	4	n/a	The number of Items in the Batch	n
Item Length	Uint32	4	n/a	The length of each Item	32
LE Key ID	Uint32	4		Unique key identifier	
Key	Array of Bytes	16		Decryption key	
Expire Time	Uint32	4		Validity period of key in seconds	
Attribute Data	Uint64	8		Data used as input to counter mode cipher	

- Attribute_Data is used to seed the AES core in counter mode for link decryption. This number shall be a unique random number produced by the SM for each LE key.
- The LEKeyBatch is a batch of LE Key ID to Key, Expire Time and Attribute Data mappings. The quadruplets are in no particular order. There shall be only one entry for each LE Key ID in the LEKeyBatch.
- The length of the variable portion of the LEKeyBatch can be determined by multiplying the Number_of_Items by the Item_length.
- Expire_Time is the valid length of time for key use in seconds, after which the SM should purge the key. Upon receipt of the key, the receiving SPB shall track time in seconds.
- LE_Key_ID is a unique identifier for each individual key. (The LE_Key_ID for LE keys is not a UUID.)
- Key is the 128 bit LE key.

LEKeyLoad Response

Item Name	Type	Len	UL	Meaning
Link Encryption Key Load Response UL Name	Pack Key	16		Identifies the LEKeyLoad Response message
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
Overflow	Uint8	1		Key buffer full
Response	Uint8	1		General Response Info

- A non-zero Overflow value indicates that the LEKeyLoad Request has not been successfully executed because the LDB key buffer would have overflowed. A non-zero value shall also cause the Response element to indicate “failed RRP”.

Note: LDB devices may have different key and key ID memory buffer capacities. The Overflow element allows the Responder to inform the Initiator when attempts to load LE keys would overwrite active keys.

8.2 LEKeyQueryID

The LEKeyQueryID command interrogates the LDB for the presence of a key which is identified by KeyID.

LEKeyQueryID Request

Item Name	Type	Len	UL	Meaning
Link Encryption Key Query ID Request UL Name	Pack Key	16		Identifies the LEKeyQueryID Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs of this Request
LE Key ID	Uint32	4		Unique LE Key ID

- LE Key ID contains the ID of the LE key to be queried.

LEKeyQueryID Response

Item Name	Type	Len	UL	Meaning
Link Encryption Key Query ID Response UL Name	Pack Key	16		Identifies the LEKeyQueryID Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
KeyPresent	Uint8	1		“1” if the key is present, “0” otherwise
Response	Uint8	1		General Response info

- KeyPresent shall have a value of one (1) if the key identified by LE Key ID is present; the value shall otherwise be zero (0).

8.3 LEKeyQueryAll

The LEKeyQueryAll command interrogates the LDB to report all of its active LE keys.

LEKeyQueryAll Request

Item Name	Type	Len	UL	Meaning
Link Encryption Key Query All Request UL Name	Pack Key	16		Identifies the LEKeyQueryAll Request
Length UL Name	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request

LEKeyQueryAll Response

Item Name	Type	Len	UL	Meaning
Link Encryption Key Query All Response UL Name	Pack Key	16		Identifies the LEKeyQueryAll Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
LEKeyIDBatch	LE Key ID Batch	8+4n		An unordered batch of LE Key ID values (see table below)
Response	Uint8	1		General Response info

LEKeyID Batch

Item Name	Type	Len	UL	Meaning	Default
Number of Items	Uint32	4	n/a	The number of Items in the Batch	n
Item Length	Uint32	4	n/a	The length of each Item	4
LE Key ID	Uint32	4		Unique key identifier(s)	

- LEKeyID Batch returns all the active LE Key IDs within the LDB. A Number_of_items of zero in the LEKeyID Batch shall mean no LE keys are present.
- The length of the variable portion of the LEKeyID Batch can be determined by multiplying the Number of items by the Item length.
- The LEKeyID Batch is a batch of LE Key ID values. The LE Key IDs are in no particular order. There shall be only one entry for each LE Key ID in the LEKeyID Batch.

8.4 LEKeyPurgeID

LEKeyPurgeID commands an LDB to purge (zeroize) the key identified by LE Key ID.

LEKeyPurgeID Request

Item Name	Type	Len	UL	Meaning
Link Encryption Purge ID Request UL Name	Pack Key	16		Identifies the LEKeyPurgeID Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request
LE Key ID	Uint32	4		Unique Key ID

- LE Key ID contains the ID of the LE key to be purged.

LEKeyPurgeID Response

Item Name	Type	Len	UL	Meaning
Link Encryption Purge ID Response UL Name	Pack Key	16		Identifies the LEKeyPurgeID Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
NoKeyID	Uint8	1		Key ID not present
Response	Uint8	1		General Response info

- A non-zero value for NoKeyID indicates that the identified key to be purged was not present. The value of Response for this condition is “RRP Successful”.

8.5 LEKeyPurgeAll

LEKeyPurgeAll commands an LDB to purge (zeroize) all its LE keys.

LEKeyPurgeAll Request

Item Name	Type	Len	UL	Meaning
Link Encryption Purge All Request UL Name	Pack Key	16		Identifies the LEKeyPurgeAll Request
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		ID of this Request

LEKeyPurgeAll Response

Item Name	Type	Len	UL	Meaning
Link Encryption Purge All Response UL Name	Pack Key	16		Identifies the LEKeyPurgeAll Response
Length	BER Length	4	n/a	Pack length
Request ID	Uint32	4		IDs the Request for which this is the Response
Response	Uint8	1		General Response info

Annex A (Normative)
Auditorium Security Messages Variable Length Universal Label (UL) Key

As a Fixed Length Pack (group of KLV elements), each individual item in the Value field comprises only an item value. All items in a pack are required.

Table A.1 – Common UL Key Value for all ASM Commands

Byte No.	Description	Value (hex)	Meaning
1	Object Identifier	06h	Object ID
2	Label size	0Eh	Length of UL
3	Designator	2Bh	Sub Identifier
4	Designator	34h	SMPTE Identifier
5	Registry Category Designator	02h	KLV Groups (Sets and Packs)
6	Registry Designator	05h	Fixed Length Pack
7	Structure Designator	01h	Groups Dictionary
8	Version Number	01h	Registry Version: Dictionary version 1
9	Item Designator	02h	Administration
10	Organization	07h	Access Control
11	Application	01h	Auditorium Security
12	Set/Pack Kind (1)	xx	Command Type (see Annex Table A.2)
13	Set/Pack Kind (2)	yy	Command Type (see Annex Table A.2)
14	Reserved	00h	Not assigned
15	Reserved	00h	Not assigned
16	Reserved	00h	Not assigned

The values of bytes 12 and 13 for specified ASM command types are given in Table A.2.

Table A.2 – Key Values for Command Types

Message Category and byte 12 node names	SMPTE UL Name Command Type and byte 13 node names	Byte 12	Byte 13
General Purpose Logs	Bad Request Response	01h	01h
Temporal Indicators	Time Request	02h	10h
Temporal Indicators	Time Response	02h	11h
Temporal Indicators	Event List Request	02h	12h
Temporal Indicators	Event List Response	02h	13h
Temporal Indicators	Event ID Request	02h	14h
Temporal Indicators	Event ID Response	02h	15h
Temporal Indicators	Secure Processing Block Query Request	02h	16h
Temporal Indicators	Secure Processing Block Query Response	02h	17h
Temporal Indicators	Projector Certificate Request	02h	18h
Temporal Indicators	Projector Certificate Response	02h	19h
Link Encryption	Link Encryption Key Load Request	03h	20h
Link Encryption	Link Encryption Key Load Response	03h	21h
Link Encryption	Link Encryption Key Query ID Request	03h	22h
Link Encryption	Link Encryption Key Query ID Response	03h	23h
Link Encryption	Link Encryption Key Query All Request	03h	24h
Link Encryption	Link Encryption Key Query All Response	03h	25h
Link Encryption	Link Encryption Purge ID Request	03h	26h
Link Encryption	Link Encryption Purge ID Response	03h	27h
Link Encryption	Link Encryption Purge All Request	03h	28h
Link Encryption	Link Encryption Purge All Response	03h	29h

Annex B (Informative)
Bibliography

This section contains informative references that provide helpful background information.

[FIPS-140-2] “Security Requirements for Cryptographic Modules” Version 2, May 25, 2001. FIPS-140-2 http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[PKCS] “Public Key Cryptography Standards (PKCS) #1” RFC 3447 See www.ietf.org/rfc/rfc3447.txt

[Rescorla] Eric Rescorla. SSL and TLS: Designing and Building Secure Systems. Addison Wesley Professional. ISBN 0201615983. October 2000

[CPL] SMPTE 429-7-2006, D-Cinema Packaging — Composition Playlist

[LogRecord] SMPTE 430-4-2008, Log Record Format Specification for D-Cinema

SMPTE 377-1-2009, Material Exchange Format (MXF) — File Format Specification

Annex C (Informative)
Explanation of TLS Length Constraints

The fragment length restriction is on the maximum ciphertext length, not the plaintext length. When the initial TLS handshake occurs, because encryption hasn't yet been established, the TLSPlaintext.length ([TLS] Sect. 6.2.1.) value is equal to the TLSCiphertext.length value (this can be confirmed by noting that the X.509 certificates are split into n-byte chunks during the initial handshake, where n is the chosen fragment size.)

Once encryption has been enabled, the length of the plaintext data in each TLS fragment must be reduced to allow for the encryption envelope. Because this standard (SMPTE ST 430-6) constrains both the compression method to one value (null) and encryption method to one value (AES-CBC 128-bit), the corresponding effective maximum TLSPlaintext.length is constant: starting with the maximum TLSCiphertext.length value of 512 bytes, we subtract the length of the padding_length value (1 byte). Assuming a padding value of zero (allowing the maximum plaintext payload), we then subtract an additional 20 bytes for the message digest (SHA-1, per [TLS-AES]), leaving 491 bytes for plaintext payload (512 - 1 - 20 = 491).