SMPTE 427-2009

SMPTE STANDARD

Link Encryption for 1.5 Gb/s[1] Serial Digital Interface

[1] Nominal total Bit Rate

Table of Contents

3 Barker Avenue, White Plains, NY 10601
(914) 761-1100

Approved
March 11, 2009

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 Standard 427 was prepared by Technology Committee DC28.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards and Engineering at SMPTE Headquarters.

Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

1 Scope

This Standard defines a method for providing secure transmission of digital pictures over a transport conforming to SMPTE 292. Encryption of data in H-ANC and V-ANC data regions is not defined by this standard This document also defines the Link Encryption metadata to synchronize the encryption and decryption processes, and a Link Encryption Key Message to carry Link Encryption keys for decryption over the 1.5Gb/s interface

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.

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.

SMPTE 291M-2006, Television — Ancillary Data Packet and Space Formatting

SMPTE 292-2008, 1.5Gb/s Signal/Data Serial Interface

SMPTE 372-2009, Dual Link 1.5 Gb/s Digital Interface for 1920 × 1080 and 2048 × 1080 Picture Formats

SMPTE 425-2008, 3 Gb/s Signal/Data Serial Interface — Source Image Format Mapping

AES, FIPS PUB 197, Advanced Encryption Standard. U.S. Department of Commerce/National Institute of Standards and Technology. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

SHA1, FIPS PUB 180-1. Secure Hash Standard. U.S. Department of Commerce/National Institute of Standards and Technology. http://csrc.nist.gov/publications/fips/fips180-1/fip180-1.txt

PKCS1, RFC 2437: PKCS #1: RSA Cryptography Specifications Version 2.0. B. Kaliski and J. Staddon. Informational, October 1998. http://www.ietf.org/rfc/rfc2437.txt

UTC, RFC 3339: Date and Time on the Internet: Timestamps. G. Klyne and C. Newman. Informational, July 2002. http://ietf.org/rfc/rfc3339.txt

4 Introduction

An informative example of the Encryption and Decryption block diagram is shown in Figure 1. One such combination enables encryption and decryption for Y' and C'_B/C'_R 10-bit signals through the interface defined by SMPTE 292. Two combinations enable encryption and decryption for R'G'B', *X'Y'Z'* and other 10/12-bit signals through the Dual Link specified in SMPTE 372 or a single 3Gb/s link conforming to SMPTE 425. These signals can be handled as dual Y' and C'_B/C'_R 10-bit system in terms of the encryption and decryption. (See Notes.)

The Encryption engine encrypts the plaintext. The encrypted data being transmitted shall not contain the reserved prohibited codes 000h (0) through 003h (3) and 3FCh (1020) through 3FFh (1023). In the encryption process, the LE_Key, Timing and Metadata generator generates the metadata and maps them into the Y' channel vertical ancillary data area of the serial interface. In the decryption process the LE_Key, Timing and Metadata demultiplexer detect the metadata and provide the Decryption engine with the same Link Encryption key, AES input and frame reset trigger to reproduce the original plaintext. Encryption is applied to all the active picture area. An informative block diagram of the encryption and decryption processes is shown in Figure 2.

Notes:

1. For convenience, throughout this document the encryption, decryption, and other functions of the equipment noted as 'Example' may be described in terms of a particular implementation method.

2. Link latency may vary upon different implementations

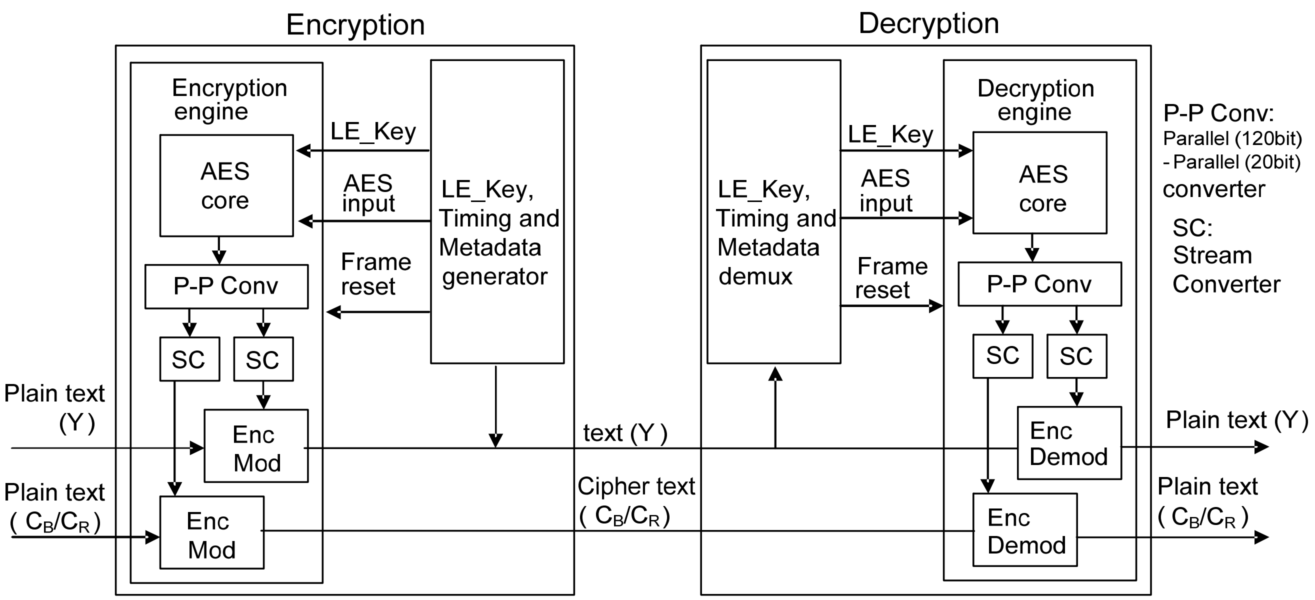
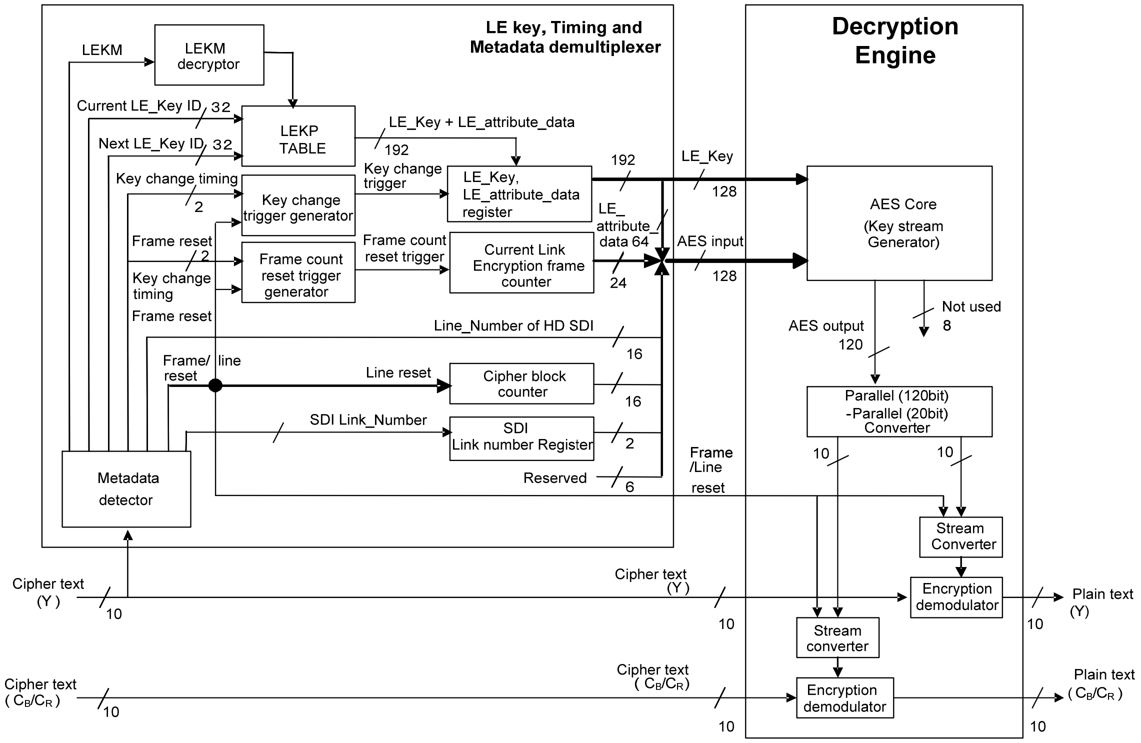


Figure 1 – Encryption and Decryption (Example)



Note: LEKP 192bit consists of LE Key 128bit and LE_attribute_data 64 bit

Figure 2 – Decryption process block diagram (Example)

5 Encryption/Decryption Engine

The Encryption/Decryption engine shall consist of a key stream generator, parallel (120-bit)-to-parallel (20-bit) converter, stream converters and Encryption modulators or Encryption demodulators. An example of the Encryption engine for Y’ and C'_B/C'_R for the 1.5 Gb/s interface is shown in Figures 1 and 2.

5.1 AES Core and Parallel (120-bit)-parallel (20-bit) Converter

The AES counter mode (128-bit) specified in FIPS PUB 197 shall be used for the key stream generation. The AES core shall generate the 128-bit pseudo-random numbers at one sixth of the clock frequency (74.25/6 MHz) of plaintext (or ciphertext). The parallel (120-bit)-to-parallel (20-bit) converter connected to the output of the AES core shall convert the lower 120 bits of the 128 bit AES key stream at 74.25/6 MHz clock frequency to 2 x 10-bit key streams at 74.25 MHz clock frequency. The bit relationship between the AES core output and the plaintext (or ciphertext) shall be as shown in Figure 3.

Clock rates scaled by 1/1.001 are also applied in the case of 60/P, 60/I, 30/P, 24/P, 30/PsF, and 24/PsF.

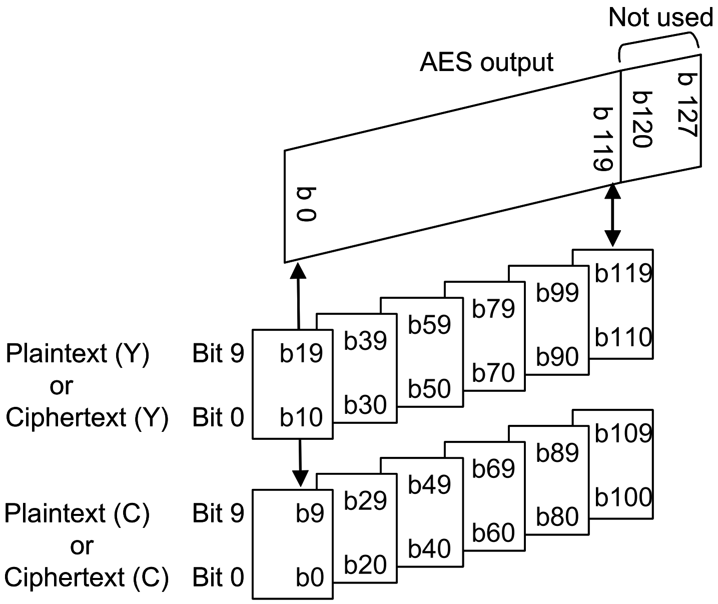


Figure 3 – Bit relationship between AES core output and plaintext (or ciphertext)

5.2 Stream Converter

The stream converter shall select 000h-3F7h (0-1015) from the 000h–3FFh (0–1023) AES key stream and shall provide the Encryption modulator (or Encryption demodulator) with 000h-3F7h (0-1015) pseudo-random numbers maintaining randomness. An example of the stream converter is shown in Figure 4. The stream converter shall have a 64-stage x 10-bit FIFO and may have a FIFO controller. The stream converter shall have a filling period and an active picture period. The stream converter shall be reset every line. The detailed timing of the filling period, active picture period and Stream converter reset shall be as specified in § 8.2.

5.2.1 Filling period

During the horizontal filling period, and when AES key stream detects 000h–3F7h (0-1015), the FIFO controller sends an input enable signal to the FIFO until all the 64 stages of FIFO are filled. After all the 64 stages are filled, the FIFO controller stops sending the input enable signal. The FIFO controller does not send any output enable signal to the FIFO during this period. The estimated probability of the FIFO being unfilled is calculated in Annex A.

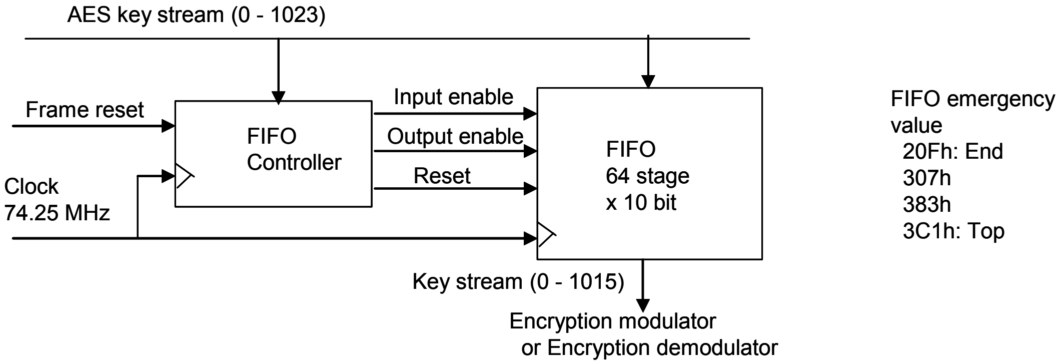


Figure 4 – Stream converter block diagram (Example)

5.2.2 Active picture period

During the active picture period (samples 0-1919) and when AES key stream detects 000h–3F7h (0-1015), the FIFO controller sends the input enable signal to the FIFO. At the same time the FIFO controller sends the output enable signal to the FIFO, and the FIFO shall provide the Encryption modulator (or Encryption demodulator) with its stored data.

Every time the AES key stream detects 3F8h–3FFh (1016-1023) during this period, one stage of the FIFO becomes empty due to the lack of filling. Since 0.8 percent average of the AES key stream takes 3F8h-3FFh (1016-1023), the average decrease of FIFO stored data during this period is 15 stages. See Annex A for the estimated probability of the FIFO being empty.

Should the FIFO become empty during the active picture period, the FIFO emergency values (3C1h, 383h, 307h, 20Fh) shall be cyclically used instead of the FIFO output. When the Stream Converter is reset, the FIFO emergency values shall be set as the initial values.

5. 3 Encryption Modulator

The encryption modulator shall encrypt the plaintext of the 10-bit Y’ channel and the 10-bit C'_B/C'_R channel at 74.25 MHz with two key streams of 004h-3FBh (4-1019), and shall generate the ciphertext of the 10-bit Y’ channel and the 10-bit C'_B/C'_R channel consisting of 004h-3FBh (4-1019) data. The calculation of the Encryption modulator and the Encryption demodulator shall be the addition, subtraction and modulo process as shown in equations 1 and 2.

Encryption: $Ci = [(Mi – N1) + Ei] \bmod (N2) + N1$ (1)

Decryption: $Mi = [(Ci – N1) – Ei] \bmod (N2) + N1$ (2)

Where,

Mi: Input data of the encryption modulator and the output data of the encryption demodulator,

Ci: Encrypted data,

Ei: Pseudo random numbers from the stream converter,

N1: Number of the prohibited codes in the lower area of the picture data. N1 equals 4 in 10-bit system including 000h, 001h, 002h and 003h .

N2: Number of the allowed codes of the picture data. N2 in 10 bit system equals 1016, excluding 8 prohibited codes, 000h through 003h (0-3) and 3FCh through 3FFh (1020-1023).

Note: Operator 'a mod b' generates the remainder of 'a' divided by 'b'. For example, 1020 mod 1016 equals 4. If 'a' is a minus number, then 'b' is added to 'a' until 'a + b' becomes a positive number. For example if 'a' is –5, then -5 mod 1016 equals 1011. And if 'a' is –2000 then -2000 mod 1016 equals 32.

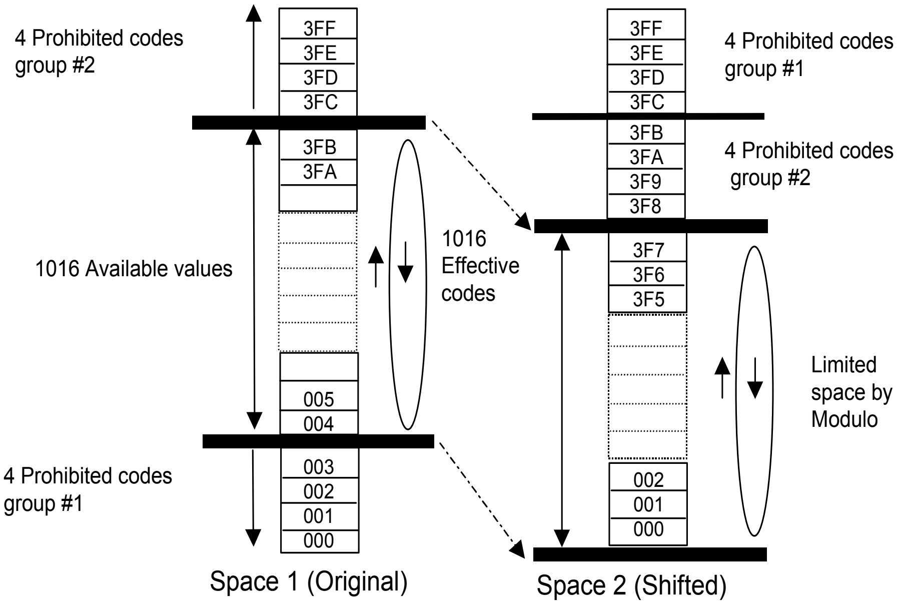


Figure 5 – Effective codes and prohibited codes

Figure 5 shows the encryption modulator code values permitted on the 1.5 Gb/s interface. Space 1 of Figure 5 shows the allocation of the prohibited codes 000h (0) through 003h (3) and 3FCh (1020) through 3FFh (1023) and available values 004h (4) through 3FBh (1019). In the equations 1 and 2, N1 equals 4 and N2 equals 1016. The encryption modulator calculates as follows:

a) Plaintext data (input picture data) Mi shall take available values 004h (4) through 3FBh (1019) in space 1. N1 (=4) shall be subtracted from Mi, then the available values shall be shifted to effective codes 000h (0) through 3F7h (1015) in space 2.

b) A pseudo-random number Ei between 000h (0) and 3F7h (1015) shall be added to the shifted available values.

c) [(Mi – 4) + Ei] modulo 1016 limits the output values from 000h (0) through 3F7h (1015). And after N1 (=4) is added, encrypted values take numbers from 004h (4) through 3FBh (1019) which do not take any prohibited codes.

5.4 Encryption Demodulator

The process of the encryption demodulator shall be the reverse calculation of the encryption modulator.

a) Encrypted data Ci shall take available values 004h (4) through 3FBh (1019) in space 1, N1 (= 4) shall be subtracted from Ci, and the available values shall be shifted to space 2 which has values between 000h (0) and 3F7h (1015).

b) A pseudo-random number Ei which has the same code of the encryption modulator shall be subtracted from the shifted available values.

[(Ci – 4) - Ei] modulo 1016 limits the output values from 000h (0) through 3F7h (1015). After adding N1 (=4), calculated values take numbers from 004h (4) through 3FBh (1019) which equal the original plaintext data due to the pseudo-random numbers being the same.

6 Link Encryption Key (LE_Key)

6.1 LE_Key Generation

The LE_Key is a 128-bit integer that shall be chosen randomly. A different series of pseudo-random numbers shall be used for each exhibition playout. The total number of the LE_Keys may be selected between 1 and 2^{32}-1. The LE_Key ID shall be used to identify each LE_Key. (see Table 1 and Note.)

6.2 LE_Key Distribution

The same LE_Key shall be provided to the encryption and decryption processes. All LE_Keys may be sent simultaneously prior to the beginning of a program, or each necessary LE_Key can be sent prior to program segments. The LE_Key shall be provided prior to the synchronization at the key change.

The LE_Key may be distributed through any distribution method supported by the equipment, such as the 1.5 Gb/s serial link, Ethernet, or solid state memory. An LE_Key shall be distributed as a Link Encryption Key Message (LEKM) which shall be protected by a secure distribution system, using RSA encryption method. Every LE_Key shall be accompanied by a series of LE_Key ID's specified in this standard. The LE_Key ID shall be distributed through a 1.5 Gb/s serial link as a part of Link Encryption metadata. Compliant receivers shall be capable of updating keying material at any picture frame boundary.

Details of the LEKM distribution are defined in § 7. Other distribution methods are outside the scope of this standard.

Note: Key updating may be initiated on a regular or random schedule, as appropriate to a particular application.

6.3 LE_Key Change Timing

A new LE_Key shall be changed at the point L_{AES} prior to the first word of EAV (End of Active Video) of the first active picture line which shall be the same as AES input reset timing. The LE_Key shall not be changed during the active picture period. The key change timing command shall generate the key change trigger to switch the current LE_Key to the next LE_Key.

The LE_Key ID shall identify the LE_Key, for example, as shown in Table 1. The LE_Key ID shall be handled as Link Encryption metadata

When the LE_Key itself is encrypted, it shall be decrypted before it is used for picture data encryption and decryption. The relationship among the (a) Key change timing command, (b) key change trigger, (c) LE_Key ID and (d) LE_Key shall be as shown in Figure 6. The detail of the Link Encryption metadata structure is specified in Table 10 of § 9.

Table 1 – (Example) LE_Key ID and LE_Key

LE_Key ID	LE_Key
0	No encryption
1	Key 1
2	Key 2
–	–
n	Key n

Note: n = 0, 1, 2, n (max 2^{32}–1). Key 0, key 1, key 2, key n are numbers between 0 and (2^{128}–1).

The transmitting equipment shall provide a unique LE_Key ID for each LE_Key.

LE_Key ID zero (0) has a special meaning, indicating that the program is not encrypted. Although the example shows LE_Key ID’s to be sequential numbers, there is no such requirement implied, and compliant equipment shall respond correctly to any valid LE_Key ID at any time.

LE_Key ID's one through four (1-4) have nonvolatile storage requirements specified in § 7.1. These LE_Key ID’s shall be used at the system start before decrypting LEKM.

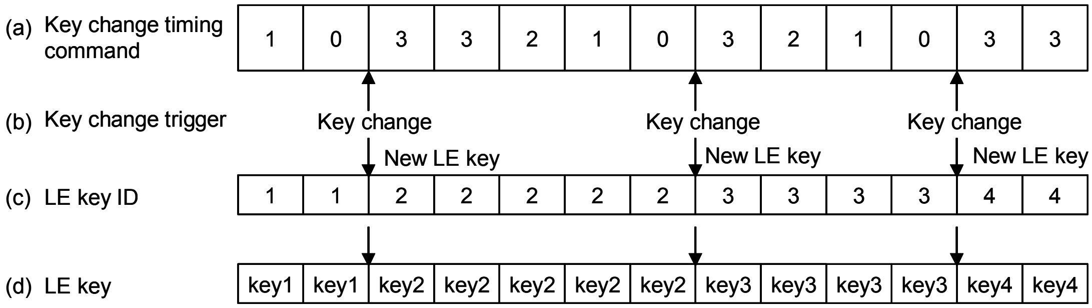


Figure 6 – Relationship among the Key change timing command, Key change trigger, LE_Key ID and LE_Key

7 Link Encryption Key Message (LEKM) and Link Encryption Key Payload (LEKP)

7.1 Overview

The Link Encryption Key Message (LEKM) enables the secure transmission of link encryption key information through an un-secured uni-directional 1.5Gb/s serial interface.

The LEKM consists of a Link Encryption Key Payload (LEKP) encrypted using a public key unique to the recipient device. The LEKP may alternatively be transmitted by means other than wrapping in a LEKM.

In order to prevent the contents from theft and illegal playback, the LEKP carries, in addition to the LE_Key, a timestamp. The latter prevents replay attacks and requires both source and recipient devices to have access to trusted, reliable and synchronized time sources.

An LE_Key expires when (1) the current time exceeds its timestamp or (2) its key is replaced by a new key, per § 6.3. The recipient device shall not use an expired LE_Key under any circumstances. The recipient device shall provide nonvolatile storage for four (4) LEKPs, and in the case of equipment shutdown LEKPs for LE_Key ID's 1 through 4 shall be retained if they are unexpired at the time of shutdown.

Source and recipient devices shall support the cryptographic algorithms specified in this Standard. This Standard does not specify the method by which the public key of the recipient device is communicated to the source device. Furthermore, this Standard does not preclude the transmission of the link encryption key and its associated ID over alternate channels. The LEKP may for instance be transmitted over a local area network. In such cases, receiving equipment shall respond to the contents of the LEKP in the same manner as specified in this Standard for LEKP delivered via LEKM.

7.2 Maximum Recipient Processing Delay for LEKM Updates

LEKM processing involves public-key computations of considerable complexity, which may cause significant delay between delivery of LEKM and the availability of the enclosed LE_Key for use in the AES block. In order to ensure interoperability, this clause specifies a delay and buffer model as a minimum performance level for recipient equipment as shown in Figure 7. With the exception of the LEKM processing delay, all other operations in the buffer model are assumed to be instantaneous.

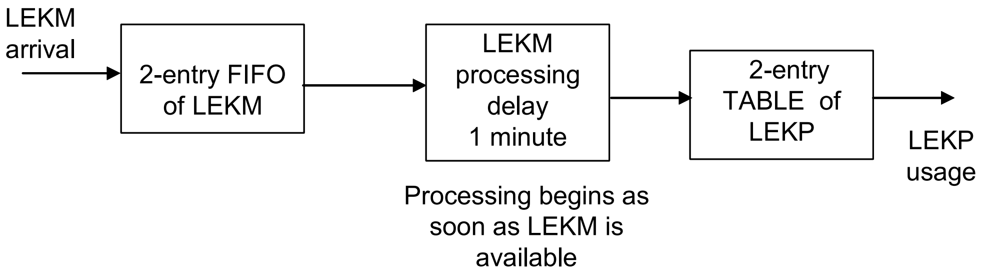

LEKM arrival
2-entry FIFO of LEKM
LEKM processing delay 1 minute
2-entry TABLE of LEKP
LEKP usage
Processing begins as soon as LEKM is available

Figure 7 – LEKM processing – Buffer model

When an incoming LEKM with the same LE_Key ID and SHA1_digest are received at the input FIFO of LEKM, it shall be ignored (see Note 2). When a processed LEKM, i.e. LEKP, has the same LE_Key ID in the output TABLE of LEKP, the new LEKP shall replace the existing LEKP. The minimum duration of the LEKM shall be one minute as shown in Figure 7.

Notes:

1. This buffer model is specified as the minimum required to achieve satisfactory performance. Compliant equipment may process the LEKM decryption faster, and may have larger FIFO storage.

2. Ignoring the LEKM with the same LE_Key ID allows LEKMs to be repeated in continuous plural frames.

7.3 Link Encryption Key Message Structure (LEKM)

Table 2 details the structure of LEKM. The parameters shall be aligned on octet boundaries and shall be packed in the order in which they appear. Integers shall be stored LSB first.

The LSB of Algorithm_type shall be mapped into bit 0, word 1 of ANC packet 1 in Table 7.

Table 2 – Link Encryption Key Message (LEKM) structure

Parameter	Description	Size (octets)
Algorithm_type	Algorithm used to encrypt the LEKP	1
LE_Key ID	ID of the link encryption key carried in LEKM	4
SHA1_digest	Unique identifier for the key used in encrypting the LEKP	8
LEKP_len	Length (in octets) of the LEKP	2
ELEKP_len	Length (in octets) of the encrypted LEKP	2
ELEKP_data	Encrypted LEKP	ELEKP_len

LE_Key ID field is redundant to identical information which is securely embedded in the LEKP. Duplication of the information assists resource-constrained receiving devices which need to detect keys during playout.

7.3.1 Algorithm_type

Size: 1 octet
Type: Unsigned integer
Valid range: [0] (see Table 3), [1..255] reserved

The Algorithm_type parameter shall identify the cryptographic algorithm used to encrypt the LEKP, as listed in Table 3. The algorithm shall provide integrity checks to ensure that random LEKP are not decrypted and used by the recipient.

Table 3 – LEKM algorithm valid value

Algorithm_type	name	Description
0	RSA_oaep_mgf1p_sha1_2048	2048-bit RSA encryption using SHA1 as the message digest algorithm(RFC2437).

7.3.2 LE_Key ID

Size: 4 octets
Type: Unsigned integer

The LE_Key ID parameter shall identify the LE_Key contained in the message, per '§ 6.3. The LE_Key ID is not encrypted to allow immediate identification of the change of LE_Key ID.

7.3.3 SHA1_digest

Size: 8 octets
Type: Unsigned integer

The SHA1_digest shall consist of the lower 64 bits of the SHA-1 digest [SHA1] of the LEKM, which is same as the encrypted LEKP per FIPS PUB 197. The combination of the SHA1_digest per FIPS PUB 180-1 and Algorithm_type parameters shall allow the recipient to uniquely determine the required decryption key.

7.3.4 LEKP_len

Size: 2 octets
Type: Unsigned integer
Valid range: [0..2^{16}-1]

The LEKP_len parameter shall contain the length in octets of the unencrypted LEKP.

7.3.5 ELEKP_len

Size: 2 octets
Type: Unsigned integer
Valid range: [0..2^{16}-1]

The ELEKP_len parameter shall contain the length in octets of the ELEKP_data parameter. The ELEKP_data parameter for the Algorithm_type 0 in Table 3 is 2048 bits.

7.3.6 ELEKP_data

Size: ELEKP_len octets
Type: Octet string

The ELEKP_data parameter shall contain the encrypted LEKP.

7.4 Link Encryption Key Payload (LEKP) Structure

Table 4 defines the structure of LEKP. The parameters shall be aligned on octet boundaries and packed in the order in which they appear. Unless stated otherwise, integers shall be stored LSB first.

Table 4 – Link Encryption Key Payload (LEKP) structure

Parameter	Short Description	Size
Not_valid_after	Expiration date of the LEKM	7
LE_attribute_len	Length in octets of the LE attribute data	1
LE_attribute_data	Attribute data, such a nonce, isolated from the LE_Key	LE_attribute_len
LE_Key ID	ID of the link encryption key carried in the LEKP	4
LE_Key_type	Cryptographic method to be used along with the link encryption key	1
LE_Key_len	Length in octets of the link encryption key	1
LE_Key	Link encryption key	LE_Key_len

7.4.1 Not_valid_after

Size: 7 octets
Type: Octet string

The Not_valid_after parameter determines the time after which the LEKM expires. The recipient device shall not use, and the source device shall not transmit, an expired LEKM. The time shall be in the form of a Universal Time Coordinated (UTC) timestamp per RFC 3339.

Table 5 details the structure of Not_valid_after parameter. The fields are aligned on octet boundaries and shall be packed in the order in which they appear.

Table 5 – Not-Valid-After structure

Fields	Type	Size (octets)	Valid Range
Date_fullyear	Unsigned Integer	2	[0..2^{16}-1]
Date_month	Unsigned Integer	1	[1..12]
Date_mday(=day)	Unsigned Integer	1	[1..31]
Time_hour	Unsigned Integer	1	[0..23]
Time_minute	Unsigned Integer	1	[0..59]
Time_second	Unsigned Integer	1	[0..60]

7.4.2 LE_attribute_len

Size: 1 octets
Type: Unsigned integer

The LE_attribute_len parameter shall contain the length in octets of the LE_attribute_data parameter. LE_attribute_len for LE_Key_type 0 is 64-bit.

7.4.3 LE_attribute_data

Size: LE_attribute_len octets
Type: Octet string

The LE_attribute_data is used as the input to the AES cipher engine as specified in § 8.1. It is recommended to use random data which changes corresponding to the LE_Key change for increasing the entropy of the AES encryption.

7.4.4 LE_Key ID

Size: 4 octets
Type: Unsigned integer

The LE_Key ID parameter shall identify the key contained in the message, per § 6.3.

7.4.5 **LE_Key_type** (Dynamic)

Size: 1 octets
Type: [0] see Table 6, [1..127], [128..255] reserved

The LE_Key_type parameter shall specify the cryptographic method to be used in conjunction with the LE_Key. The cryptographic method comprises all components of cryptoprocessing (e.g. block algorithm, operating mode, stream converter, encryption modulator, etc.) Tables 6a and 6b list the defined values of this parameter. While the intrinsic length of the LE_Key should be inferred through this parameter, the length of the LE_Key as contained in the LEKP should be determined using the LE_Key_len. Compliant equipment shall not use the LE_Key nor the LE_attribute_data in any method other than that specified by this field.

Table 6a – LE_Key_type valid values

LE_key_type	Short Description	Intrinsic length (octets)
0	128-bit key for AES, applied as defined in § 4 and § 5 of this standard	1

Table 6b – LE_Key_type valid values

LE_Key_type	Short Description	Intrinsic length (octets)
[128..255]	Reserved- not part of this standard	1

7.4.6 LE_Key_len

Size: 1 octets
Type: [0..255]

The LE_Key_len parameter shall contain the length in octets of the LE_Key parameter.

7.4.7 LE_Key

Size: LE_Key_len octets
Type: Octet string

The LE_Key parameter shall contain the LE_Key. If the intrinsic length of the LE_Key is not an octet-multiple, then zero-bit padding shall be inserted.

7.5 LEKM distribution through 1.5Gb/s Serial Interface(s)

The LEKM shall be mapped into the Vertical Ancillary data area of the Y' channel of the 1.5 Gb/s interface. Data structure of the LEKM shall be as shown below.

The maximum size of the LEKM shall be 4080 bits. The 4080 bits shall be divided into 8 bit x 510 words and mapped into the user area of the two ANC packets from the LSB to MSB, from word 1 to word 255 and from ANC packet 1 to ANC packet 2.

The size of the LEKM for Algorithm_type 0 in Table 3 is 273 octets and shall be mapped into the user data area of ANC packets 1 and 2 as shown in Table 7.

Table 7 – LEKM (2184 bit) mapped on the user area of 2 ANC packets

(a) User data area of ANC packet 1

Bit number	Word 1	Words 2 - 5	Words 6 - 13	Words 14 - 15	Words 16 - 17	Words 18 - 255
Bit 9 (MSB)	Complement of bit 8					
Bit 8	Even parity for bit 7 through bit 0					
Bit 7	(bit 7)	(bit 31)	(bit 63)	(bit 15)	(bit 15)	(bit 1903)
Bit 6						
Bit 5						
Bit 4	Algorithm_type	LE_Key ID	SHA1_Digest	LEKP_len	ELEKP_len	ELEKP_data
Bit 3						
Bit 2						
Bit 1						
Bit 0 (LSB)	(bit 0)	(bit 0)	(bit 0)	(bit 0)	(bit 0)	(bit 0)

(b) User data area of ANC packet 2

Bit number	Words 1-18
Bit9 (MSB)	Complement of bit 8
Bit 8	Even parity for bit 7 through bit 0
Bit 7	(bit 2047)
Bit 6	
Bit 5	
Bit 4	ELEKP_data
Bit 3	
Bit 2	
Bit 1	
Bit 0 (LSB)	(bit 1904)

The Ancillary data packet of the LEKM shall adopt the Type 2 data identification of SMPTE 291M, having first a Data Identification (DID) word followed by a Secondary Data Identification (SDID) word. The DID word shall be set to the value ‘40h’. The SDID word shall be set to the value of ‘04h’ for ANC packet 1 and ‘05h’ for ANC packet 2. Table 8 outlines the ancillary data packet words and values.

Table 8 – Ancillary Data Packet Structure of the LEKM

Name	Acronym	Value
Ancillary Data Flag	ADF	000h,3FFh,3FFh
Data Identification	DID	40h
Secondary Data Identification	SDID	04h
Data Count	DC	FFh
User data (LEKM)	UDW	255 words
Check Sum	CS	CS

(a) ANC packet 1

Value
000h,3FFh,3FFh
40h
05h
12h
18 words
CS

(b) ANC packet 2

8 AES Input

8.1 AES Input Items

The AES input (128-bit) shall consist of a 2-bit Link_Number, 6-bits Reserved 64-bit LE_attribute_data, 24-bit Current Link Encryption frame count, 16-bit Line_Number and 16-bit Cipher block count as shown in Figure 2 and Table 9. The AES Input shall be loaded into the AES core most-significant byte (bit 127 through bit 120) first. The LSB of the 128 bits of AES input shall correspond to bit 0 of the Cipher block count. The MSB of the 128 bits of AES input shall correspond to bit 1 of the Link_Number. Other bits of each AES item shall be allocated on the 128 bits of AES input from LSB to MSB order. The default value of the 6-bit Reserved number shall be all zero. Equipment shall not use duplicate line numbers within a single frame.

Note: During each epoch between key changes, the definition in this clause ensures that every cipherblock is derived from a unique 128-bit AES input block, which is essential to system security. The sending equipment should be designed to take precautions to preclude reuse of keys and Le_attribute_data in a manner which allows the same 128-bit AES input block to occur twice under the same LE_Key.

Table 9 – AES input item, description, size and bit allocation on the 128 bit of AES input

AES input item	Description	Size of each item	Bit allocation of each AES input item	Bit allocation of AES input (128-bit)
Link_Number	0: Single link or Link-A of Dual Link, 1: Link-B of Dual Link. 2 -3: Reserved.	2 bit	Bit 1	Bit 127
Reserved	Default 0.	6 bit	Bit 5	Bit 125
LE_attribute_data	The LE_attribute_data is used as the input to the AES cipher engine.	64 bit	Bit 63	Bit 119
Current Link Encryption frame count	Number of frames from the previous key change. The number is set to 0 at each key change and increments every frame thereafter.	24 bit	Bit 23	Bit 55
Line_Number	Line number defined in SMPTE 292M.	16 bit	Bit 15	Bit 31
Cipher block count	Number of the cipher block count within a single line. The number is set to zero for the first cipher block of each line and increments for every block thereafter.	16 bit	Bit 15	Bit 15
			Bit 0	Bit 0

8.2 AES Input and Stream Converter Change Timing

The same AES input shall be provided to the Encryption and Decryption processors. Change timing for each AES input items in Table 9 shall be as shown below.

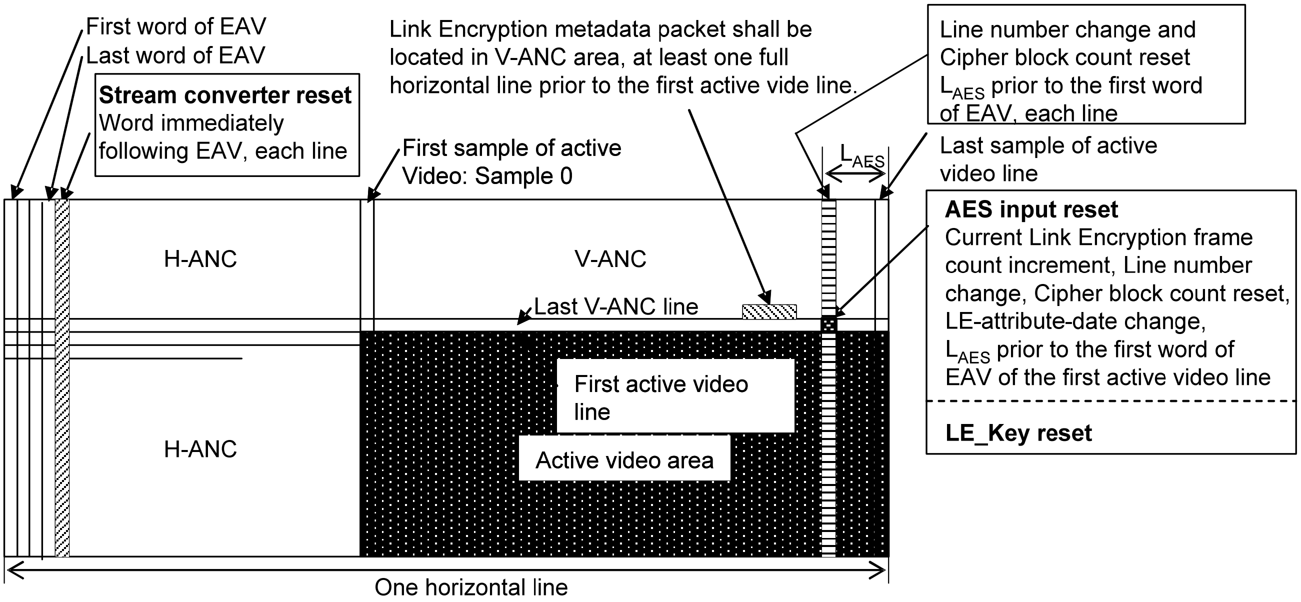
8.2.1 Change timing for the first active picture line

The stream converter shall be reset at the word immediately following EAV of the first active picture line. The Current Link Encryption frame count shall increment its output value at the point L_{AES} prior to the first word of EAV of the first active picture line considering the different AES core latency. Cipher block count shall be reset at the same point. The detailed timing of the Stream converter reset, Current Link Encryption frame count increment, Line_Number change and Cipher block count reset shall be as shown in Figures 8 and 9a.

8.2.2 Change timing for other active picture lines

The Stream converter shall be reset at the word immediately following EAV, each line. Line_Number shall change at the point L_{AES} prior to the first word of EAV of each line except for the first active picture line. Cipher block count shall be reset at the same point. See Figures 8 and 9b.

A different AES core has different latency between AES input and AES output. Assuming this latency as L_{AES}, Current Link Encryption frame count, Line_Number and Cipher block count shall be reset at the point L_{AES} prior to the first word of EAV to realize convertibility among different AES cores.


First word of EAV
Last word of EAV
Stream converter reset
Word immediately following EAV, each line
Link Encryption metadata packet shall be located in V-ANC area, at least one full horizontal line prior to the first active vide line.
Line number change and Cipher block count reset L_{AES} prior to the first word of EAV, each line
First sample of active Video: Sample 0
L_{AES}
Last sample of active video line
AES input reset
Current Link Encryption frame count increment, Line number change, Cipher block count reset, LE-attribute-date change, L_{AES} prior to the first word of EAV of the first active video line
LE_Key reset
H-ANC
V-ANC
Last V-ANC line
First active video line
Active video area
H-ANC
One horizontal line

AES input reset, LE_Key change and Stream converter reset referring to EAV
(see the detail in Figures 8 and 9).

Item	Generic timing	Example 1920x1080 system	Example 2048x1080 system
Active picture period (each line)	0 to the last sample of active picture	0 -1919	0 - 2047
(Reference point of the encryption)	First word of EAV	1920	2048
AES input reset LE_Key change	L_{AES} prior to the first word of EAV	1920 - L_{AES}	2048 - L_{AES}
Stream converter reset	Word immediately following EAV	1924	2052

Figure 8 – Timing relationship among AES input, Stream converter reset and LE_Key change

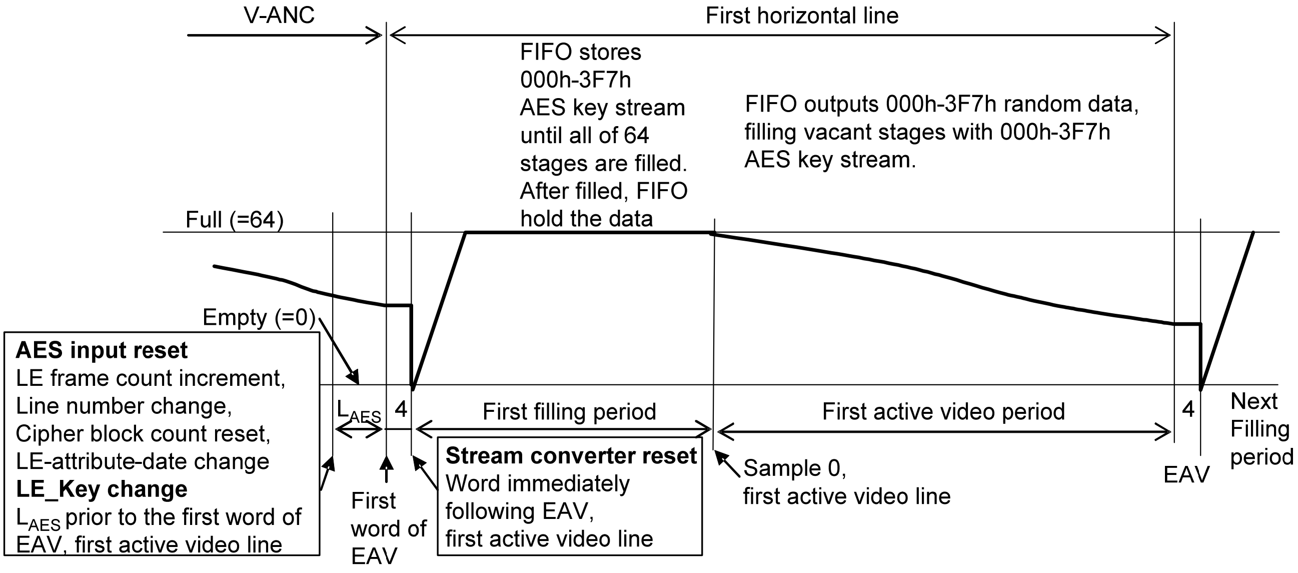

V-ANC
First horizontal line
FIFO stores 000h-3F7h AES key stream until all of 64 stages are filled. After filled, FIFO hold the data
FIFO outputs 000h-3F7h random data, filling vacant stages with 000h-3F7h AES key stream.
Full (=64)
Empty (=0)
AES input reset
LE frame count increment,
Line number change,
Cipher block count reset,
LE-attribute-date change
LE_Key change
L_AES prior to the first word of EAV, first active video line
L_AES
4
First word of EAV
First filling period
Stream converter reset
Word immediately following EAV, first active video line
First active video period
Sample 0, first active video line
4
EAV
Next Filling period

Figure 9a – FIFO behavior in the first active picture line

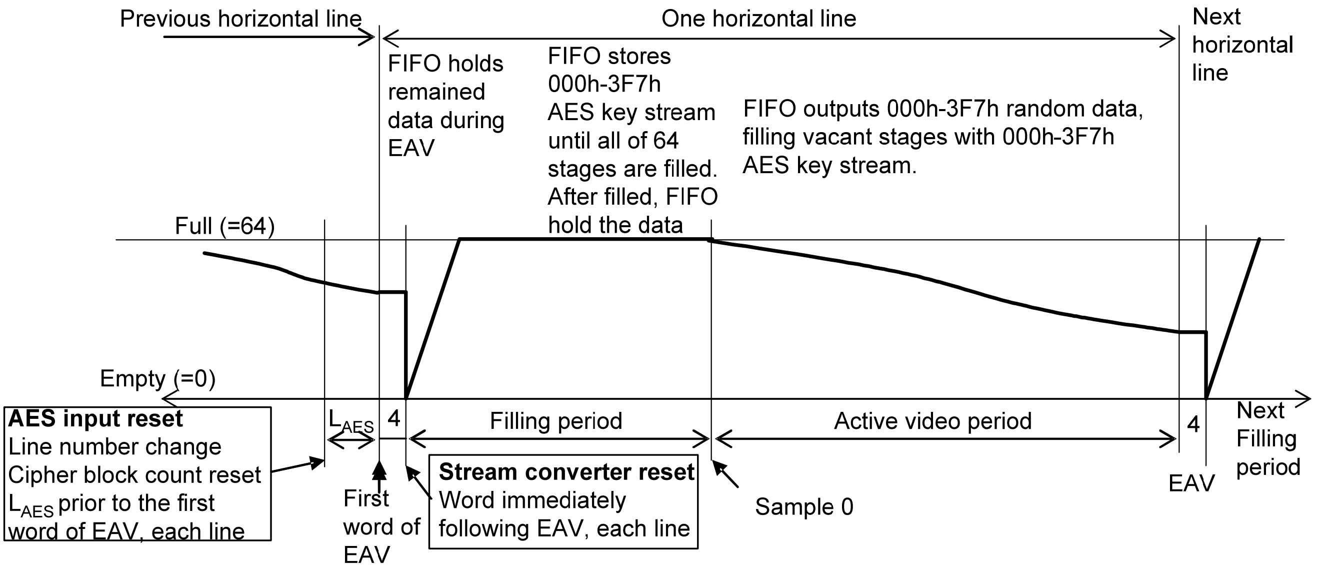

Previous horizontal line
One horizontal line
Next horizontal line
FIFO holds remained data during EAV
FIFO stores 000h-3F7h AES key stream until all of 64 stages are filled. After filled, FIFO hold the data
FIFO outputs 000h-3F7h random data, filling vacant stages with 000h-3F7h AES key stream.
Full (=64)
Empty (=0)
AES input reset
Line number change
Cipher block count reset
L_AES prior to the first word of EAV, each line
L_AES
4
First word of EAV
Filling period
Stream converter reset
Word immediately following EAV, each line
Active video period
Sample 0
4
EAV
Next Filling period

Figure 9b – FIFO behavior in other horizontal lines

9 Link EncryptionMetadata

9.1 Link Encryption Metadata Items

The Link Encryption metadata items shall be as shown in Table 10.

Table 10 –Link Encryption Metadata

Item	Description	Size	Related section
Next LE_Key ID	ID to identify the LE_Key for the next	32-bit	5.3
Current LE_Key ID	ID to identify the LE_Key currently in use.	32-bit	
Current Link Encryption frame count	Number of frames from the previous key change. The number is set to 0 at each key change and increments every frame thereafter.	24 bit	7.1
Version	Zero (0) for first issue of this Standard.	6-bit	
Key change timing	Command to generate the key change trigger and to show the number of frames to the next key change trigger. During the frame number exceeds 3, Key change timing keeps the frame number value as 3.	2-bit	5.3
Link_Number	0: Single link or Link-A of Dual Link, 1: Link-B of Dual Link, 2 – 3: Reserved.	2-bit	7.1

9.2 Data Structure of the Link Encryption Metadata

The data structure of the Link Encryption metadata shall be based on SMPTE 291M and as shown in Table 11. The Link Encryption metadata shall be mapped into the vertical ancillary data area of the Y' channel as specified in SMPTE 292 and as shown in Figure 8. Each metadata item shall be mapped on the user data area of the ANC packet from LSB to MSB and from lower byte to upper byte.

Table 11 – Data structure of the Link Encryption Metadata

Bits	**Byte 1 to 4**	**Byte 5-8**	**Byte 9 to 11**	**Byte 12**	**Byte 13**
Bit 9 (MSB)	Complement of bit 8				
Bit 8	Even parity of bit 0 through bit 7				
Bit 7	(bit 31)	(bit 31)	(bit 23)	(bit 5)	(bit 5)
Bit 6					
Bit 5					
Bit 4	Next LE_Key ID	Current LE_Key ID	Current Link Encryption Frame count	Version	Reserved
Bit 3					
Bit 2				(bit 0)	(bit 0)
Bit 1				Key change timing	Link _Number
Bit 0 (LSB)	(bit 0)	(bit 0)	(bit 0)		

9. 3 Ancillary Data Specification

The Ancillary data packet used by the Link Encryption metadata shall use the Type 2 data identification as defined in SMPTE 291M. The DID word shall be set to the value ‘40h’. The SDID word shall be set to the value of ‘06h’. Table 12 outlines the ancillary data packet words and values. The total size of the ancillary data packet is 20 words.

Table 12 – Ancillary Data Packet Structure for the Link Encryption Metadata

Name	Acronym	Value
Ancillary Data Flag (10-bit words)	ADF	000h, 3FFh, 3FFh
Data Identification	DID	40h
Secondary Data Identification	SDID	06h
Data Count	DC	0Dh
User data (Link Encryption Metadata)	UDW	(13 words)
Check Sum	CS	-

Annex A (Informative)
Probability of FIFO being Unfilled and Vacant

A.1 Probability of FIFO (64 stages) being Unfilled During an Example Filling Period (100 Samples)

The minimum filling period is 273 samples for 1920x1080/30/P system. The 100 samples used for the calculation allow for future extensibility.

The probability of FIFO (64 stages) being unfilled during the filling period (100 samples) can be calculated as follows.

The values 000h-003h and 3FCh-3FFh(1020-1023) are designated synchronizing signals; these values are prohibited for data. The probabilities where a LE_Key takes the valid and invalid codes are 1016/1024 and 8/1024 respectively. The probability of an LE_Key taking 37 invalid codes during the filling period (100 samples) can be calculated as shown in equation A.1.1, where ${}_{100}C_{37}$ denotes this combination.

$$P = \left(\frac{1016}{1024}\right)^{100-37} \text{x} \left(\frac{8}{1024}\right)^{37} \text{x} \ \ 100\,C\,37 \qquad \text{(A.1.1)}$$

The probability of an LE_Key taking more than 37 invalid codes during the filling period can be calculated as shown in equation A.1.2.

$$P = \sum_{n=37}^{100} \left[\left(\frac{1016}{1024}\right)^{100-n} \text{x} \left(\frac{8}{1024}\right)^{n} \text{x} \ \ 100\,C\,n \right] \approx 10^{-50} \qquad \text{(A.1.2)}$$

A.2 Probability of FIFO (64 stages) being Empty During an Example Horizontal Active Picture Period (e.g. 2048 Samples)

The probability of the FIFO (64 stages) being empty during the horizontal active picture period (e.g. 2048 samples) can be calculated in the same way as shown in equation A.2.1.

$$P = \sum_{n=65}^{2048} \left[\left(\frac{1016}{1024}\right)^{2048-n} \text{x} \left(\frac{8}{1024}\right)^{n} \text{x} \ \ 2048\,C\,n \right] \approx 10^{-20} \qquad \text{(A.2.1)}$$

Annex B (Informative)
System Behavior Under Error Conditions

B.1 Repetition of Link Encryption Metadata

The Ancillary data packet used by the link encryption metadata may appear more than once during the VANC for a single frame. All such packets appearing during a single VANC interval shall carry exactly the same data. (Repetition of data packets provides protection against data errors.)

B.2 Behavior in the Absence of Link Encryption Metadata

Link encryption metadata may be lost due to data errors. The metadata scheme is designed such that a receiving device may predict the value of the lost data based on previously received metadata. In particular:

1. the Link Encryption frame count may “freewheel” based on the last correctly-received frame count

2. the Key change event may be predicted based on a correctly-received packet containing a “Key change timing” value of 2 or 1, even if the packet containing value 0 is lost.

If no valid link encryption metadata packet has been received by the end of the picture line one line prior to the first line of active picture, the receiver may follow the recovery methods suggested above.

B.3 Recovery from Link Loss

The entire link may be lost and then reconnected. Upon receiving a link encryption metadata packet the receiving device can be immediately resynchronized by using the LE_Key ID currently in use.

After an extended link loss or power outage, the computational delay described in § 7.2 may interfere with prompt restoration of the program. Sending devices may support rapid restart by employing procedures based on the nonvolatile nature of key IDs 1-4, as specified in § 7.1.

B.4 “Hot Switching” Between Program Streams

The input to a receiving device (projector) may be switched either electronically (through a signal router) or manually from one server to another. This situation is similar to link loss and recovery, with the additional factor that the LE_Keys for the new program will likely be derived from an independent sequence. If this type of operation is anticipated, it is recommended that the most significant bits of the LE_Key ID field be dedicated to program source identification, and the lower order bits be used for change of keys within a program. This will ensure that keys delivered to the projector may be uniquely associated with the correct program source by means of the LE_Key ID alone.

B.5 Resistance to Line Noise

When operating in electrically noisy environments the serial interface may encounter bit errors. Certain special conditions may also cause error propagation by means of missed sync words (EAV or SAV), false sync words, or corrupted line count values. Reference to the embedded line count associated with the EAV sync word provides resistance to major timing disturbances.

Annex C (Informative)
Consideration of Frame Rates, P, PsF and I

C.1 Frame Rates

SMPTE 292 and SMPTE 372 define operation at many picture frame rates: 24/1.001, 24, 25, 30/1.001, 30, 50, 60/1.001 and 60. Since change timings of LE_Key and AES input items are based on the EAV of the first active picture line, changes in frame rate do not affect these timings.

C .2 Progressive, Progressive Segmented Frame (PsF) and interlace

SMPTE 292 and SMPTE 372 allow configurations of progressive (P), progressive segmented frame (PsF) and interlace (I). Note that P signals have one V-ANC and one active picture area in a frame, while PsF and I signals have two V-ANCs and two active picture areas in a frame. Although this difference does not affect the change timing of LE_Key and AES inputs, the frequencies of the Current link encryption frame count differ one to two between P and PsF/I as shown in Figure C.1.

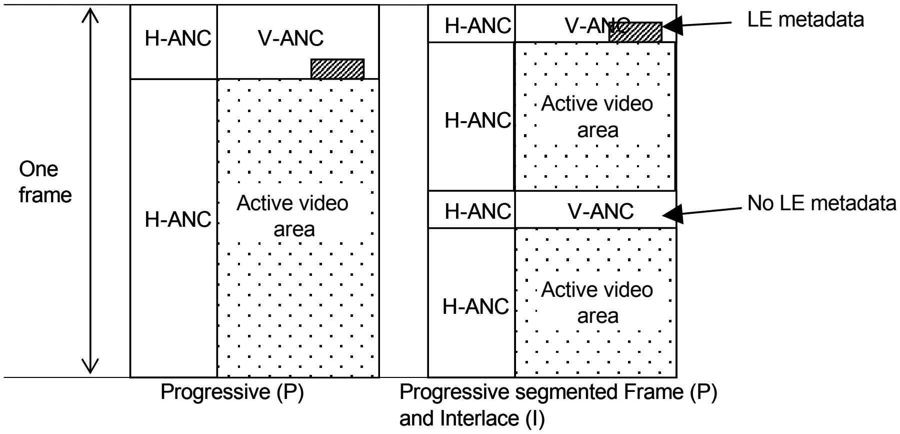


Figure C.1 – P, PsF and I configurations

Annex D (Informative)
Ancillary Packet Structure of LEKM and Link Encryption Metadata

The Ancillary packet structure of LEKM, its relationship to LEKP and the ancillary packet structure of Link Encryption Metadata are as shown in Figure D.1.

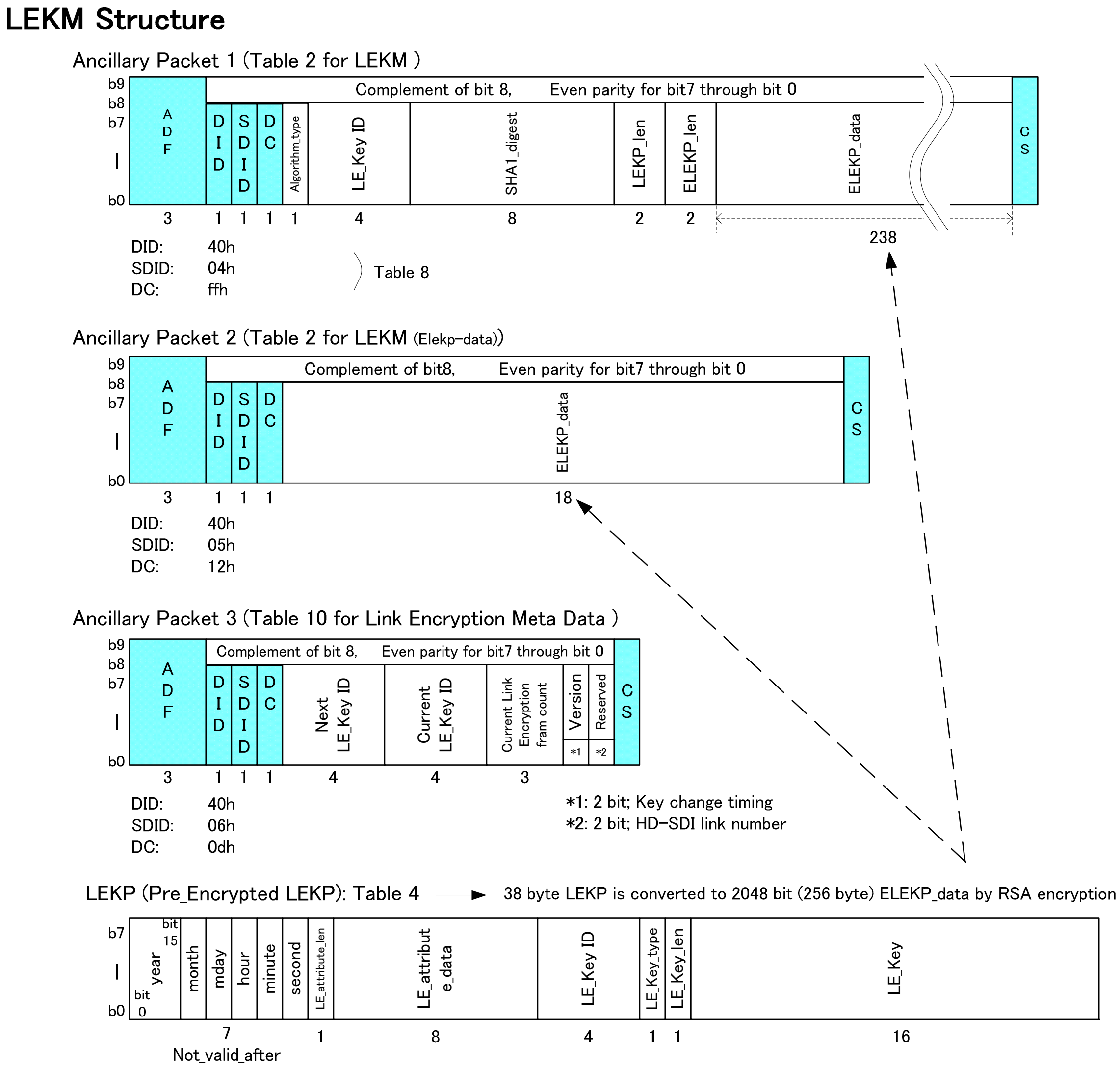

LEKM Structure
Ancillary Packet 1 (Table 2 for LEKM)
b9
b8
b7
b0
ADF
Complement of bit 8, Even parity for bit7 through bit 0
DID
SDID
DC
Algorithm_type
LE_Key ID
SHA1_digest
LEKP_len
ELEKP_len
ELEKP_data
CS
3 1 1 1 1 4 8 2 2
238
DID: 40h
SDID: 04h
DC: ffh
Table 8
Ancillary Packet 2 (Table 2 for LEKM (Elekp-data))
Complement of bit8, Even parity for bit7 through bit 0
ELEKP_data
3 1 1 1
18
DID: 40h
SDID: 05h
DC: 12h
Ancillary Packet 3 (Table 10 for Link Encryption Meta Data)
Complement of bit 8, Even parity for bit7 through bit 0
Next LE_Key ID
Current LE_Key ID
Current Link Encryption fram count
Version
Reserved
*1
*2
3 1 1 1 4 4 3
DID: 40h
SDID: 06h
DC: 0dh
*1: 2 bit; Key change timing
*2: 2 bit; HD-SDI link number
LEKP (Pre_Encrypted LEKP): Table 4
38 byte LEKP is converted to 2048 bit (256 byte) ELEKP_data by RSA encryption
bit 15
year
bit 0
month
mday
hour
minute
second
LE_attribute_len
LE_attribute_data
LE_Key ID
LE_Key_type
LE_Key_len
LE_Key
7 1 8 4 1 1 16
Not_valid_after

Figure D.1 – Ancillary packet structure of LEKM and Link Encryption Metadata

Annex E (Informative)
Stream Converter FIFO 1st Write Data at Filling Period and 65th Write Data at Active Picture Period

Figures E1, E2 and E3 show the Stream converter FIFO 1st active data at Filling period (common in 1920x1080/ 24P and 2048x1080/ 24P), 65th Write data at Active picture of 1920x1080/ 24P and 65th Write data at Active picture of 2048x1080/ 24P, respectively.

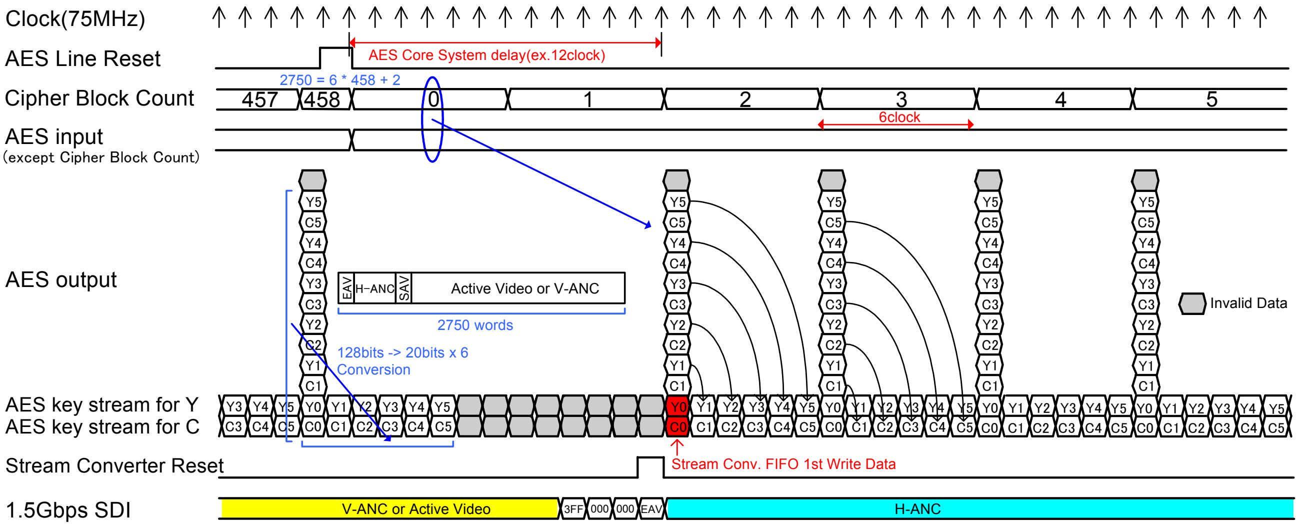


Figure E.1 – Stream Converter FIFO 1st Write Data at Filling Period (1920/2048-1080 24P)

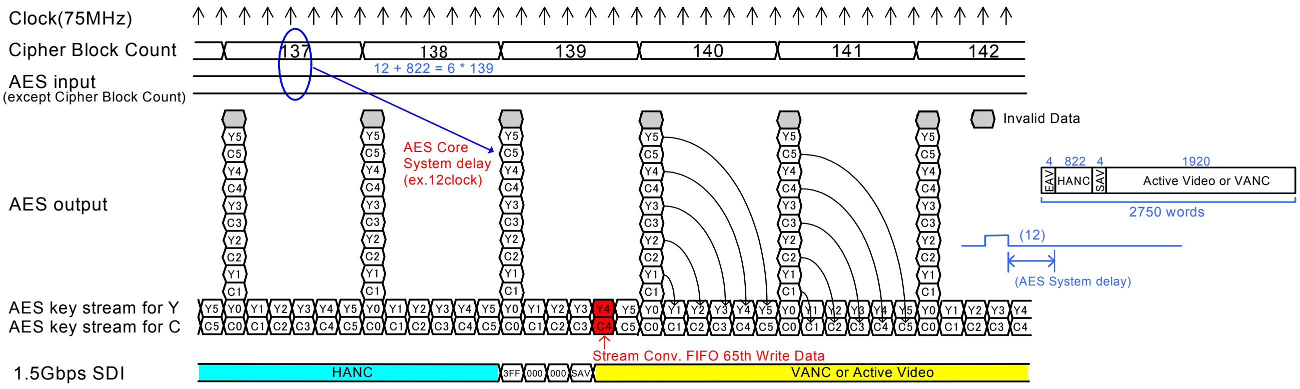


Figure E. 2 – Stream Converter FIFO 65th Write Data at Active Video Period (1920-1080 24P)

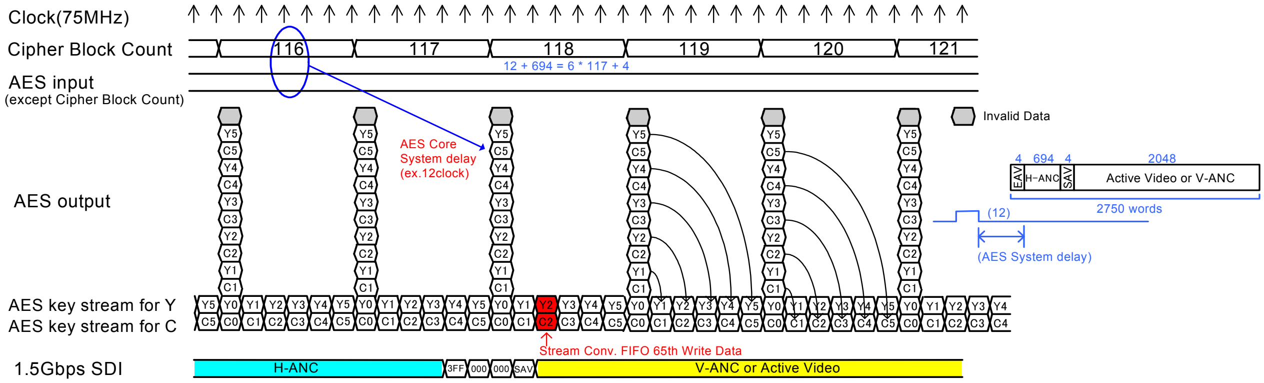


Figure E.3 – Stream Converter FIFO 65th Write Data at Active Video Period (2048-1080 24P)

Annex F (Informative) Test Vector Description

The two 128-bit inputs to the AES core are the AES Input and the Link Encryption Key. These inputs are loaded into the AES core most-significant byte first. The most-significant byte of the AES core output is not used. The lower 16 bits of the AES Input are variable because the cipher block count is reset to 0x0000 at the beginning of each line and increments every six clocks.

The test vector values below are formatted from MSB on the left to LSB on the right. The notation used for the Link Encryption Key and AES_Input inputs are the same as used for the PLAINTEXT and KEY inputs of Appendix C.1 in FIPS 197.

Plain_y & Plain_C are at a fixed value “0x136” in the example in order to make easy to evaluate the AES encryption/decryption system.

Cipher_y & Cipher_c are encrypted data.

The point is to evaluate if the right keys are generated and stored in the 64-stage FIFO, and applied to the correct data. The 65th key is particularly important to ensure interoperability.

For Example:
Link Encryption Key = all “0” for 128 bit
AES input data by Table 9:

	PlainTX_136A	PlainTX_136B
SDI Link_Number	00 (bin)	01 (bin)
Reserved	000000 (bin)	000000 (bin)
LE-attribute-data	0x887844a4c07444d0 (hex)	0x887844a4c07444d0 (hex)
Current Link Encryption frame count	0x000024 (hex)	0x000024 (hex)
Line_Number	0x002a (hex)	0x002a (hex)
Cipher block count *1	0x0000 to 0x008a (hex)	0x0000 to 0x008a (hex)

*1: Inhibited keys are rejected and the FIFO stores up to 64 keys for Y & C data in HANC, but the cipher block counter is increasing continuously.

See the following 3 test vectors

PlainTX_136A (Link A: 1920x1080x24P)

Link Encryption Key is 00000000000000000000000000000000
AES_input 00887844a4c07444d0000024002a0000
Cipher Block Count = 0000 to 008a (70 Keys for 70 bytes of Y & C Data)

Plain_y

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Plain_c

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Cipher_y

3a8	0d7	0ab	207	264	38c	0b9	37a	1bf	198	328	30c	0c0	025	085	326
206	057	172	389	0a3	0f0	054	1b3	389	1aa	0c2	37e	242	05c	257	2b1
135	301	045	3c2	1ff	0ab	082	005	184	1c3	3d0	256	2cc	078	0af	091
3e4	0e2	157	2ff	232	1d9	14d	31f	362	3c5	35e	1ce	21c	3e9	0f2	25f
1da	145	1d2	2b9	07e	2f6										

Cipher_c

065	216	35b	2e4	0e5	27b	2e3	3f9	047	258	1c1	317	00a	3d5	109	0ae
3aa	186	1b4	3ea	2cf	214	3c7	371	136	051	032	32d	204	243	2a3	2f8
184	31d	076	15d	1d1	14a	3c8	3eb	1cd	0a9	11e	00f	1aa	3a2	263	0b0
2ed	3f1	320	0f5	0ed	2ca	032	385	10a	173	180	16c	370	321	1a1	157
3db	32f	1b2	313	372	0a1										

PlainTX_136B (Link B: 1920x1080x24P)

Link Encryption Key is 00000000000000000000000000000000
AES_input 40887844a4c07444d0000024002a0000
Cipher Block Count = 0000 to 008a (70 Keys for 70 bytes of Y & C Data)

Plain_y

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Plain_c

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Cipher_y

1a1	264	2ee	37c	3ee	1d9	366	37	24d	31d	b8	325	39e	200	30e	2bc
220	1c3	81	3cd	af	2aa	3ca	1ae	27	3cc	2fa	320	87	3e	3a4	2f9
1cf	ac	2c2	25	130	1d3	dd	4	153	264	2c4	2d9	37c	3c	7e	b8
265	285	385	17d	181	288	27d	38e	3c3	240	c4	160	2ec	2e0	267	396
b0	30	31c	16b	260	305										

Cipher_c

323	3bc	01f	2bd	302	020	01d	361	3ee	3a8	3cd	1de	2a5	1f6	0b5	067
24e	36a	33d	0a1	037	06d	1ee	14e	1f0	041	21e	0c9	36f	25f	383	2bc
2c6	363	3c1	3ee	2e2	113	037	265	2ed	109	3d5	2c5	195	162	28f	185
161	2d8	05b	359	143	387	1a1	3b5	3e3	3ec	0a4	0e2	32c	02d	29a	171
33c	111	019	140	108	134										

PlainTX_136A_00 (Link A: 1920x1080x24P)

Link Encryption Key is 00000000000000000000000000000000
AES_input 000000000000000000000024002a0000
Cipher Block Count = 0000 to 008a (70 Keys for 70 bytes of Y & C Data)

Plain_y

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Plain_c

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Cipher_y

10c	10e	16c	2fc	3eb	359	1d6	292	29a	123	073	3cf	388	0b1	37f	178
1fa	0d6	034	31c	088	1e5	3d0	37a	17a	036	2f1	29a	073	0a6	291	291
2de	203	362	327	389	2d6	104	2a6	2bf	032	212	29a	0dd	313	0fd	137
350	3e3	281	3cf	0c8	1b2	282	154	2a8	0b0	094	11d	0c4	239	1de	1d1
312	26b	0f6	27a	144	2e8										

Cipher_c

2e7	02d	34d	3f5	1b1	16e	081	373	262	2b6	0f9	1a0	3be	065	265	32f
387	076	00a	2be	1c1	00b	3cc	0a4	3be	0cd	09a	0a0	2fe	2e4	29d	05c
2d1	0c9	04b	386	249	28c	2e2	217	3e4	16e	02e	1d1	1b3	2a2	29a	077
34e	127	179	3bf	2d4	144	2d1	20e	2b2	13d	2fe	12f	1e3	1f9	1c1	05a
0e8	1f0	1b1	130	119	364										

PlainTX_136B_00 (Link B: 1920x1080x24P)

Link Encryption Key is 00000000000000000000000000000000
AES_input 400000000000000000000024002a0000
Cipher Block Count = 0000 to 008a (70 Keys for 70 bytes of Y & C Data)

Plain_y

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Plain_c

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Cipher_y

006	218	322	064	30b	17f	29f	3e6	2eb	3bb	3cb	18d	20e	1e1	1ce	26f
006	3ea	065	190	363	3af	067	07b	01e	263	2da	2ff	186	35b	3ab	1df
30d	1a8	195	2c4	3fb	213	08e	3f3	2a8	140	331	36e	121	074	03f	3cd
242	1a8	3a8	096	32e	38c	19c	39b	283	126	03e	1f1	2bf	2da	2eb	178
275	2f6	101	37c	2de	326										

Cipher_c

30e	268	2fb	124	282	32c	2a7	075	3ed	27a	10b	0db	244	3cb	359	15a
0a2	076	1b0	32e	3c2	1e5	34d	313	30d	3a5	12b	25a	29c	248	3ad	059
126	384	3da	1c0	3e1	032	071	389	37b	0a5	390	1e4	095	394	068	32b
11e	153	0f0	1b3	324	06f	337	230	32c	12a	0bb	383	1b0	0d7	38b	395
3e9	278	247	33e	3f9	0ce										

PlainTX_136A_00_2048 (Link A: 2048x1080x24P)

Link Encryption Key is 00000000000000000000000000000000
AES_input 000000000000000000000024002a0000
Cipher Block Count = 0000 to 0075 (70 Keys for 70 bytes of Y & C Data)

Plain_y

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Plain_c

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Cipher_y

10c	10e	16c	2fc	3eb	359	1d6	292	29a	123	073	3cf	388	0b1	37f	178
1fa	0d6	034	31c	088	1e5	3d0	37a	17a	036	2f1	29a	073	0a6	291	291
2de	203	362	327	389	2d6	104	2a6	2bf	032	212	29a	0dd	313	0fd	137
350	3e3	281	3cf	0c8	1b2	282	154	2a8	0b0	094	11d	0c4	239	1de	1d1
2af	3ec	142	194	101	379										

Cipher_c

2e7	02d	34d	3f5	1b1	16e	081	373	262	2b6	0f9	1a0	3be	065	265	32f
387	076	00a	2be	1c1	00b	3cc	0a4	3be	0cd	09a	0a0	2fe	2e4	29d	05c
2d1	0c9	04b	386	249	28c	2e2	217	3e4	16e	02e	1d1	1b3	2a2	29a	077
34e	127	179	3bf	2d4	144	2d1	20e	2b2	13d	2fe	12f	1e3	1f9	1c1	05a
075	2d5	219	390	0f9	065										

PlainTX_136B_00_2048 (Link B: 2048x1080x24P)

Link Encryption Key is 00000000000000000000000000000000
AES_input 40000000000000000000000024002a0000
Cipher Block Count = 0000 to 0075 (70 Keys for 70 bytes of Y & C Data)

Plain_y

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Plain_c

136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136	136	136	136	136	136	136	136	136	136	136
136	136	136	136	136	136										

Cipher_y

006	218	322	064	30b	17f	29f	3e6	2eb	3bb	3cb	18d	20e	1e1	1ce	26f
006	3ea	065	190	363	3af	067	07b	01e	263	2da	2ff	186	35b	3ab	1df
30d	1a8	195	2c4	3fb	213	08e	3f3	2a8	140	331	36e	121	074	03f	3cd
242	1a8	3a8	096	32e	38c	19c	39b	283	126	03e	1f1	2bf	2da	2eb	178
3d2	390	2e9	213	0ab	2f8										

Cipher_c

30e	268	2fb	124	282	32c	2a7	075	3ed	27a	10b	0db	244	3cb	359	15a
0a2	076	1b0	32e	3c2	1e5	34d	313	30d	3a5	12b	25a	29c	248	3ad	059
126	384	3da	1c0	3e1	032	071	389	37b	0a5	390	1e4	095	394	068	32b
11e	153	0f0	1b3	324	06f	337	230	32c	12a	0bb	383	1b0	0d7	38b	395
318	349	358	079	389	242										

Annex G (Informative)
Index of Acronyms and Terms Used

The following table shows the location of acronyms and terms defined or used in this document.

Acronym or term	Location of primary description	Other Location used in this document
Active picture period	5.2	6.3, 8.2.2,
ADF (Ancillary Data Flag)	Refer to SMPTE 291M	7.5, 9.3
AES input	8	
AES input reset	8.2.2	6.3, 8.2.1
Algorithm_type	7.3.1	7.3.3, 7,3,5
Cipher block count	8.1	8.2.1, 8.2.2
Current LE_Key	9.1	6.3
Current Link Encryption frame count	8.1, 9.1	8.2.1, 8.2.2, 9.2
DC (Data Count)	Refer to SMPTE 291M	7.5, 9.3
DID (Data Identification)	Refer to SMPTE 291M	7.5, 9.3
ELEKP_data (Encrypted LEKP)	7.3.6	7.3, 7.3.5, 7.5
ELEKP_len (Length of Encrypted LEKP)	7.3.5	7.3, 7.3.6
Encryption Modulator	5.3	
Encryption Demodulator	5.4	
FIFO emergency value	5.2.2	
Filling period	5.2.1	5.2
Link_Number	8.1, 9.1	
Key change timing	6.3	9.1, 9.2
Key change trigger	6.3	9.1
L_{AES} (Latency of AES core)	6.3	8.2.1, 8.2.2
LE_attribute_data	7.4.3	7.4, 7.4.2, 7.4.5, 8.1
LE_attribute_len	7.4.2	7.4, 7.4.3
LE_Key	6	7, 8
LE_Key change	8.2.2	7.4.3, 8.2.1
LE_Key ID	6.2	6.1, 6.3, 7.1, 7.2, 7.3, 7.4, 7.5
LEKM (Link Encryption Key Message)	7	6.2, 6.3
LE_Key change timing	6.3	
LE_Key_len	7.4	
LE_Key_type	7.4.5	7.4
LEKP (Link Encryption Key Payload)	7	
LEKP_len (Length of LEKP)	7.3.4	7.3
Line_Number	8.1	8.2.1
Link Encryption Metadata	9	2, 6.2, 6.3
Next LE_Key	9.1	6.3
Non volatile storage	7.1	6.3
Not_valid_after	7.4.1	7.4
Parallel (120bit)-parallel(20bit) converter	5.1	
Prohibited codes	4	5.3
SDID (Secondary Data Identification)	Refer to SMPTE 291M	7.5, 9.3
SHA1_digest	7.3.3	7.2, 7.3
Stream converter	5.2	5, 5.3, 7.4.5, 8.2
Stream converter reset	8.2	5.2
UDW (User Data Word)	Refer to SMPTE 291M	7.5, 9.3