

SMPTE STANDARD

Link Encryption for 1.5 Gb/s¹ Serial Digital Interface



¹ Nominal total Bit Rate

Page 1 of 34 pages

Table of Contents

Page

Foreword	3
Intellectual Property	3
1 Scope	4
2 Conformance Notation	4
3 Normative References	4
4 Introduction	5
5 Encryption/Decryption Engine.....	7
5.1 AES Core and Parallel (12Bit)-Parallel (20-Bit) Converter	7
5.2 Stream Converter.....	7
5.2.1 Filling period.....	8
5.2.2 Active picture period	8
5.3 Encryption Modulator	8
5.4 Encryption Demodulator	10
6 Link Encryption Key (LE_Key)	10
6.1 LE_Key Generation.....	10
6.2 LE_Key Distribution.....	10
6.3 LE_Key Change Timing	10
7 Link Encryption Key Message (LEKM) and Link Encryption Key Payload (LEKP).....	11
7.1 Overview	11
7.2 Maximum Recipient Processing Delay for LEKM Updates.....	12
7.3 Link Encryption Key Message Structure (LEKM).....	12
7.3.1 Algorithm_type	13
7.3.2 LE_Key ID	13
7.3.3 SHA1_digest	13
7.3.4 LEKP_len	14
7.3.5 ELEKP_len.....	14
7.3.6 ELEKP_data.....	14
7.4 Link Encryption Key Payload (LEKP) Structure.....	14
7.4.1 Not_valid_after	15
7.4.2 LE_attribute_len	15
7.4.3 LE_attribute_data.....	15

7.4.4	LE_Key ID	15
7.4.5	LE_Key_type (Dynamic).....	16
7.4.6	LE_Key_len	16
7.4.7	LE_Key	16
7.5	LEKM Distribution through 1.5 Gb/s Serial Interface(s)	16
8	AES Input.....	18
8.1	AES Input Items.....	18
8.2	AES Input and Stream Converter Change Timing	18
8.2.1	Change timing for the first active picture line	18
8.2.2	Change timing for other active picture lines	19
9	Link Encryption Metadata	21
9.1	Link Encryption Metadata Items	21
9.2	Data Structure of the Link Encryption Metadata.....	21
9.3	Ancillary Data Specification	22
Annex A	Probability of FIFO being Unfilled and Vacant (Informative)	23
A.1	Probability of FIFO (64 stages) being Unfilled During an Example Filling Period (100 Samples)	23
A.2	Probability of FIFO (64 stages) being Empty During an Example Horizontal Active Picture Period (e.g. 2048 Samples).....	23
Annex B	System Behavior Under Error Conditions (Informative)	24
B.1	Repetition of Link Encryption Metadata	24
B.2	Behavior in the Absence of Link Encryption Metadata	24
B.3	Recovery from Link Loss.....	24
B.4	“Hot Switching” Between Program Streams	24
B.5	Resistance to Line Noise	24
Annex C	Consideration of Frame Rates, P, PsF and I (Informative)	25
C.1	Frame Rates	25
C.2	Progressive, Progressive Segmented Frame (PsF) and Interlace	25
Annex D	Ancillary Packet Structure of LEKM and Link Encryption Metadata (Informative)	26
Annex E	Stream Converter FIFO 1st Write Data at Filling Period and 65th Write Data at Active Picture Periods (Informative).....	27
Annex F	Test Vector Description (Informative)	29
Annex G	Index of Acronyms and Terms Used (Informative).....	34

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

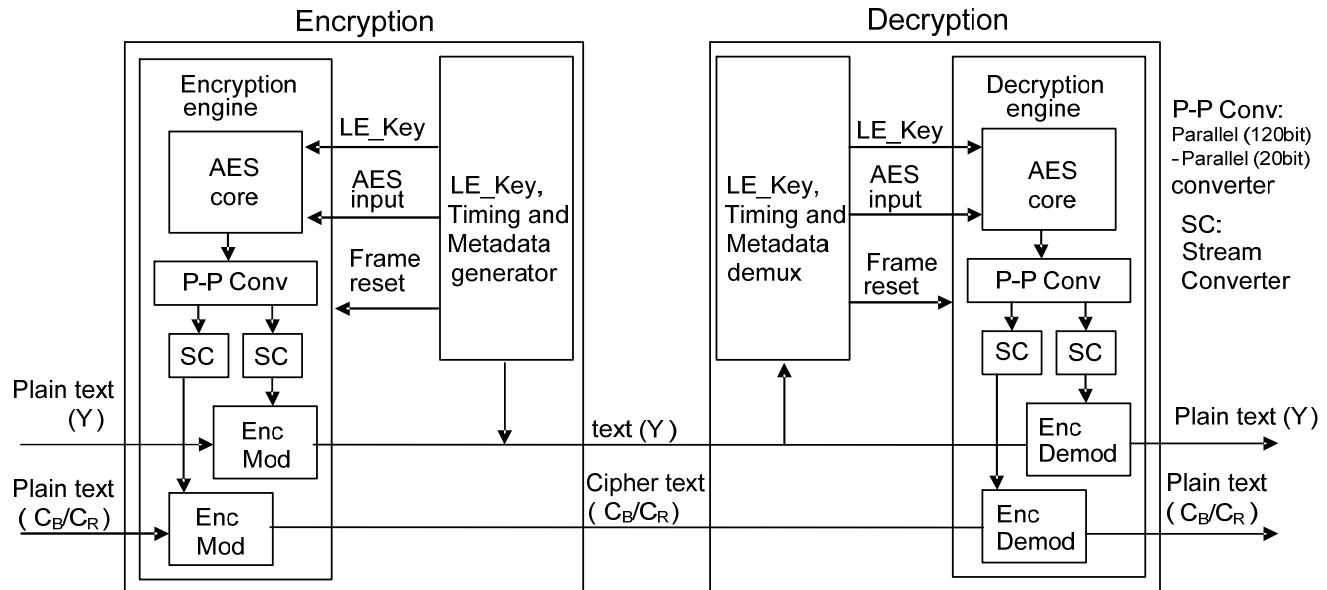
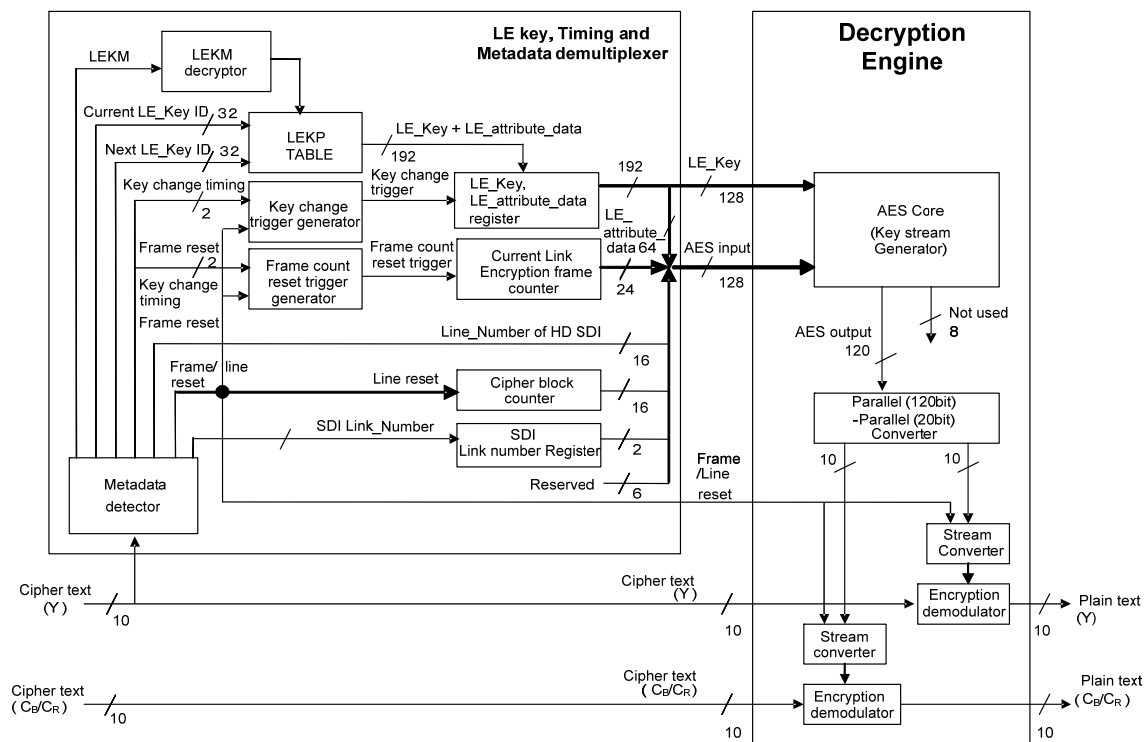


Figure 1 – Encryption and Decryption (Example)



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.

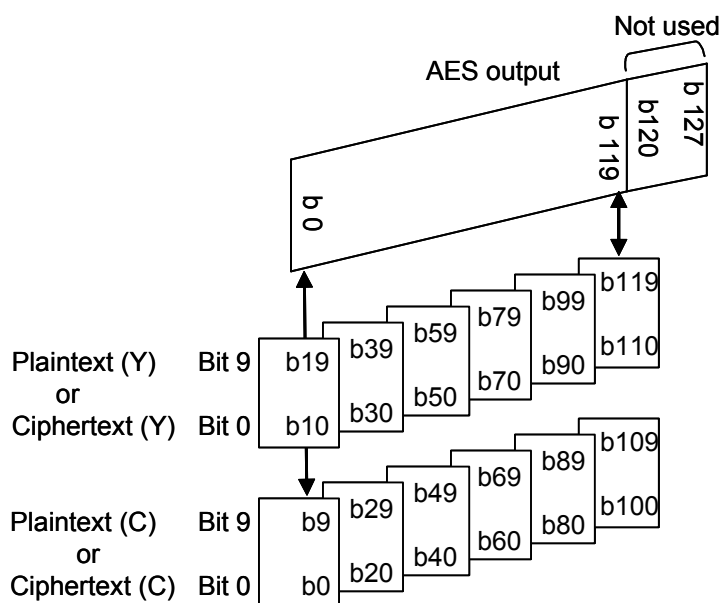


Figure 3 – Bit relationship between AES core output and plaintext (or ciphertext)

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.

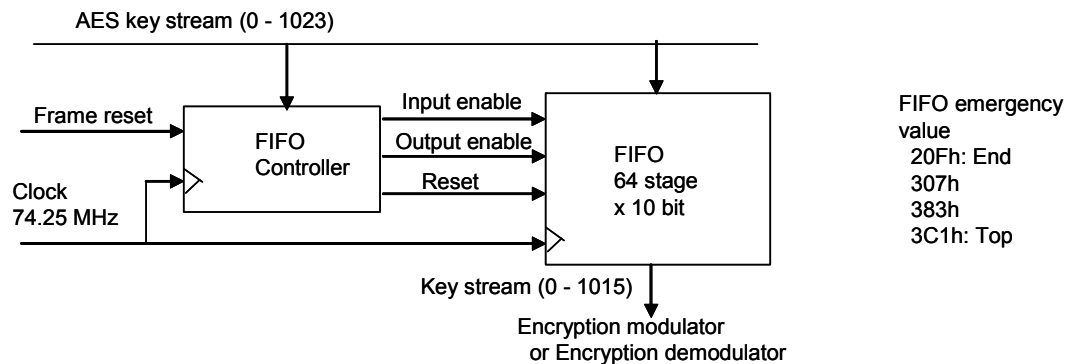


Figure 4 – Stream converter block diagram (Example)

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.

$$\text{Encryption: } C_i = [(M_i - N_1) + E_i] \bmod (N_2) + N_1 \quad (1)$$

$$\text{Decryption: } M_i = [(C_i - N_1) - E_i] \bmod (N_2) + N_1 \quad (2)$$

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.

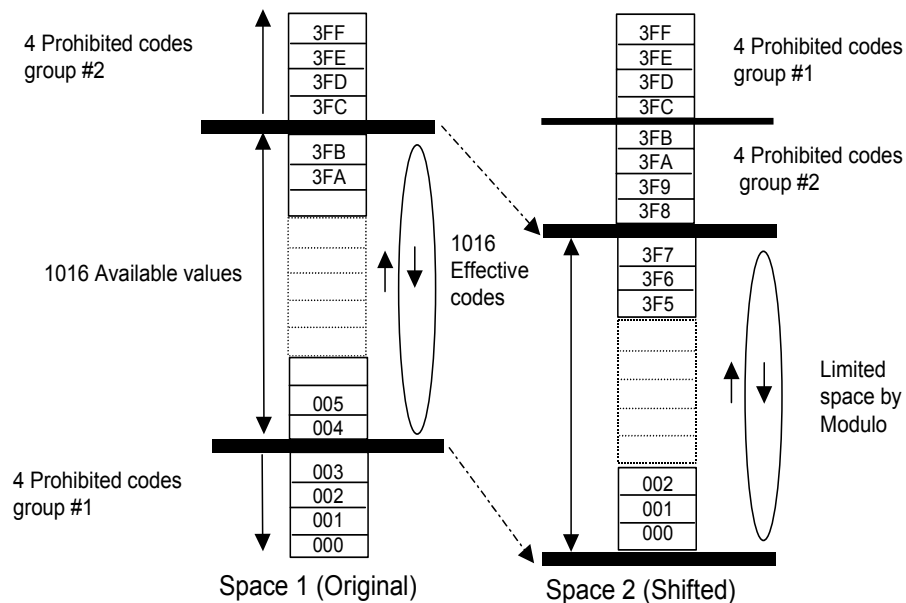


Figure 5 – Effective codes and prohibited codes

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:

- Plaintext data (input picture data) M_i shall take available values 004h (4) through 3FBh (1019) in space 1. $N_1 (=4)$ shall be subtracted from M_i , then the available values shall be shifted to effective codes 000h (0) through 3F7h (1015) in space 2.
- A pseudo-random number E_i between 000h (0) and 3F7h (1015) shall be added to the shifted available values.
- $[(M_i - 4) + E_i] \text{ modulo } 1016$ limits the output values from 000h (0) through 3F7h (1015). And after $N_1 (=4)$ 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 C_i shall take available values 004h (4) through 3FBh (1019) in space 1, N_1 (= 4) shall be subtracted from C_i , and the available values shall be shifted to space 2 which has values between 000h (0) and 3F7h (1015).

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

$[(C_i - 4) - E_i]$ modulo 1016 limits the output values from 000h (0) through 3F7h (1015). After adding N_1 (=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 payout. 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.

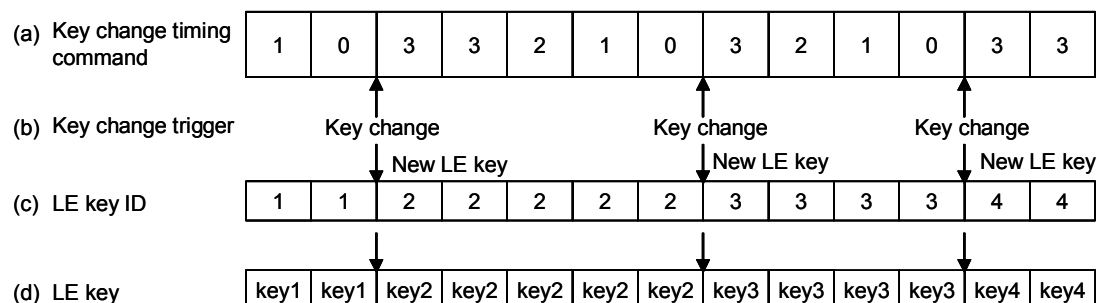


Figure 6 – Relationship among the Key change timing command, Key change trigger, LE_Key ID and LE_Key

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.