Digital Video Broadcasting (DVB);
Content Scrambling Algorithms for DVB-IPTV Services using
MPEG2 Transport Streams
Contents

Intellectual Property Rights ................................................................. 4
Foreword .......................................................................................... 4
1 Scope ......................................................................................... 5
2 References .................................................................................. 5
  2.1 Normative references .............................................................. 5
  2.2 Informative references .......................................................... 5
3 Definitions, symbols and abbreviations ......................................... 6
  3.1 Definitions .............................................................................. 6
  3.2 Symbols .................................................................................. 6
  3.3 Abbreviations ........................................................................ 6
4 General Aspects ........................................................................... 7
5 Hardware-oriented Scrambling Algorithms ..................................... 7
  5.1 Introduction .......................................................................... 7
  5.2 DVB Common Scrambling Algorithm version 3 ......................... 7
    5.2.1 Introduction .................................................................... 7
    5.2.2 Technical details .............................................................. 8
    5.2.3 Licensing ................................................................. 8
    5.2.4 CSA-3 for DVB IPTV Scrambling .................................... 8
  5.3 DVB Common Scrambling Algorithm version 1 ......................... 8
    5.3.1 Introduction .................................................................... 8
    5.3.2 Technical details .............................................................. 9
    5.3.3 Licensing ................................................................. 9
    5.3.4 CSA for IPTV scrambling ............................................... 9
6 Common IPTV Software-oriented Scrambling Algorithm (CISSA) Version 1 ............................................................ 9
  6.1 Introduction .......................................................................... 9
  6.2 General Description (informative) ............................................ 9
    6.2.1 TS level scrambling ....................................................... 10
    6.2.2 PES level scrambling .................................................... 10
  6.3 Normative Elements ............................................................. 11
    6.3.1 Encryption Elements ..................................................... 11
      6.3.1.1 Block Cipher .......................................................... 11
      6.3.1.2 Initialization Vector ................................................. 11
      6.3.1.3 Chaining Mode ..................................................... 11
    6.3.2 TS packet Scrambling and Descrambling ......................... 11
    6.3.3 PES level Scrambling and Descrambling ......................... 12
7 Signalling ..................................................................................... 13
  7.1 Scrambling descriptor (informative) ........................................ 13
  7.2 Use of the scrambling descriptor ............................................ 13
Annex A (informative): CISSA Implementation Guidelines ................... 14
Annex B (informative): CISSA Test Vectors for TS level scrambling ........... 15
History ........................................................................................... 20
Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for ETSI members and non-members, and can be found in ETSI SR 000 314: “Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards”, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ÉLECTrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members’ activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardization, interoperability and future proof specifications.
1 Scope

The present document provides specifications for the scrambling of MPEG2-Transport Stream based content conveyed by DVB-IPTV services for both live media broadcast and content on demand services. In addition, it provides information regarding the relevant signalling that enables conditional access systems or digital rights management systems to protect their content using the scrambling methods described in the present document.

The scrambling of IPTV content not contained in MPEG2 Transport Stream is out of scope of the present document.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.


2.2 Informative references

The following referenced documents are necessary for the application of the present document.

[i.1] ISO/IEC 13818-1: "Information Technology -- Generic coding of moving pictures and associated audio: Systems, Recommendation H.222.0".

[i.2] ETSI ETR 289 (1996): "Digital Video Broadcasting (DVB); Support for use of scrambling and Conditional Access (CA) within digital broadcasting systems".
3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

cipher: algorithm used to provide confidentiality for an element of data of defined size

hardware-oriented scrambling algorithm: algorithm that is easy to implement in hardware and whose hardware implementation is efficient, and, inversely, that it is difficult and costly to implement efficiently in software on general purpose CPUs

scrambling algorithm: cipher-based algorithm used to encrypt DVB audiovisual and associated content

software-oriented scrambling algorithm: algorithm that is easy to implement in either hardware or software; and for which both software implementations on general purpose CPUs and hardware implementations are efficient

3.2 Symbols

For the purposes of the present document, the following symbols apply:

\[ E(K)(M) \] encryption of message M using key K
\[ D(K)(C) \] decryption of cipher text C using key K
\[ \oplus \] bitwise XOR operation on two blocks of 16 bytes
\[ \text{mod} \] modulo
\[ P_x \] \( x \)th plaintext block of 16 bytes, or possibly less for the last block
\[ C_x \] \( x \)th ciphertext block of 16 bytes, or possibly less for the last block
\[ 00^a \] string of \( a \) bytes with value 0
\[ \text{Array} \] All Array indices start at 0
\[ \text{Array}[a] \] Indexing: the \((a + 1)\)th element in an array
\[ i = a…b \] represents a range of numbers from \( i = a \) to \( i = b \) inclusive, \( a,b \in \mathbb{Z} \)

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AES Advanced Encryption Standard
CBC Cipher Block Chaining
CISSA Common IPTV Software-oriented Scrambling Algorithm
CPU Central Processing Unit
CSA Common Scrambling Algorithm
CSA1 Common Scrambling Algorithm Version 1
CSA3 Common Scrambling Algorithm Version 3
CW Control Word
DVB Digital Video Broadcast
IP Internet Protocol
IPTV Internet Protocol Television
IV Initialization Vector
NDA Non-Disclosure Agreement
PES Packetized Elementary Stream
TS Transport Stream
uimsbf unsigned integer most significant bit first
XRC eXtended emulation Resistant Cipher
4 General Aspects

The present document defines two hardware-oriented scrambling algorithms and one software-oriented scrambling algorithm, as per the definitions in clause 3.2 that are suitable for IPTV content transported in an MPEG2-TS container. Conformance with the present document is achieved, if and only if, at least one of the said algorithms is implemented.

To avoid any confusion, definitions of hardware-oriented and software-oriented scrambling algorithms shall be interpreted as follows:

- A hardware-oriented scrambling algorithm is intended to be implemented in hardware only, wherein it should provide acceptable performance in order to support all common use cases. It is expected, however, that any implementation in software - such as general purpose CPUs - is uncomfortably slow.

- A software-oriented algorithm is intended to be implemented primarily in software, with the possibility of a hardware implementation. It should provide acceptable performance in order to support all common use cases, regardless of the technology, hardware or software. In particular, the restriction to software-only implementations, without assistance of any dedicated hardware, should not restrict the usability of the scrambling algorithm; the use of a dedicated hardware should not be mandatory, nor excluded, yet the primary use case is a pure software implementation.

The aforementioned definition of hardware-oriented scrambling algorithm should be understood to mean "hardware-only", whereas software-oriented scrambling algorithm should be understood to mean "software and hardware".

5 Hardware-oriented Scrambling Algorithms

5.1 Introduction

The present document identifies two algorithms for which the descrambling method is hardware-oriented.

Clause 5 includes descrambling elements protected by intellectual property rights, in order to enable the enforcement of licensing conditions. The licensing fees will cover no more than the administrative costs.

Some parts of the descrambling elements included in clause 5 are kept confidential and the full scale details are not publicly available. Details are further given in clauses 5.2.3 and 5.3.3.

Two algorithms have been selected as solutions for the DVB IPTV scrambling algorithm: DVB Common Scrambling Algorithm Version 3, which shall be used for all normal implementations of the DVB IPTV Scrambler, and DVB Common Scrambling Algorithm Version 1 which should only be used for the purpose of backwards compatibility and in support of legacy implementations.

5.2 DVB Common Scrambling Algorithm version 3

5.2.1 Introduction

The DVB Common Scrambling Algorithm version 3 (CSA3) is a device, apparatus or mechanism (whether implemented as hardware or software) designed or specifically adapted, totally or partially, to render unintelligible a service compatible with Standards by the use of CSA3 Scrambling Technology and any modifications and improvements thereof and which can be descrambled using a common descrambling system in the form approved by the Steering Board of the DVB Project for Standards.

CSA3 is comprised of the DVB CSA3 Descrambling System and Scrambling Technology. The specification for each is distributed separately under arrangements with the European Telecommunications Standards Institute (ETSI), which acts as custodian for the companies which have developed the DVB CSA3 algorithm.
5.2.2 Technical details

CSA3, as specified for common DVB applications has been designed to minimise the likelihood of piracy attack over a long period of time and thus contains highly security sensitive information. The technical details of the scrambling algorithm can only be made available to bona-fide users upon signature of a Non-Disclosure Agreement (NDA) administered by ETSI. This clause contains an informative summary of the scrambling method and some of the implementation issues.

The scrambling algorithm operates on the payload of a Transport Stream (TS) packet in the case of TS-level scrambling. CSA3 uses a 128-bit key (Control Word) to encrypt and decrypt data blocks of any size over 16 bytes (with a granularity of 1 byte).

The encryption algorithm is based on two block ciphers: a variation of the "Advanced Encryption Standard" (AES128), specified in NIST FIPS 197 [1], called AES' and the "eXtended emulation Resistant Cipher" (XRC), which is a DVB-confidential cipher.

The CSA3 Key Derivation Mechanism uses a subset of IDEA-NXT, a block cipher published in 2004 in the academic world, which has been security assessed by several independent crypto experts.

5.2.3 Licensing

The CSA3 Descrambling System is licensed to manufacturers of decoders and their components, and to providers, designers and other entities engaged in conditional access. The CSA3 Scrambling Technology is licensed to manufacturers of scramblers, who in turn sublicense to the purchasers of scramblers.

CSA3 is made available by the custodian upon signature of a non-disclosure agreement and provided potential users are bone fide. The custodian is ETSI itself and for information can be obtained by contacting:

European Telecommunications Standards Institute (ETSI)

Algorithms and Codes service
650 Route de Lucioles
F-06921 Sophia Antipolis Cedex
FRANCE
Tel.: +33 4 92 94 42 16
Fax: +33 4 92 94 42 70

5.2.4 CSA-3 for DVB IPTV Scrambling

For the purpose of DVB IPTV scrambling, CSA3 shall only be used in standard mode as specified in TS 100 289 [3].

5.3 DVB Common Scrambling Algorithm version 1

5.3.1 Introduction

CSA Version 1 (CSA1) is a device, apparatus or mechanism (whether implemented as hardware or software) designed or specifically adapted, totally or partially, to render unintelligible a service compatible with Standards by the use of CSA1 Scrambling Technology and any modifications and improvements thereof and which can be descrambled using a common descrambling system in the form approved by the Steering Board of the DVB Project for Standards.

CSA1 was approved by the Steering Board of the DVB Project in May 1994, and was published as European Telecommunications Standards Institute (ETSI) Technical Report ETR 289 [i.2] in October 1996. It is comprised of the Common Descrambling System and Scrambling Technology. The specification for each is distributed separately under arrangements with the European Telecommunications Standards Institute (ETSI), which acts as custodian for the four Companies which have developed the Common Scrambling Algorithm.
5.3.2 Technical details

The CSA1 Scrambling Algorithm has been designed to minimise the likelihood of piracy attack over a long period of time (at least 10 years from inception) and thus contains highly security sensitive information. The technical details of the scrambling algorithm can only be made available to bona-fide users upon signature of a Non-Disclosure Agreement (NDA) administered by ETSI.

The scrambling algorithm operates on the payload of a Transport Stream (TS) packet in the case of TS-level scrambling. A structuring of Packetized Elementary Stream (PES) packets is used to implement PES-level scrambling with the same scrambling algorithm. CSA1 uses a 64 bit key (Control Word) to encrypt and decrypt data blocks of any size over 8 bytes (with a granularity of 1 byte).

5.3.3 Licensing

CSA1 is licensed to manufacturers of decoders and their components, and to providers, designers and other entities engaged in conditional access.

The Scrambling Technology is licensed to manufacturers of scramblers, who in turn sublicense to the purchasers of scramblers.

CSA1 is made available by the custodian upon signature of a Non-Disclosure Agreement and provided potential users are bone fide. The custodian is ETSI itself and information can be obtained by contacting:

European Telecommunications Standards Institute (ETSI)
Algorithms and Codes service
650 Route de Lucioles
F-06921 Sophia Antipolis Cedex
FRANCE
Tel.: +33 4 92 94 42 16
Fax: +33 4 92 94 42 70

5.3.4 CSA for IPTV scrambling

CSA-1 was designed in 1994 to provide adequate security for a period of at least ten years. In 2007, DVB adopted CSA-3 in order to continue providing an adequate level of security taking into account advances in technology.

CSA-1 is not expected to provide adequate security for another ten years and is not recommended to be used in new IPTV deployments, except for the purposes of backwards compatibility and legacy support.

6 Common IPTV Software-oriented Scrambling Algorithm (CISSA) Version 1

6.1 Introduction

This clause describes the Common IPTV Software-oriented Scrambling Algorithm (CISSA) version 1, which is the software-friendly scrambling algorithm which shall be used for MPEG2-TS IPTV services. CISSA is an algorithm for which the descrambling method is software-friendly. In other words, it is suitable for efficient execution in software on general purpose CPUs, and is also possible to implement efficiently in hardware.

This section comprises an informative general description indicating how CISSA is used to encrypt/decrypt an MPEG2 Transport Stream, followed by a normative section giving a detailed formal description of the algorithm.

6.2 General Description (informative)

The Common IPTV Software-oriented Scrambling Algorithm (CISSA) uses the AES cipher described in [1] as its basic building block, with 128-bit (16 bytes) keys, called Control Words (CWs).
6.2.1 TS level scrambling

The basic unit of scrambling (or descrambling) is an MPEG-2 transport stream packet (TS). Each packet is scrambled independently from all others, allowing random access. A packet consists of a header and optional adaptation field that are left in the clear followed by a payload that is processed as described below.

AES-128, as defined in NIST FIPS 197 [1], is used to encrypt blocks of 16 bytes of the data payload of a given TS packet. The payload, in general, is not an integer multiple of 16 bytes. Only a part that is made up of a multiple of 16 bytes, is encrypted by the scrambling process. If there is a remaining part (up to 15 bytes), it stays in the clear. If the payload is less than 16 bytes, it is left in the clear.

An integer number of 16 byte contiguous blocks of the payload is encrypted using a Cipher Block Chaining (CBC) technique as described in NIST SP800-38A [2]. The Initialization Vector (IV) is a constant.

When the payload size is not a multiple of 16, the remaining part of the payload that stays in the clear is located at the end of the TS packet as shown in figure 1.

![Figure 1: TS Scrambled Packet Overview](image)

6.2.2 PES level scrambling

The PES level scrambling method requires that the PES packet header is not scrambled (as required in ISO/IEC 13818-1 [i.1]) and TS packets containing parts of a scrambled PES packet do not contain an Adaptation Field (with the exception of the TS packet containing the end of the PES packet). The header of a scrambled PES packet is required not to span multiple TS packets. The TS packet carrying the start of a scrambled PES packet is filled by the PES header and the first part of the PES packet payload. In this way, the first part of the PES packet payload is scrambled exactly as a TS packet with a similar size payload. The remaining part of the PES packet payload is split in super-blocks of 184 bytes. Each superblock is scrambled exactly as a TS packet payload of 184 bytes. The end of the PES packet payload is aligned with the end of the TS packet (as required in ISO/IEC 13818-1 [i.1]) by inserting at the beginning an Adaptation Field of suitable size. If the length of the PES packet is not a multiple of 184 bytes, the last part of the PES packet payload (from 1 to 183 bytes) is scrambled exactly as a TS packet with a similar size payload. A schematic diagram describing the mapping of scrambled PES packets into TS packets is given in figure 2.

![Figure 2: PES Scrambled Packet Overview](image)
For applications that scramble MPEG-2 Sections, a problem occurs as the MPEG-2 specified syntax does not include any scrambling control bits. Therefore, the scrambling of Sections shall be at the TS level and shall be signalled by the scrambling control field bits. Clear and scrambled Sections cannot be combined in a single TS packet. The MPEG-2 defined padding mechanism can be used to create TS packets with only clear or only scrambled Sections. This means that the end of a TS packet carrying a Section shall be filled with bytes having a value of 0xFF, in order to separate clear and scrambled Sections into different TS packets.

In DVB CISSA context, PES level scrambling is expected to be used only for service-driven professional applications and is not intended to be used in Consumer Electronics applications.

6.3 Normative Elements

This clause provides the normative description of CISSA version 1. Notations are as defined in clause 3.2.

6.3.1 Encryption Elements

6.3.1.1 Block Cipher

AES-128, as defined in NIST FIPS 197 [1], shall be used as underlying block cipher for CISSA.

6.3.1.2 Initialization Vector

The Initialization Vector shall have the following value:

- \( IV = 0x445642544d43505441455343495341 \)

6.3.1.3 Chaining Mode

CBC, as defined in NIST SP800-38A [2], shall be used as chaining mode.

6.3.2 TS packet Scrambling and Descrambling

Each TS packet shall be processed separately.

The TS packet header and adaptation field, if present, shall be left in the clear.

The payload size (\( \text{payload\_size} \)) and the encrypted payload size (\( \text{encrypted\_payload\_size} \)) shall be computed as follows:

- \( \text{payload\_size} = 188 - (\text{header\_size} + \text{adaptation\_field\_size}) \)

- \( \text{encrypted\_payload\_size} = \text{payload\_size} - \lfloor \text{payload\_size} \mod 16 \rfloor \)

where \( \text{header\_size} \) is the size of the TS packet header in bytes, and \( \text{adaptation\_field\_size} \) is the size of the adaptation field in bytes.

The \( \text{encrypted\_payload\_size} \) bytes immediately following the adaptation field (or, if there is no adaptation field, following the TS packet header) shall then be encrypted or decrypted using the encryption elements as defined in clause 6.3.1.

Any remaining byte shall be left in the clear.

EXAMPLE 1: If there is no adaptation field; \( \text{encrypted\_payload\_size} \) is 176 bytes and there will be 8 bytes at the end of the TS packet that will remain in the clear.

EXAMPLE 2: If the adaptation field size is 17, \( \text{encrypted\_payload\_size} \) is 160 bytes and there will be 7 bytes at the end of the TS packet that will remain in the clear.

EXAMPLE 3: If the adaptation field size is 24, \( \text{encrypted\_payload\_size} \) is 160 bytes and there will be no bytes left in the clear at the end of the TS packet.
EXAMPLE 4: If the adaptation field size is 169, there will be no encrypted payload and the 15 bytes of the entire payload will remain in the clear.

6.3.3 PES level Scrambling and Descrambling

In order to apply PES level scrambling or descrambling, the following conditions shall be verified:

- Scrambling shall only occur at one level (TS or PES) and is not allowed to occur at both levels simultaneously.
- The header of a scrambled PES packet shall not exceed 184 bytes.
- The TS packets carrying parts of a scrambled PES packet, shall not have Adaptation fields with the exception of TS packets containing the end of a PES packet. The TS packet carrying the end of a scrambled PES packet, may carry an Adaptation Field to align of the end of the PES packet with the end of the TS packet.

NOTE: These recommendations clearly do not apply to unscrambled PES packets or in the case of TS-level scrambling.

Each PES packet shall be processed independently.

The PES packet header shall be left in the clear.

The payload size (payload_size) and the encrypted payload size (encrypted_payload_size) shall be computed as follows:

- For the TS packet carrying the PES header:
  - payload_size = 184 - pes_header_size
  - encrypted_payload_size = payload_size - [payload_size mod 16]
  where pes_header_size is the size of the PES packet header in bytes.

- For following TS packets, excepted the TS packet carrying the end of the PES packet:
  - payload_size = 184
  - encrypted_payload_size = 176

- For the TS packet carrying the end of the PES packet:
  - payload_size = pes_tail_size
  - encrypted_payload_size = payload_size - [payload_size mod 16]
  where pes_tail_size is the size of end of the PES packet.

If pes_tail_size is not 184, an Adaptation Field of size 184 - pes_tail_size shall be added at the beginning of the TS packet carrying the end of the PES packet, where pes_tail_size is the size of end of the PES packet.

The encrypted_payload_size bytes that:

- immediately follow the PES header in the TS packet carrying the PES header;
- immediately follow the TS header in following TS packets, excepted the TS packet carrying the end of the PES packet;
- immediately follow the TS header, or the Adaptation Field if any, in the TS packet carrying the end of the PES packet;

shall then be encrypted or decrypted using the encryption elements as defined in clause 6.3.1.

Any other byte shall be left in the clear.
7 Signalling

7.1 Scrambling descriptor (informative)

This clause explains the selection mechanism as defined in EN 300 468 [4] for the transport of signalling information to indicate which DVB IPTV scrambling algorithm is used.

The scrambling descriptor indicates the selected mode of operation for the scrambling system. As defined in EN 300 468 [4], it is located in the program map section at the program info loop level.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Number of bits</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>scrambling_descriptor()</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>descriptor_tag</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>descriptor_length</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>scrambling_mode</td>
<td>8</td>
<td>uimsbf</td>
</tr>
</tbody>
</table>

Semantics for the scrambling_descriptor:

scrambling_mode: This 8-bit field identifies the selected mode of the scrambling algorithm (see table 2).

<table>
<thead>
<tr>
<th>scrambling_mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>reserved for future use</td>
</tr>
<tr>
<td>0x01</td>
<td>reserved for other uses (see EN 300 468 [4])</td>
</tr>
<tr>
<td>0x02</td>
<td>This value indicates use of DVB-CSA1</td>
</tr>
<tr>
<td>0x03</td>
<td>This value indicates use of DVB-CSA3 in standard mode.</td>
</tr>
<tr>
<td>0x04</td>
<td>reserved for other uses (see EN 300 468 [4])</td>
</tr>
<tr>
<td>0x05</td>
<td>reserved for other uses (see EN 300 468 [4])</td>
</tr>
<tr>
<td>0x06 to 0x0F</td>
<td>reserved for future use</td>
</tr>
<tr>
<td>0x10</td>
<td>This value indicates use of DVB-CISSA version 1</td>
</tr>
<tr>
<td>0x11 to 0x1F</td>
<td>These values are reserved for future DVB-CISSA versions</td>
</tr>
<tr>
<td>0x20 to 0x6F</td>
<td>reserved for future use</td>
</tr>
<tr>
<td>0x70 to 0x7F</td>
<td>reserved for other uses (see EN 300 468 [4])</td>
</tr>
<tr>
<td>0x80 to 0xFE</td>
<td>user defined</td>
</tr>
<tr>
<td>0xFF</td>
<td>reserved for future use</td>
</tr>
</tbody>
</table>

NOTE: Possible usages of scrambling descriptor are further described in annex E of EN 300 468 [4].

7.2 Use of the scrambling descriptor

For the purposes of the present document, only values 0x02, 0x03, and any value from the range for DVB-CISSA (0x10 to 0x1F) shall be used.

NOTE: As specified in EN 300 468 [4], other scramblers that are not listed in table 2 can be used and signalled by user defined scrambler descriptors (in the range 0x80 - 0xFE).
Annex A (informative):
CISSA Implementation Guidelines

The software-oriented scrambling mechanism specified in this document is recommended to be implemented in such a way that attacks against this mechanism are extremely unlikely to be successful.

Attacks that are recommended to be considered include:

- Side-channel attacks which consist of monitoring parameters (e.g. elapsed time, power consumption…) that are external to the system when the algorithm is running. An observation may reveal secret data or secret temporary values.
- Attacks which consist of monitoring parameters (e.g. cache memory) that form an internal part of the system when the algorithm is running. An observation may reveal secret data or secret temporary values.
- Bugs exploitation (e.g. triggering a buffer overflow may reveal some secret data).
- Code modifications (for software implementation).
- Fault injection attacks.

The above list is not exhaustive. In addition, other types of attacks are likely to appear after the publication of the present document.
Annex B (informative):
CISSA Test Vectors for TS level scrambling

This clause describes 4 test cases for the MPEG-2 Transport Stream adaptation of the CISSA.

All TS packets are scrambled using the following Control Word:

\[
00 \ 11 \ 22 \ 33 \ 44 \ 55 \ 66 \ 77 \ 88 \ 99 \ aa \ bb \ cc \ dd \ ee \ ff
\]

NOTE 1: These test vectors show the action of the scrambler on the TS packet. This includes both encryption of the payload and change of `transport_scrambling_control` bits as specified in TS 100 289 [3].

NOTE 2: The normative part of the present document only describes the encryption of the payload.

<table>
<thead>
<tr>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx xx xx</td>
</tr>
<tr>
<td>xx xx xx</td>
</tr>
<tr>
<td>xx xx xx</td>
</tr>
<tr>
<td>xx xx xx</td>
</tr>
</tbody>
</table>
## Test Case 1: No Adaptation Field

<table>
<thead>
<tr>
<th>Clear Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>47 60 80 11</td>
</tr>
<tr>
<td>70 61 79 6c</td>
</tr>
<tr>
<td>20 63 72 65</td>
</tr>
<tr>
<td>73 74 20 76</td>
</tr>
<tr>
<td>68 65 20 44</td>
</tr>
<tr>
<td>6d 62 6c 65</td>
</tr>
<tr>
<td>72 2e 20 54</td>
</tr>
<tr>
<td>61 79 6c 6f</td>
</tr>
<tr>
<td>63 72 65 61</td>
</tr>
<tr>
<td>74 75 73 65</td>
</tr>
<tr>
<td>6d 74 76 65</td>
</tr>
<tr>
<td>65 20 44 56</td>
</tr>
<tr>
<td>62 6c 65 72</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scrambled Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>47 60 80 91</td>
</tr>
<tr>
<td>7a 4a d1 20</td>
</tr>
<tr>
<td>19 fa ee 10</td>
</tr>
<tr>
<td>00 a5 2c ec</td>
</tr>
<tr>
<td>e0 41 cb c3</td>
</tr>
<tr>
<td>82 04 ba d7</td>
</tr>
<tr>
<td>75 55 97 20</td>
</tr>
<tr>
<td>c1 9d 44 81</td>
</tr>
<tr>
<td>af 45 b2 5f</td>
</tr>
<tr>
<td>f4 34 6a 1b</td>
</tr>
<tr>
<td>10 a8 e0 b2</td>
</tr>
<tr>
<td><strong>Test Case 2: 7-byte Adaptation Field</strong></td>
</tr>
<tr>
<td>-----------------------------------------</td>
</tr>
<tr>
<td><strong>Clear Packet</strong></td>
</tr>
<tr>
<td>[47 60 80 31] 06 00 FF FF FF FF FF FF 54 68 69 73 20</td>
</tr>
<tr>
<td>69 73 20 74 68 65 20 70 61 79 6c 6f 61 64 20 75</td>
</tr>
<tr>
<td>73 65 64 20 66 6f 72 20 63 72 65 61 74 69 6e 67</td>
</tr>
<tr>
<td>20 74 68 65 20 74 65 73 74 20 76 65 63 74 6f 72</td>
</tr>
<tr>
<td>73 20 66 6f 72 20 74 68 65 20 44 56 42 20 49 50</td>
</tr>
<tr>
<td>54 56 20 73 63 72 61 6d 62 6c 65 72 6f 72 65 73</td>
</tr>
<tr>
<td>63 72 61 6d 62 6c 65 72 2e 20 54 68 65 20 70 61 79 6c 6f 61 64 20 75 73</td>
</tr>
<tr>
<td>65 64 20 66 6f 72 20 63 72 65 61 74 69 6e 67 20</td>
</tr>
<tr>
<td>74 68 65 20 74 65 73 74 20 76 65 63 74 6f 72 73</td>
</tr>
<tr>
<td>20 66 6f 72 20 74 68 65 20 44 56 42 20 49 50 54</td>
</tr>
<tr>
<td>56 20 73 63 72 61 6d 62 6c 65 72 2f</td>
</tr>
<tr>
<td><strong>Scrambled Packet</strong></td>
</tr>
<tr>
<td>[47 60 80 b1] 06 00 FF FF FF FF FF FF 15 ce 67 e0 cb</td>
</tr>
<tr>
<td>01 b5 3c e7 60 54 e5 7a 4a d1 2a 0a df a4 ea aa</td>
</tr>
<tr>
<td>e9 32 c6 78 3f 51 ae 19 fa ee 10 8b db 78 f3 11</td>
</tr>
<tr>
<td>3e c2 b5 72 cc 20 85 00 a5 2c ec a1 14 12 6c 58</td>
</tr>
<tr>
<td>24 4d f5 63 e7 a9 b4 e0 41 cb c3 fb ff ff d8 3c</td>
</tr>
<tr>
<td>8f bf fb 10 e8 3e a3 82 04 ba d7 02 fb 01 a2 7b</td>
</tr>
<tr>
<td>62 2c 4f 85 aa b6 aa 75 55 97 20 d6 5a b8 44 ce</td>
</tr>
<tr>
<td>a2 8c f2 e1 fe 5e 7a c1 9d 4d 81 89 19 c2 32 49</td>
</tr>
<tr>
<td>f1 40 75 7b 5d 16 c0 af 45 b2 5f 50 9b 9d a0 61</td>
</tr>
<tr>
<td>97 12 c5 9f 0b 39 b0 6f 1f be 90 12 3f 21 29 83</td>
</tr>
<tr>
<td>93 6a 95 31 7f cb 62 f4 34 6a 1b 1e 16 48 40 30</td>
</tr>
<tr>
<td>3a ff 83 8a 01 9b f8 10 a8 e0 b2 2f</td>
</tr>
</tbody>
</table>
### Test Case 3: 8-byte Adaptation Field

<table>
<thead>
<tr>
<th>Clear Packet</th>
<th>Scrambled Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Clear Packet" /></td>
<td><img src="image" alt="Scrambled Packet" /></td>
</tr>
</tbody>
</table>
## Test Case 4: 9-byte Adaptation Field

<table>
<thead>
<tr>
<th>Clear Packet</th>
<th>Scrambled Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>47 60 80 31 08 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 69</td>
<td>47 60 80 b1 08 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 67</td>
</tr>
<tr>
<td>73 20 69 73 20 74 68 65 20 70 61 79 6c 6f 61 64</td>
<td>e0 cb 01 b5 3c e7 60 54 e5 7a 4a d1 20 a0 df a4</td>
</tr>
<tr>
<td>20 75 73 65 64 20 66 6f 72 20 74 68 65 20 44 56 42 20</td>
<td>ea aa e9 32 c6 78 3f 51 19 fa ee 10 8b db 78</td>
</tr>
<tr>
<td>6e 67 20 74 68 65 20 74 65 73 74 20 76 65 63 74</td>
<td>f3 11 3e c2 b5 72 cc 20 85 00 a5 2c ec a1 14 12</td>
</tr>
<tr>
<td>6f 72 73 20 66 6f 72 20 74 68 65 20 44 56 42 20 49</td>
<td>6c 58 24 4d f5 63 e7 a9 b4 e0 41 cb c3 fb ff fb</td>
</tr>
<tr>
<td>49 50 54 56 20 73 63 72 61 6d 62 6c 65 72 2e 20 54 68 69 73</td>
<td>d8 3c 8f b1 f0 10 e8 3e a3 82 04 ba d7 02 fb 01</td>
</tr>
<tr>
<td>20 69 73 20 74 68 65 20 70 61 79 6c 6f 61 64 20</td>
<td>a2 7b 62 2c 4f 85 aa b6 aa 75 55 97 20 d6 5a b8</td>
</tr>
<tr>
<td>75 73 65 64 20 66 6f 72 20 74 68 65 20 74 65 73 74 20 76 65 63 74 6f</td>
<td>44 ce a2 8c f2 e1 fe 5e 7a c1 9d 44 81 89 19 c2</td>
</tr>
<tr>
<td>72 73 20 66 6f 72 20 74 68 65 20 44 56 42 20 49</td>
<td>32 49 f1 40 75 7b 5d 16 c0 af 45 b2 5f 50 9b 9d</td>
</tr>
<tr>
<td>60 54 56 20 73 63 72 61 6d 62 6c 65</td>
<td>a0 61 97 12 c5 9f 0b 39 b0 6f 1f be 90 12 3f 21</td>
</tr>
<tr>
<td></td>
<td>29 83 93 6a 95 31 7f cb 62 f4 34 6a 1b 42 20 49</td>
</tr>
<tr>
<td></td>
<td>50 54 56 20 73 63 72 61 6d 62 6c 65</td>
</tr>
</tbody>
</table>
## History

<table>
<thead>
<tr>
<th>Document history</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>V1.1.1</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>