An introduction to MPEG transport streams

all you should know before using TSDuck

Agenda

- Transport streams
  - packets, sections, tables, PES, demux
- DVB SimulCrypt
  - architecture, synchronization, ECM, EMM, scrambling
- Standards
  - MPEG, DVB, others

transport streams

packets and packetization

Standard key terms

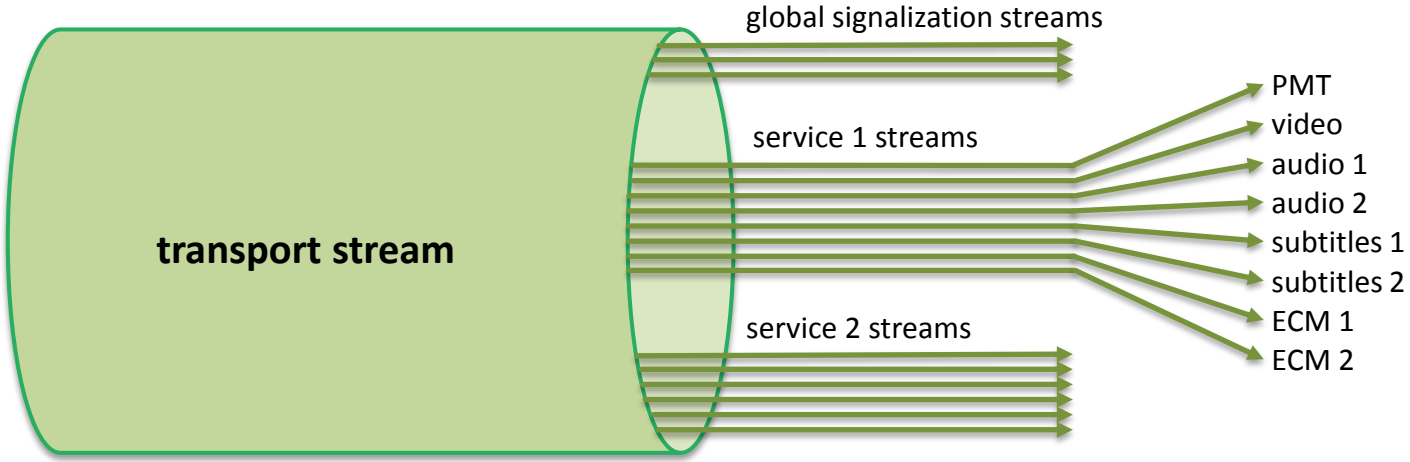
- Service / Program
  - DVB term : service
  - MPEG term : program
  - TV channel (video and / or audio)
  - data service (software download, application data)
- Transport stream
  - aka. « TS », « multiplex », « transponder »
  - continuous bitstream
  - modulated and transmitted using one given frequency
  - aggregate several services
- Signalization
  - set of data structures in a transport stream
  - describes the structure of transport streams and services

MPEG-2 transport stream

- Structure of MPEG-2 TS defined in ISO/IEC 13818-1
- One operator uses several TS
- TS = synchronous stream of 188-byte TS packets
  - 4-byte header
  - optional « adaptation field », a kind of extended header
  - payload, up to 184 bytes
- Multiplex of up to 8192 independent elementary streams (ES)
  - each ES is identified by a Packet Identifier (PID)
  - each TS packet belongs to a PID, 13-bit PID in packet header
  - smooth muxing is complex, demuxing is trivial
- Two types of ES content
  - PES, Packetized Elementary Stream : audio, video, subtitles, teletext
  - sections : data structures

Multiplex of elementary streams

- A transport stream is a multiplex of elementary streams
  - elementary stream = sequence of TS packets with same PID value in header
  - one set of elementary streams for global signalization
    - describe the TS, the network, the operator, the services, the events, EMM's, etc.
  - one set of elementary streams per service
    - a service is typically a TV channel



TS packet

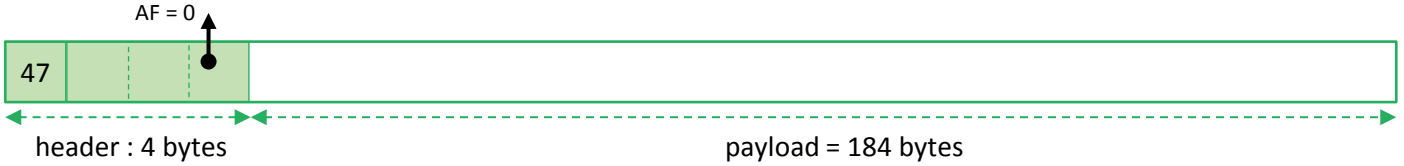
4-byte header includes :

- Sync byte = 0x47
- PID : 13 bits
- Continuity counter : 4 bits
- Payload Unit Start Indicator (PUSI) : 1 bit
- Transport scrambling control : 2 bits
- Adaptation field presence : 1 bit
- Payload presence : 1 bit
- More…

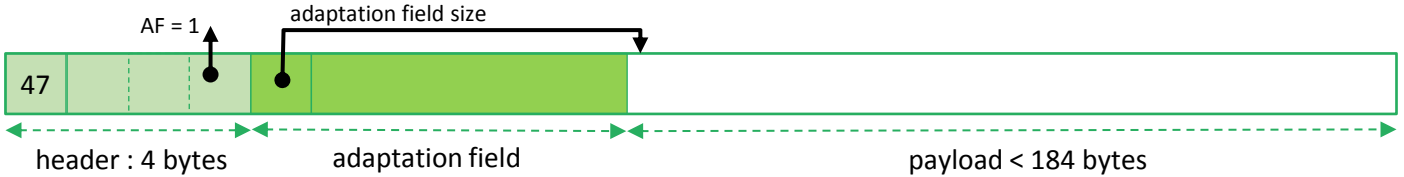
Adaptation field may include :

- Program Clock Reference (PCR / OPCR)
- Private data
- Stuffing (for PES stream padding)
- More…

TS packet without adaptation field

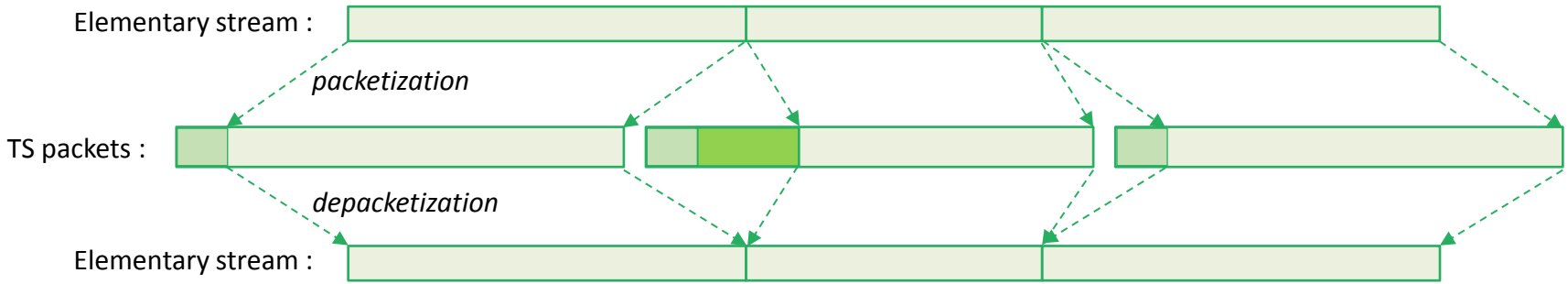


TS packet with adaptation field



Multiplexing and demultiplexing

- Elementary stream = concatenation of all payloads of all TS packets with same PID
- Elementary stream transport
  - packetization = cutting ES into packets payloads with same PID
    - setting Payload Unit Start Indicator (PUSI) in TS header on « unit » boundary
  - multiplexing = mixing with packets from other PID's to build a complete TS
  - demultiplexing = extracting all packets with same PID from TS
  - depacketization = rebuilding ES from packets payloads with same PID
    - using PUSI to resynchronize on « unit » boundary

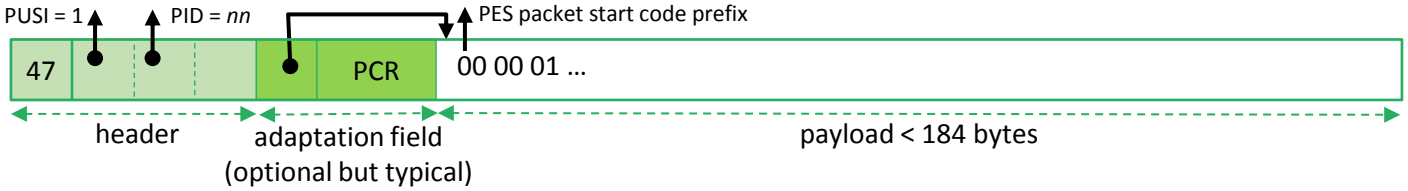


Packetized Elementary Stream (PES)

- A stream of PES packets
  - up to 65536 bytes per PES packet
  - start of PES packet identified by PUSI bit in TS header
- PES packets can contain
  - video : MPEG-2 (H.262), AVC (H.264), HEVC (H.265), etc.
  - audio : MPEG-2 Layer 2, AAC, HE-AAC, AC-3, DTS, DTS-HD, etc.
  - DVB subtitles (text or bitmap)
  - teletext (deprecated but still used)
- One elementary stream contains one single type of content
  - video
  - audio for one language (with or without « audio description »)
    - multi-channel audio (stereo, 5+1, etc.) within same PID
  - subtitles for one language (with or without « for hard of hearing »)
  - exception : one teletext stream is a multiplex of several text streams (« pages »)

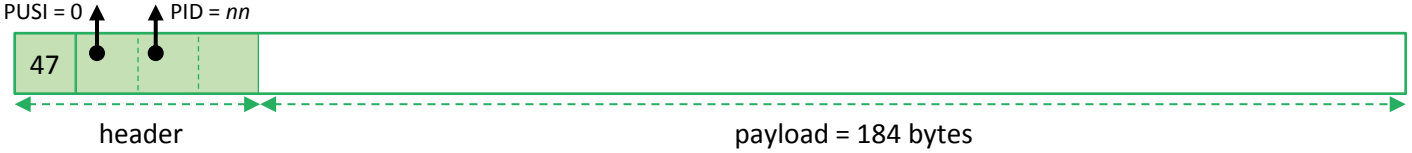
Typical PES packetization

First TS packet for PES packet

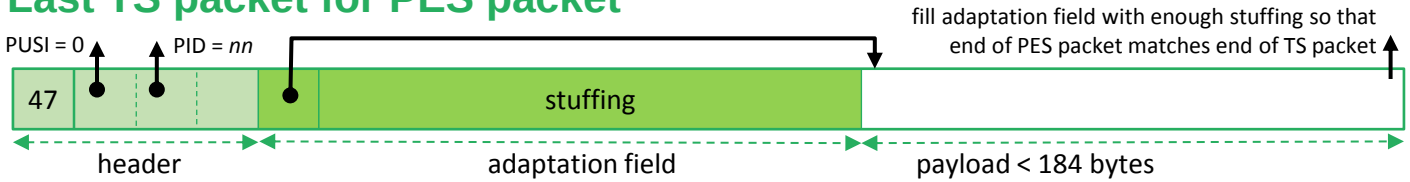


As many intermediate TS packets as required for current PES packet

(multiplexed with TS packets from others PID's)



Last TS packet for PES packet



PES streams robustness

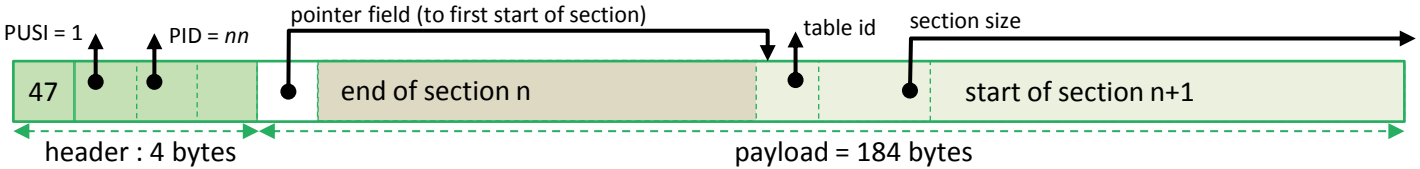
- TS packet loss is tolerated in audio and video streams
  - video « macro-block » effect
  - audio « glitch » effect
  - quality of recovery based on decoder implementation
- TS packet loss detection based on *continuity_counter*
  - 4-bit field in TS packet header
  - cannot detect loss of an exact multiple of 16 TS packets
  - resynchronization on next TS packet with PUSI
- But video / audio decoders can resynchronize within PES packet
  - video / audio bitstream formats usually contain synchronization patterns
  - example : NAL unit boundary in AVC encoding

Sections streams

- Contain data structures named « tables »
- A table is split into one or more « sections »
  - section = smallest data unit, up to 4096 bytes
  - standard header and type-specific payload
  - table type identified by *table_id* in header
  - two types of section syntax : « short » and « long »
    - based on 1 bit in header
- Each type of table defines its own syntax
  - use long or short sections
  - payload bitstream syntax
- Descriptor
  - standard substructure with standard header and type-specific payload
  - most tables use generic « lists of descriptors »

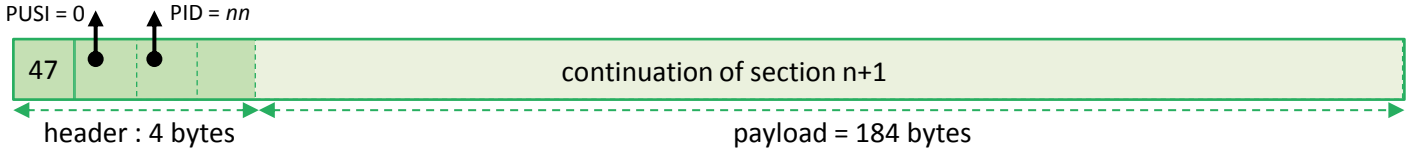
Typical section packetization

TS packet containing the start of section n+1

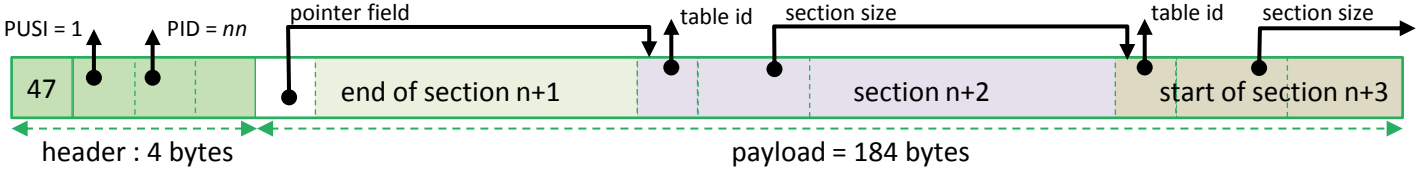


As many intermediate TS packets as required for section n+1

(multiplexed with TS packets from others PID's)



Last TS packet for section n+1, start of next section



Tables with short section

- One section per table
  - section and table are equivalent
- Each table brings new information
  - CAS EMM / ECM
  - date and time information (TDT / TOT)
- No standard integrity check
  - except section length in section header
  - some table-specific mechanisms
    - cryptographic integrity in EMM / ECM
    - CRC32 in TOT

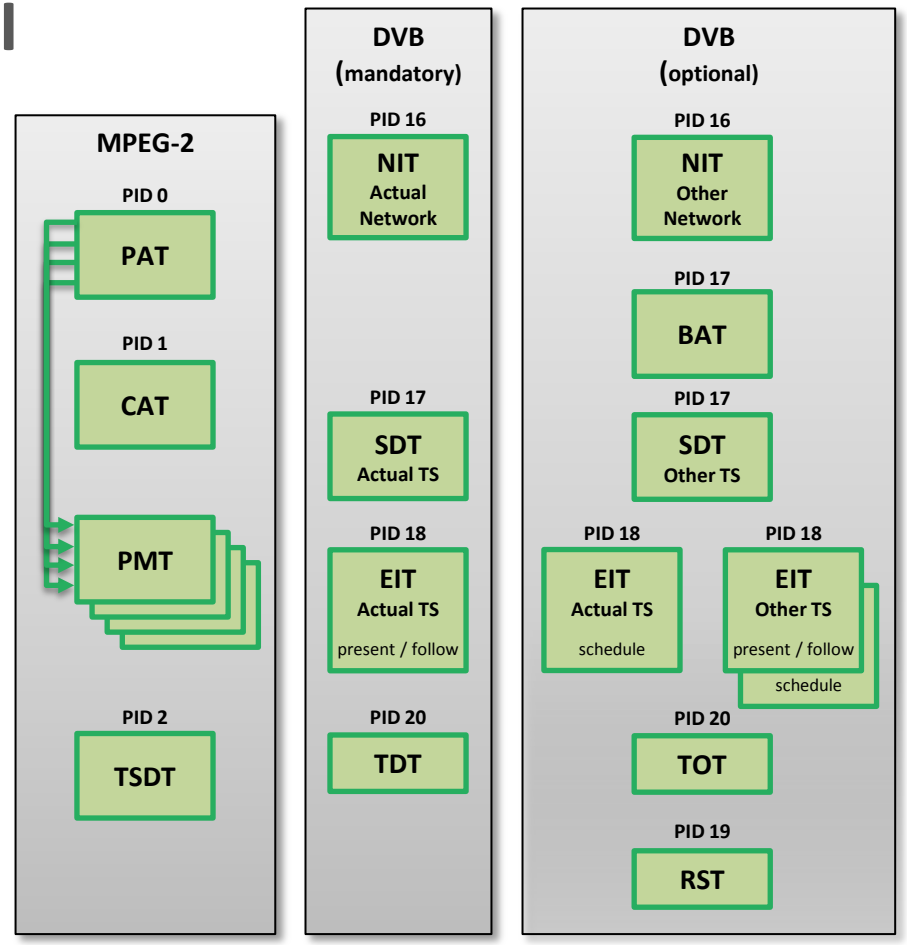
Tables with long sections

- Up to 256 sections per table
  - need to receive all sections to rebuild the complete table
- Same table repeatedly cycled
- Content change notification
  - version number in long section header
  - each table is repeatedly broadcast with same version number
  - version number changes when table content changes
  - STB software sets demux filters to be notified of new tables only
- Integrity check
  - CRC32 in each section
  - section rejected in case of corruption, can be detected at demux level
  - resynchronization on next TS packet with PUSI

Signalization: PSI / SI

- PSI : Program Specific Info.
  - MPEG-defined
  - ISO / IEC 13818-1
  - TS structure: PAT, PMT
  - CA : CAT
- SI : Service Information
  - DVB-defined
  - ETSI EN 300 468
  - private sections in MPEG terms

Extracted from
DVB standard
ETSI EN 300 468



MPEG-defined PSI

- PAT : Program Association Table
  - repeated in PID 0
  - list of « services » in the TS, ie. TV channels or data channels
  - service id and PMT PID
- PMT : Program Map Table
  - technical description of one service
  - list of elementary streams in the service
    - PID, type (audio, video, etc.), additional info using a list of descriptors
  - list of ECM streams for this service
- CAT : Conditional Access Table
  - repeated in PID 1
  - list of EMM streams on this TS
  - CAT not present when no EMM on TS

DVB-defined SI (1/2)

- SDT : Service Description Table
  - editorial description of the services in a TS
  - either in « actual » TS or « other » TS
  - service names and ancillary services
- BAT : Bouquet Association Table
  - commercial operator description and services
  - several commercial operators may sell the same services
- NIT : Network Information Table
  - technical description of a network
  - either « actual » network or « other » network
  - list of TS in this network
    - usually with frequency and tuning parameters
    - used for fast network scanning
  - list of services in each TS
    - service ids and « logical channel number »

DVB-defined SI (2/2)

- EIT : Event Information Table
  - editorial description of events
  - either in « actual » TS or « other » TS
  - EIT « present / following »
    - short description of current and next event on each service
    - used to display information banner on screen
  - EIT « schedule »
    - long description of all events in the forthcoming days
    - used to display the EPG
    - optional, depends on operator’s good will and bandwidth availability
    - complete 7-day EPG for a large operator uses several Mb/s
    - sparse EIT schedule sections, rarely complete tables
- TDT / TOT : Time and Date Table / Time Offset Table
  - current date and time, UTC (TDT) and local offset by region (TOT)
  - used to synchronize STB system time
  - typically one table every 10 to 30 seconds only

DVB SimulCrypt

one network, several conditional access systems

Standard key terms

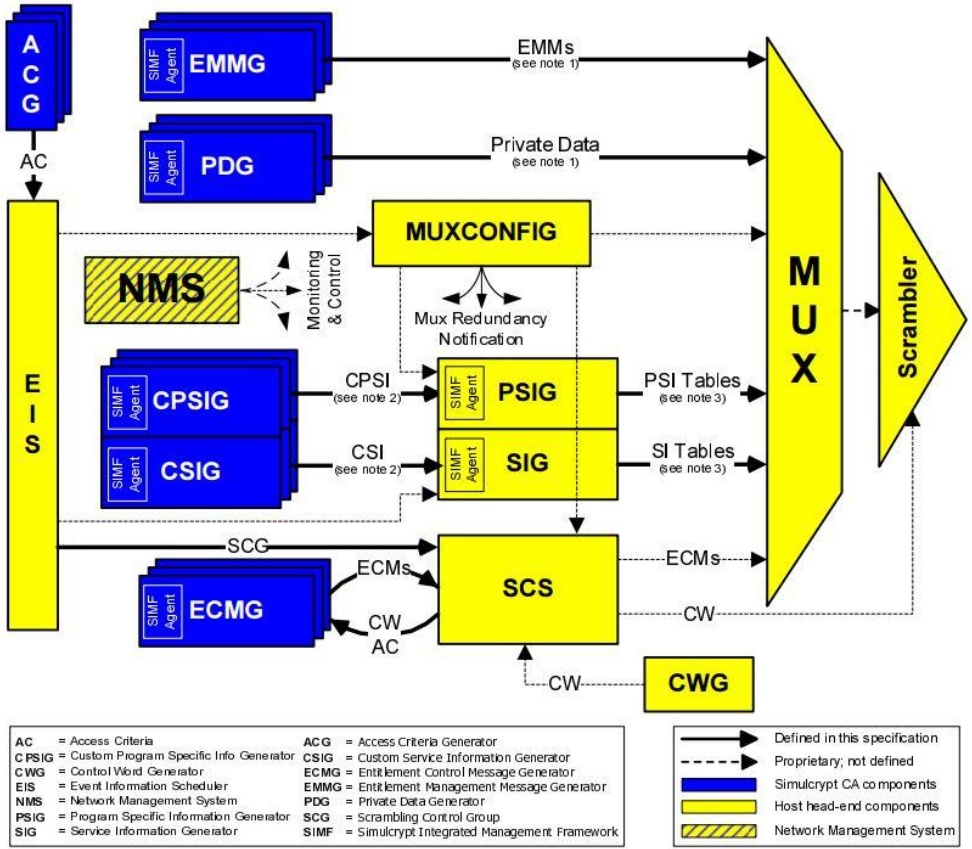
- CAS : Conditional Access System
- CW : Control Word
  - content encryption key for video & audio
- EMM : Entitlement Management Message
  - CAS-specific message to manage rights, smartcards, subscribers
  - sent to some identified set of subscribers, possibly only one
- ECM : Entitlement Control Message
  - CAS-specific message to control a scrambled service
  - sent to everyone willing to watch the service

DVB SimulCrypt

- Enforce coexistence of multiple CAS to protect the same content
  - DVB-defined standard
- Use-cases
  - one broadcast operator, multiple commercial operators
  - transition between CAS generations
- Broadcast
  - very simple architecture
  - common scrambling
  - multiple EMM and ECM streams with standard signalization
- Head-end
  - complex architecture
  - multiple CAS equipment
  - common synchronization

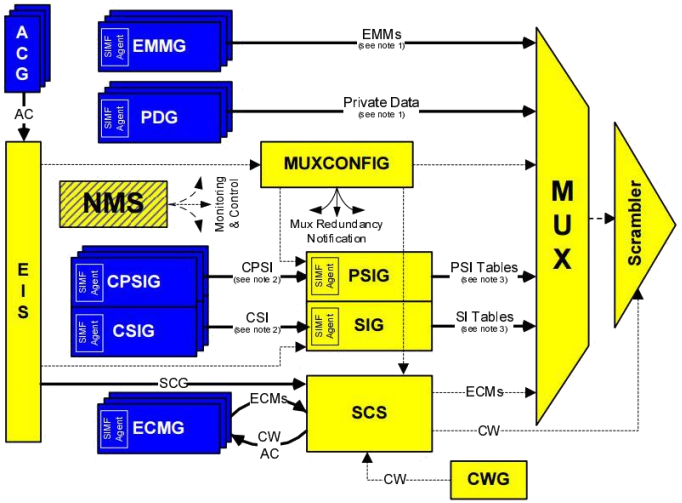
DVB SimulCrypt head-end diagram

Extracted from
DVB standard
ETSI TS 103 197



DVB SimulCrypt head-end

- Interface between two worlds
  - one « MUX system » vendor
    - yellow components
  - multiple CAS vendors
    - blue components
- DVB SimulCrypt protocols
  - specified between components of distinct worlds
  - protocols within the same world are not specified
    - proprietary, vendor specific
  - consistent nested tag-length-value (TLV) structures
    - using logical « channels » and « streams »
    - except ACG ⇔ EIS protocol (XML protocol)
  - EIS ⇔ SCS protocol is specified
    - so that EIS and SCS may in fact come from distinct vendors
- TSDuck plugins
  - *scrambler* interacts with any standard ECMG
  - *datainject* interacts with any standard EMMG or PDG



EMM signalization

- Using CA_descriptor in the CAT of the TS
  - standard part of CA_descriptor: CA system id, EMM PID
    - CA_system_id are allocated by DVB
    - http://www.dvbservices.com/identifiers/ca_system_id
  - private part of CA_descriptor: CAS-specific
    - used by the CA software in the STB
- Number of EMM streams is CAS-specific
  - for instance, one EMM stream may contain all EMM's for
    - one operator
    - one EMM type (e.g. individual, group, global)
    - or any other configuration
    - when they exist, operator id and EMM types are CAS-specific concepts
    - they are usually identified in the private part of the CA_descriptor

ECM broadcast

- An ECM usually transports a CW pair and access criteria
  - specific to one or more audio or video streams
  - specific to one CAS
- Each service (i.e. channel) has dedicated ECM streams
  - per scrambling group
  - per CAS
    - base mechanism for DVB SimulCrypt
- Scrambling group
  - a set of audio or video elementary streams scrambled with the same CW
  - subtitles are usually not scrambled in practice (but could be in theory)
  - usually, all audio and video streams of a service are in the same scrambling group
  - in rare cases, audio and video streams are scrambled with distinct CW

ECM signalization

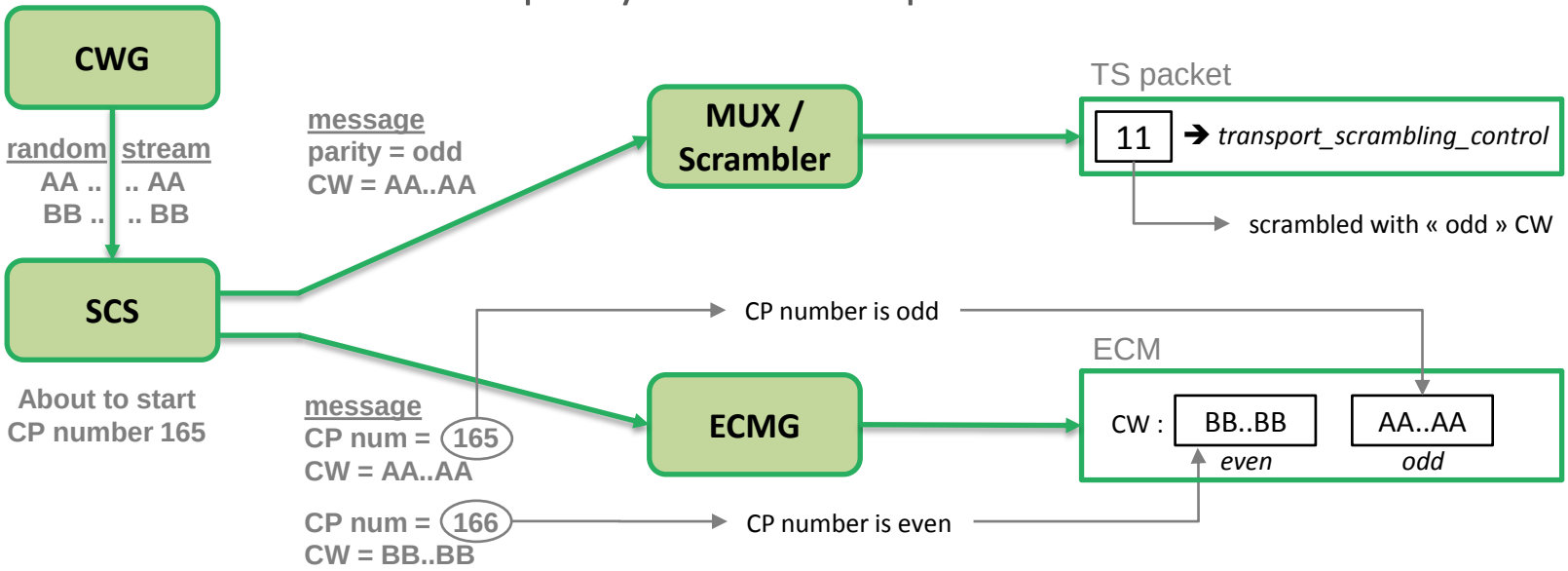
- Using CA_descriptor in the PMT of the service
  - standard part of CA_descriptor : CA system id, ECM PID
    - same as EMM signalization
  - private part of CA_descriptor: CAS-specific
    - used by the CA software in the STB
    - CA_descriptor private part is usually different in CAT (EMM) and PMT (ECM)
    - sample content : operator id, public subset of access criteria
- Two possible positions for CA_descriptors in PMT
  - at program level
    - only if one single scrambling group
  - at stream level
    - mandatory if different ES use different CW
    - take precedence over program level if both are used for same CA_system_id

Scrambling synchronization (1/3)

- During one crypto-period (CP) number N
  - typically 10 seconds
  - scrambling using same CW_N
- ECM_N carries CW_N and CW_{N+1}
  - initial ECM broadcast delayed from start of CP (CAS specific)
  - ECM_N is repeated several times during CP_N (typically 10 ECM/s)
  - if first ECM_{N+1} is missed, the descrambler already knows CW_{N+1} anyway
- The CA software configures the descrambler with both CW_N and CW_{N+1}
  - either N or N+1 is « even », the other one is « odd »
- TS packet header contains 2-bit *transport_scrambling_control*
  - used by the descrambler to select the appropriate CW
    - 00 : clear, do not descramble (MPEG-defined: ISO 13818-1)
    - 10 : use even CW (DVB-defined: ETR 289)
    - 11 : use odd CW (DVB-defined: ETR 289)
- Implemented in TSDuck plugin *scrambler*

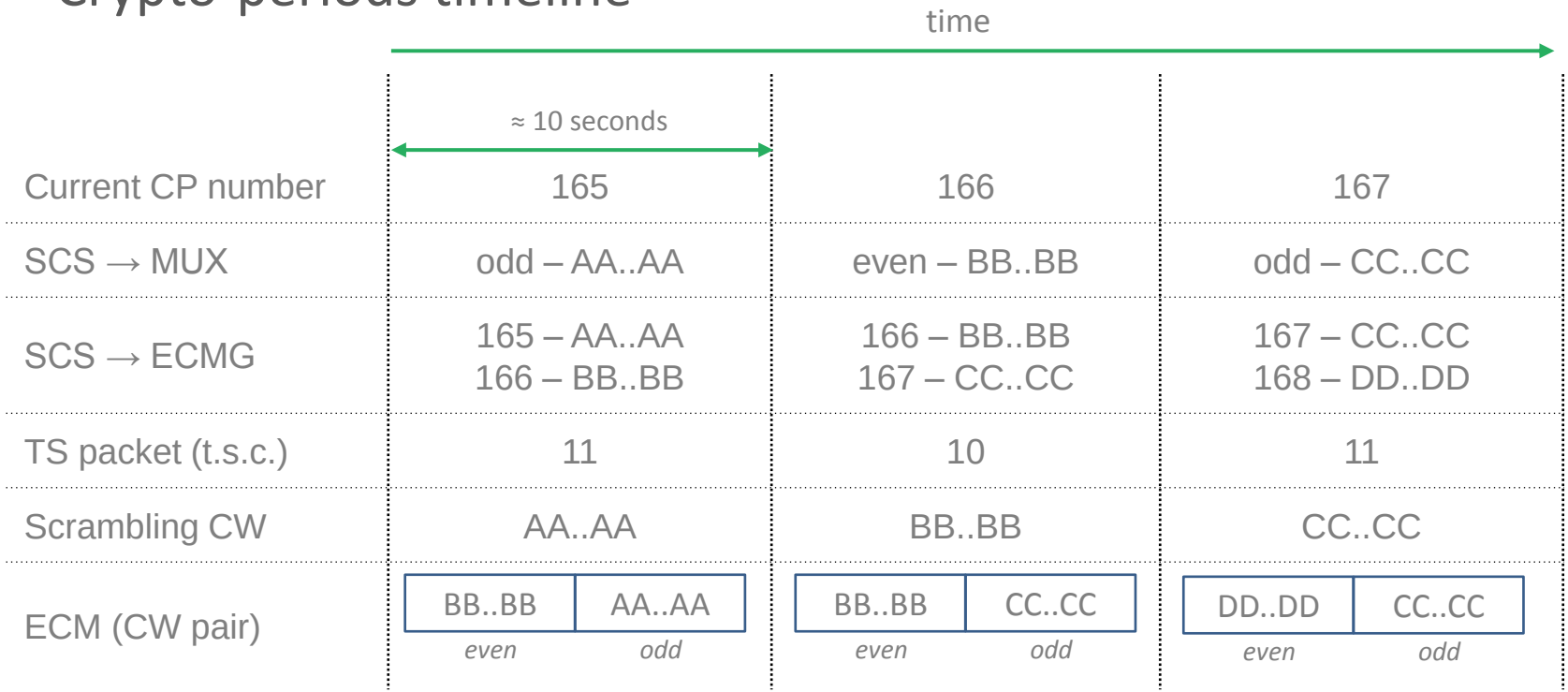
Scrambling synchronization (2/3)

- Based on crypto-period (CP) number
  - CP numbers are sequentially allocated by SCS
  - the full CP number stays on head-end
  - its parity is used in TS packets and ECM's



Scrambling synchronization (3/3)

- Crypto-periods timeline



TS vs. PES scrambling

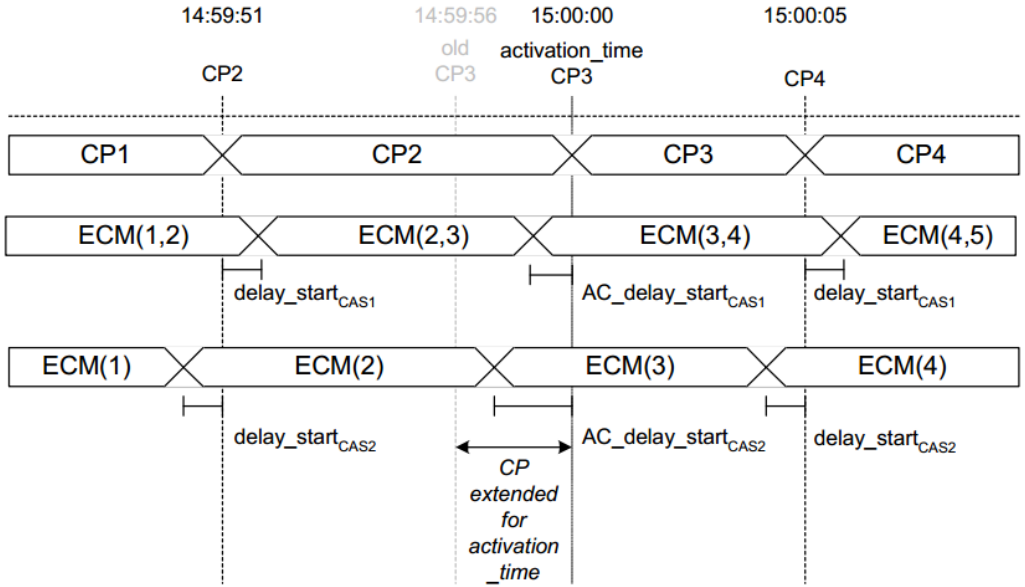
- ISO 13818-1 defines two possible levels of scrambling
  - TS level
    - each TS packet is scrambled individually
    - clear TS header and adaptation field, scrambled TS payload
  - PES level
    - each demuxed PES packet is scrambled individually
    - TS packet header marked as clear
    - PES packet header contains similar 2-bit *PES_scrambling_control*
    - clear PES header, scrambled PES payload
- In practice, only TS-level scrambling is used
  - PES-level scrambling is technically much more difficult
    - scrambling is performed on multiplexed TS
    - ETR 289 specifies sub-scrambling of 184-byte super-blocks
    - PES packet boundaries not aligned on crypto-period boundaries
  - PES-level scrambling is never used in practice

EMM & ECM tables

- CA-private in DVB-defined range
  - ETSI ETR 289 defines the range of private CA table ids
    - 0x80 – 0x81 : ECM
    - 0x82 – 0x8F : « CA private »
  - defined as « short sections »
    - no versioning
    - each section is an independent new table
- Typical usage
  - 0x80 and 0x81 alternating with crypto periods
    - ECM table id change used as trigger by CA software to submit ECM to smartcard or TEE
    - ECM table id and CP number do not necessarily have the same parity
  - 0x82 – 0x8F used for EMM's
    - CAS-specific
    - typically one table id for each EMM type, easier to filter in STB

Access criteria transition

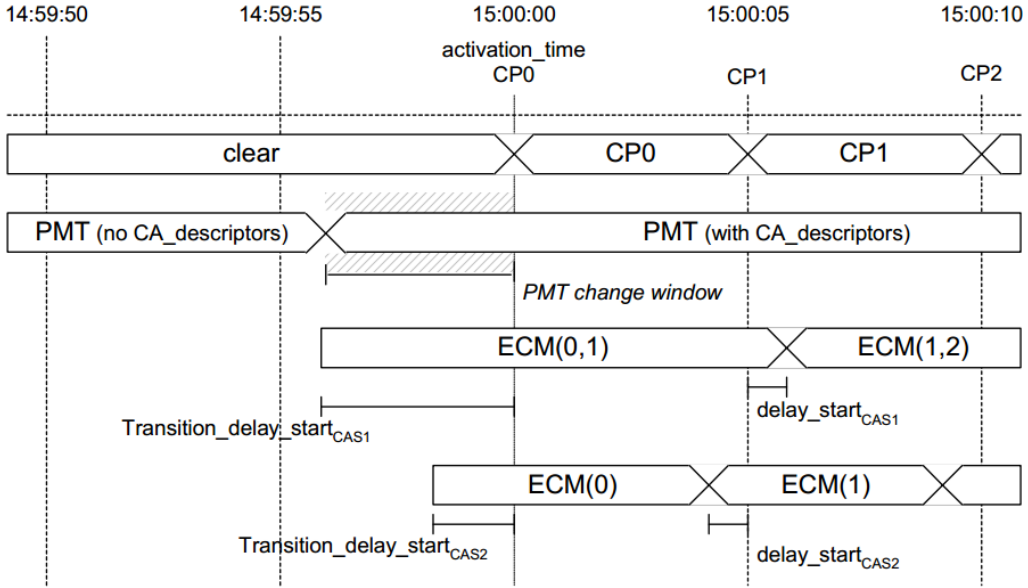
- Use case : restricted event or pay-per-view event transition
- Scenario :
  - the ECMG of each CAS had sent its own timing requirements to SCS
  - SCS synchronizes the generation of the ECM from each CAS



Extracted from DVB standard ETSI TS 103 197

Clear-to-scramble transition

- Use case : Pay-TV channel with public periods in the clear
- Scenario :
  - the ECMG of each CAS had sent its own timing requirements to SCS
  - SCS synchronizes the generation of the ECM from each CAS



Extracted from
DVB standard
ETSI TS 103 197

DVB CSA-2

- DVB Common Scrambling Algorithm
  - DVB proprietary algorithm
  - supposed to be « secret »
  - fully described in Wikipedia
  - open-source implementations online (*libdvbcsa*)
- Algorithm
  - 64-bit key (also known as « Control Words » or CW)
  - first pass : block cipher in reverse-CBC mode
    - use CW as key
    - block size : 64 bits
    - residue ignored
  - second pass : stream cipher
    - use CW as key and first block as seed (last processed block from reverse-CBC)
    - residue included
  - short payloads (1 to 7 bytes) are not encrypted
    - even if *transport_scrambling_control* is non-zero

DVB CSA-2 entropy reduction

- Entering the twilight zone….
- 64-bit key
  - some national regulations from the 90's prohibited 64-bit entropy
  - entropy was artificially reduced to 48 bits
    - cw[3] = (cw[0] + cw[1] + cw[2]) mod 256
    - cw[7] = (cw[4] + cw[5] + cw[6]) mod 256
  - entropy reduction is no longer required but still often applied
- Operational issues
  - hardware scramblers and descramblers use plain 64-bit keys
  - CWG internally generates 64 random bits
  - where is the entropy reduction applied ?
    - common chain : CWG? SCS?
    - scrambling chain : MUX? scrambler?
    - descrambling chain : ECMG? smartcard? CA software in STB? descrambler?
    - who knows if entropy reduction must be applied anyway?

Standards

our essential references

Essential standards

- MPEG
  - ISO 13818-1, MPEG-2 system layer (TS, packetization, PSI)
    - transport stream ➔ broadcast, blu-ray discs
    - program stream ➔ DVD
- DVB / ETSI (Europe and more)
  - EN 300 468, DVB service information specifications (signalization)
  - TS 103 197, DVB simulcrypt head-end (CAS head-end)
- ATSC (USA), ISDB (Japan, Brazil)
  - equivalent features as defined in DVB

Obtaining standards documents

- ISO
  - https://www.iso.org/standards.html
  - must be purchased
- DVB
  - http://www.etsi.org/standards
  - direct search : http://www.etsi.org/standards-search
  - allocated identifiers : http://www.dvbservices.com/identifiers/
- ITU
  - http://www.itu.int/ITU-T/recommendations/
  - H.xxx series : http://www.itu.int/rec/T-REC-H/
- IETF
  - https://tools.ietf.org/
- NIST
  - http://csrc.nist.gov/publications/

Audio and video standards and nicknames

Origin	Type	ISO / IEC	ITU-T	Nicknames
MPEG-1	Video	11172-2	H.261	MPEG-1 video
MPEG-1	Audio	11172-3		MPEG audio layer 1
MPEG-2	Video	13818-2	H.262	MPEG-2 video
MPEG-2	Audio	13818-3		Layer 2: MPEG audio layer 2 Layer 3: MP3
MPEG-2	Audio	13818-7		AAC (Advanced Audio Coding)
Dolby Digital	Audio			AC-3 (Audio Coding 3)
MPEG-4	Video	14496-2	H.263	DivX, Xvid (codecs)
MPEG-4	Audio	14496-3		HE-AAC, EAAC (High Efficiency, Enhanced AAC)
MPEG-4	Video	14496-10	H.264	AVC (Advanced Video Coding)
MPEG-H	Video	23008-2	H.265	HEVC (High Efficiency Video Coding)
Dolby Digital	Audio			AC-4

Thank you