Configure DVB-C Lab Environment with cBR-8, TSDuck, and VLC

Contents

Introduction

This document describes how to configure a Digital Video Broadcasting - Cable (DVB-C) lab scenario with the TSDuck toolkit, VLC, and cBR-8.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- DVB-C
- Symulcrypt
- VoD
- cBR-8

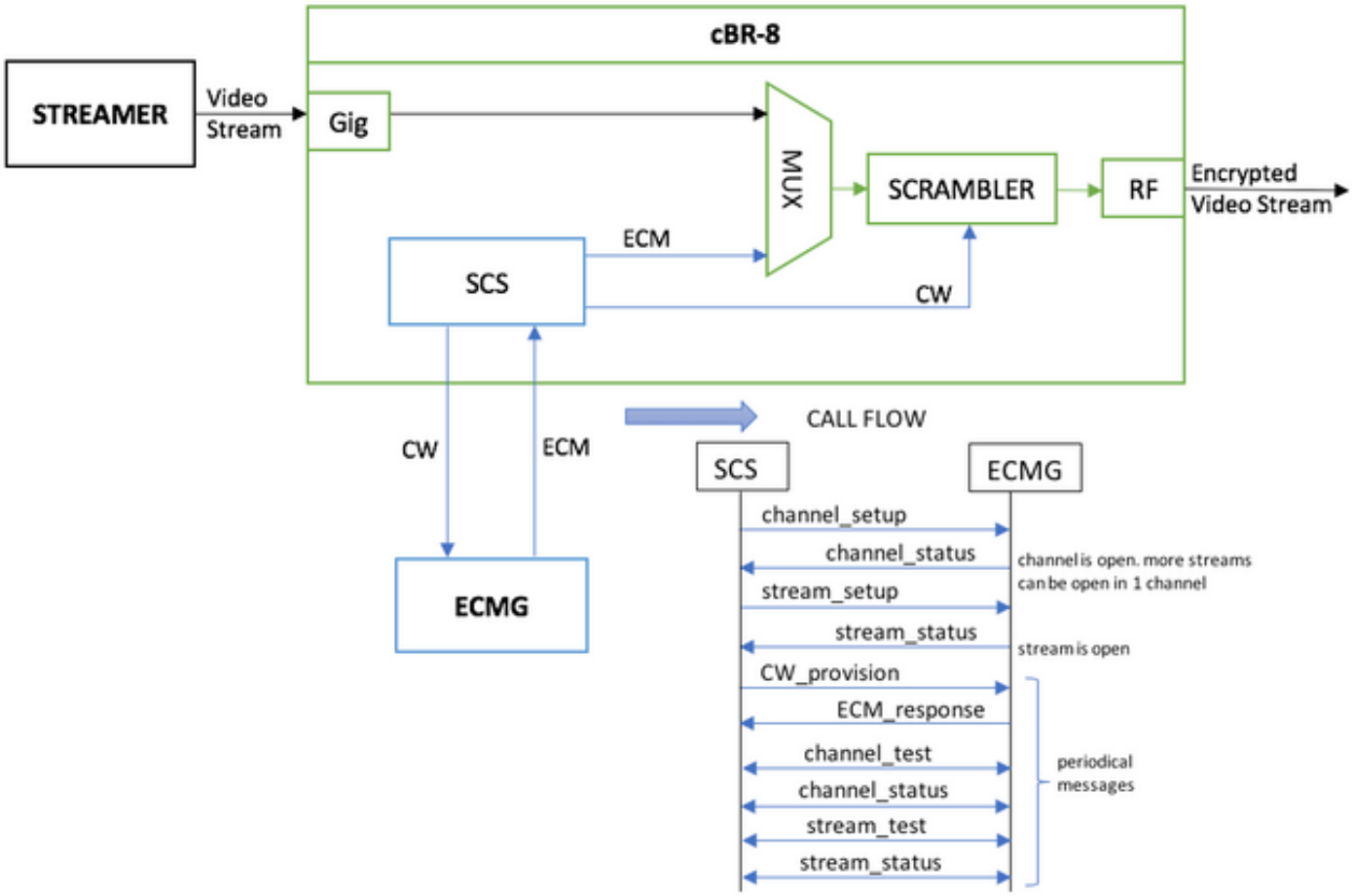
Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

The scenario presented in this document, illustrated in the figure below, involves the cBR-8 as iCMTS, a Linux Virtual Machine (VM) used as video streamer with VLC, and a Linux VM with TSDuck. The DVB-Symulcrypt encryption system is recreated, where the cBR8 acts as Simulcrypt Synchronizer (SCS), and the TSDuck VM plays the Entitlement Control Message Generator (ECMG) role as it would be a Nagra server.



The VM that acts as a streamer, simply sends a locally stored videoclip, which loops in order to simulate a continuous stream. The cBR-8 has one table-based (static) session configured for this simulation, and there is no Set-Top Box (STB) or modem that requests the VoD stream, it is manually initiated on the streamer.

When the stream is received, the cBR-8 tries to communicate witht the configured ECMG server, in order to encrypt the video stream, and exchanges the messages described in the call flow in the figure above. These messages are exchanged in clear with TSDuck, which is good to analyze the content of the messages and debugs. Also TSDuck replies to all the requests sent, without checking the correctness of the parameters as ca-system-id, access-criteria, etc.

If the cBR-8 fails to communicate with the ECMG, the stream is sent out in clear because of the instruction fail-to-clear.

In a real case scenario, there is the need to send to the STBs an Entitlement Management Message (EMM), which authorizes the receiver to decrypt a specific Control Word (CW). The EMMs can be sent through the cBR-8 or on a separate channel to the receivers, and TSDuck has also the function to simulate the EMM Generator (EMMG)

Configure

cBR-8 Video Sessions

Here is an example on how to configure DVB video sessions on cBR-8. The access-criteria is normally provided by the Conditional Access System (CAS), in this simulation case you can generate a random Hex number, as well as for the ca-system-id.

The virtual-edge-input-ip is the IP destination of the stream, which in this case is not a real destination, but it has to be the same IP used to send the video stream from the streamer.

```
cable video
  encryption
    linecard 1/0 ca-system dvb scrambler dvb-csa
    dvb
      ecmg NAGRA_ELK id 1
        mode tier-based
        type nagra
        ca-system-id 2775 3
        auto-channel-id
        ecm-pid-source auto 48 8190
        connection id 1 priority 1 10.48.88.12 3337
        overrule
          min-cp-duration 300000
      tier-based
        ecmg name NAGRA_ELK access-criteria c972bfd7701e6d28069ae85f5d701d63ac1aec4a
        fail-to-clear
        enable
  service-distribution-group SDG-ACDC-LAB-TEST1 id 1
    onid 100
    rf-port integrated-cable 1/0/3
  virtual-carrier-group VCG-ACDC-LAB-TEST1 id 1
    encrypt
    service-type narrowcast
    rf-channel 32-35 tsid 42496-42499 output-port-number 1-4
  bind-vcg
    vcg VCG-ACDC-LAB-TEST1 sdg SDG-ACDC-LAB-TEST1
  logical-edge-device LED-ACDC-LAB-TEST1 id 1
    protocol table-based
      virtual-edge-input-ip 10.10.10.10 input-port-number 1
      vcg VCG-ACDC-LAB-TEST1
      active
  table-based
    vcg VCG-ACDC-LAB-TEST1
      rf-channel 32
        session vod1 input-port 1 start-udp-port 65 num-sessions-per-qam 1 processing-type remap
start-program 1
!
controller Integrated-Cable 1/0/3
 max-carrier 44
 base-channel-power 40
 rf-chan 32 35
  type VIDEO
  frequency 850000000
  rf-output NORMAL
  power-adjust 0.0
  qam-profile 3
```

Streamer

On this device, you can simply install VLC from command line, and start a stream of a locally stored video file.
You can refer to the official Documentation.

Once installed VLC, the command line below shows how to start a stream of the file named cisco-tac-lab.mov, specify the destination IP and port, the tsid and port on the cBR-8, and loop the video in order to simulate a continuous flow (--repeat):

cvlc cisco-tac-lab.mov --sout '#duplicate{dst=udp{mux=ts,dst=10.10.10.10:65,tsid=42496,port=65}}' --repeat &

ECMG

Download TSDuck from the official website: TSDuck, and refer to the user guide documentation in order to install and find features information.

When TSDuck is installed, you can run the ECMG feature on a specific port (-p), with verbose option (-v) and desired level of debugs (-d#).

Example:

sudo tsecmg -p 3337 -v -d7

Verify

On cBR-8

After you configure the video session on the cBR-8, you can verify that the session is created, since this is a table-based configuration the session is always present, and it shows no input stream:

```
acdc-cbr8-2#show cable video session all

Session     Output Frequency  Streaming    Sess Session Source                  UDP    Output
Input       Output  Input     Output    Encrypt  Encrypt        Low PMV   Session
Id          Port    Hz          Type        Type Ucast Dest IP/Mcast IP (S,G)    Port   Program
State       State    Bitrate  Bitrate  Type     Status         Lat NUM   Name
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
1048576     1       850000000  Remap       UDP  10.10.10.10                     65     1        OFF
ON      0         0         DVB      Pending        N   -     vod1.1.0.1.32.65
```

Once you start the video stream, you can see that it is sent in clear, as per the instruction fail-to-clear on the cBR-8 if the ECMG is not up yet:

```
acdc-cbr8-2#show cable video sess logical-edge-device id 1

Session     Output Frequency  Streaming    Sess Session Source                  UDP    Output
Input       Output  Input     Output    Encrypt  Encrypt        Low PMV   Session
Id          Port    Hz          Type        Type Ucast Dest IP/Mcast IP (S,G)    Port   Program
State       State    Bitrate  Bitrate  Type     Status         Lat NUM   Name
-------------------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------------
1048576    1      850000000  Remap       UDP  10.10.10.10                    65    1
ACTIVE-PSI ON      15403951 15164562 DVB      Clear          N   -     vod1.1.0.1.32.65
```

When you start the ECMG as well, you can see that the video session is now encrypted:

```
acdc-cbr8-2#sh cable video sess logical-edge-device id 1

Session    Output Frequency  Streaming   Sess Session Source                 UDP   Output
Input      Output  Input    Output   Encrypt  Encrypt        Low PMV   Session
Id         Port   Hz         Type        Type Ucast Dest IP/Mcast IP (S,G)    Port  Program
State      State   Bitrate  Bitrate  Type     Status         Lat NUM   Name
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
1048576    1      850000000  Remap       UDP  10.10.10.10                    65    1
ACTIVE-PSI ON      15353613 15476997 DVB      Encrypted      N   -     vod1.1.0.1.32.65
```

The encrypted session in detail:

```
acdc-cbr8-2#sh cable video sess logical-edge-device id 1 session-id 1048576
Session Name        : vod1.1.0.1.32.65
Session Id          : 1048576
Creation Time       : Thu Dec  6 14:12:54 2018

Output Port         : 1
TSID                : 42496
ONID                : 100
Number of Sources   : 1
  Destination IP    : 10.10.10.10
  UDP Port          : 65
Config Bitrate      : not specified
Jitter              : 100 ms
Processing Type     : Remap
Stream Rate         : VBR
Program Number      : 1
Idle Timeout        : 2000 msec
Init Timeout        : 2000 msec
Off Timeout         : 60 sec
Encryption Type     : DVB
Encryption Status   : Encrypted

Input Session Stats:
====================
  State: ACTIVE-PSI, Uptime: 0 days 00:31:33
  IP Packets: In 899927, RTP 0, Drop 0
  TP Packets: In 6299489, PCR 6408, PSI 4424, Null 0
              Unreference 2212, Discontinuity 0
  Errors: Sync loss 0, CC error 795, PCR Jump 7,
          Underflow 215, Overflow 4, Block 0
  Bitrate: Measured 16483732 bps, PCR 17930489 bps

Output Session Stats:
=====================
  State: ON, Uptime: 0 days 00:31:33
  TP Packets: In 6297330, PCR 6395, PSI 4416,
              Drop 12801, Forward 6280113, Insert 6029
  Errors: Info Overrun 0, Info Error 0, Block 0, Overdue 54210,
          Invalid Rate 0, Underflow 0, Overflow 0
  Bitrate: Measured 16433824 bps

PAT Info:
=========
```

```
  Version 26, TSID 8724, len 16, section 0/0
  Program 1: PMT 32

Input PMT Info:
===============
  Program 1, Version 28, PCR 100, Info len 0
  PID 100: Type 27, Info len 6, (lang eng)

Output PMT Info:
================
  Program 1, Version 5, PCR 49, Info len 6, (CA SYS-ID 10101, PID 79)
  PID 49: Type 27, Info len 6, (lang eng)

Output PID Map:
===============
  PID 32 -> 48
  PID 100 -> 49
```

And the command to show the ECMG connection status:

```
acdc-cbr8-2#show cable video encryption dvb ecmg id 1 connection
------------------------------------------------------------------------------------------------------
-------------------------------------------------
ECMG ECMG                 ECMG     CA Sys     CA Subsys  PID     Lower  Upper  Streams/   Open
Streams/   Auto Chan  Slot  ECMG         ECMG
ID   Name                 Type     ID         ID         Source  limit  limit  ECMG       ECMG
ID              Connections  Application
------------------------------------------------------------------------------------------------------
-------------------------------------------------
1    NAGRA_ELK            nagra    0x2775     0x3        auto    48     8190   1          1
Enabled    RP    1            Tier-Based

ECMG Connections for ECMG ID = 1

-----------------------------------------------------------------
Conn Conn     IP               Port    Channel Conn       Open
-ID  Priority Address          Number  ID      Status     Streams
-----------------------------------------------------------------
1    1        10.48.88.12      3337    1       Open       1
-----------------------------------------------------------------
```

> **Note**: Once a ECM is received by the cBR-8, it is stored in the cache, and if the connection with the ECMG is lost, the cached ECM is used for encryption until a new one is received.

On The ECMG

Thanks to the debugs enabled, you can see all the messages exchanged between the ECMG and SCS (refer to the call flow illustrated in the initial figure):

```
cisco@simulcrypt:~$ sudo tsecmg -p 3337 -v -d7
debug level set to 7
* Debug: setting socket reuse address to 1
* Debug: binding socket to 0.0.0.0:3337
* Debug: server listen, backlog is 5
* TCP server listening on 0.0.0.0:3337, using ECMG <=> SCS protocol version 2
* Debug: server accepting clients
* Debug: received connection from 88.88.88.89:56102
* Debug: server accepting clients
```

```
* 88.88.88.89:56102: 2018/12/06 14:38:35: session started
* Debug: received message from 88.88.88.89:56102
    channel_setup (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0001
    ECM_channel_id = 0x0001
    Super_CAS_id = 0x27750003

* Debug: sending message to 88.88.88.89:56102
    channel_status (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0003
    ECM_channel_id = 0x0001
    section_TSpkt_flag = 1
    AC_delay_start = 200
    AC_delay_stop = 200
    delay_start = 200
    delay_stop = 200
    transition_delay_start = -500
    transition_delay_stop = 0
    ECM_rep_period = 100
    max_streams = 0
    min_CP_duration = 10
    lead_CW = 1
    CW_per_msg = 2
    max_comp_time = 100

* Debug: received message from 88.88.88.89:56102
    stream_setup (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0101
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    ECM_id = 0x0001
    nominal_CP_duration = 100

* Debug: sending message to 88.88.88.89:56102
    stream_status (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0103
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    ECM_id = 0x0001
    access_criteria_transfer_mode = 0

* Debug: received message from 88.88.88.89:56102
    CW_provision (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0201
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    CP_number = 0
    access_criteria (20 bytes) =
        C9 72 BF D7 70 1E 6D 28 06 9A E8 5F 5D 70 1D 63 AC 1A EC 4A
    CP = 0
    CW (8 bytes) = 4E 0A 45 9D DC 10 4A 36
    CP = 1
    CW (8 bytes) = AB FF 00 AA 9C 4F 11 FC

* Debug: sending message to 88.88.88.89:56102
    ECM_response (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0202
    ECM_channel_id = 0x0001
```

```
    ECM_stream_id = 0x0001
    CP_number = 0
    ECM_datagram (188 bytes) =
        47 5F FF 10 00 80 70 35 80 AA 03 00 30 00 10 00 08 4E 0A 45 9D DC
        10 4A 36 00 11 00 08 AB FF 00 AA 9C 4F 11 FC 00 12 00 14 C9 72 BF
        D7 70 1E 6D 28 06 9A E8 5F 5D 70 1D 63 AC 1A EC 4A 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
        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 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 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 FF FF FF FF FF FF FF
        FF FF FF FF FF FF FF FF FF FF FF FF

* Debug: received message from 88.88.88.89:56102
    channel_test (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0002
    ECM_channel_id = 0x0001

* Debug: sending message to 88.88.88.89:56102
    channel_status (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0003
    ECM_channel_id = 0x0001
    section_TSpkt_flag = 1
    AC_delay_start = 200
    AC_delay_stop = 200
    delay_start = 200
    delay_stop = 200
    transition_delay_start = -500
    transition_delay_stop = 0
    ECM_rep_period = 100
    max_streams = 0
    min_CP_duration = 10
    lead_CW = 1
    CW_per_msg = 2
    max_comp_time = 100

* Debug: received message from 88.88.88.89:56102
    stream_test (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0102
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001

* Debug: sending message to 88.88.88.89:56102
    stream_status (ECMG<=>SCS)
    protocol_version = 0x02
    message_type = 0x0103
    ECM_channel_id = 0x0001
    ECM_stream_id = 0x0001
    ECM_id = 0x0001
    access_criteria_transfer_mode = 0
```

Troubleshoot

On the cBR-8, you can troubleshoot encryption problems with the corresponding supervisor platform traces set to debug or noise level (do not forget to restore the notice level at the end):

set platform software trace sup-veman rp active scs debug

A correct exchange of messages between cBR-8 and ECMG looks like this:

show platform software trace message sup-veman rp active reverse

```
12/07 15:34:43.963 [scs]: [47872]: (debug): ECMG Send channel_setup for channel_id 1

12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Received channel_status for channel_id 1

12/07 15:34:43.965 [scs]: [47872]: (info): ECMG Channel 0 setup to ip 10.48.88.12 port 3337

12/07 15:34:43.965 [scs]: [47872]: (debug): Open stream 1

12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Send stream_setup for channel_id 1, stream_id 1

12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Received stream_status for channel_id 1,
stream_id 1

12/07 15:34:43.965 [scs]: [47872]: (info): ECMG Stream 1 setup to ip 10.48.88.12 port 3337

12/07 15:34:43.965 [scs]: [47872]: (debug): Request ECM for CP 0

12/07 15:34:43.965 [scs]: [47872]: (debug): ECMG Send CW_provision with 20 AC bytes for
channel_id 1, stream_id 1

12/07 15:34:43.966 [scs]: [47872]: (debug): Received ECM_response for channel_id 1, stream_id 1

12/07 15:34:43.966 [scs]: [47872]: (debug): ECMGp: Forward ECM pkts to SCS

12/07 15:34:43.966 [scs]: [47872]: (debug): Received ECM for CP 0

12/07 15:34:56.015 [scs]: [47872]: (debug): ECMG Send channel_test for channel_id 1

12/07 15:34:56.016 [scs]: [47872]: (debug): ECMG Received channel_status for channel_id 1

12/07 15:35:18.039 [scs]: [47872]: (debug): ECMG Send stream_test for channel_id 1, stream_id 1

12/07 15:35:18.042 [scs]: [47872]: (debug): ECMG Received stream_status for channel_id 1,
stream_id 1
```

Related Information

- DVB Simulcrypt technical specification, latest at the time of creation of this article: ETSI TS 103 197 V1.5.1 (2008-10)
- **Technical Support & Documentation - Cisco Systems**