

# Configuración del entorno de laboratorio DVB-C con cBR-8, TSDuck y VLC

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Configurar](#)

[Sesiones de vídeo cBR-8](#)

[Streamer](#)

[ECMG](#)

[Verificación](#)

[En cBR-8](#)

[Sobre el ECMG](#)

[Troubleshoot](#)

[Información Relacionada](#)

## Introducción

Este documento describe cómo configurar un escenario de laboratorio Digital Video Broadcasting - Cable (DVB-C) con el kit de herramientas TSDuck, VLC y cBR-8.

## Prerequisites

### Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

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

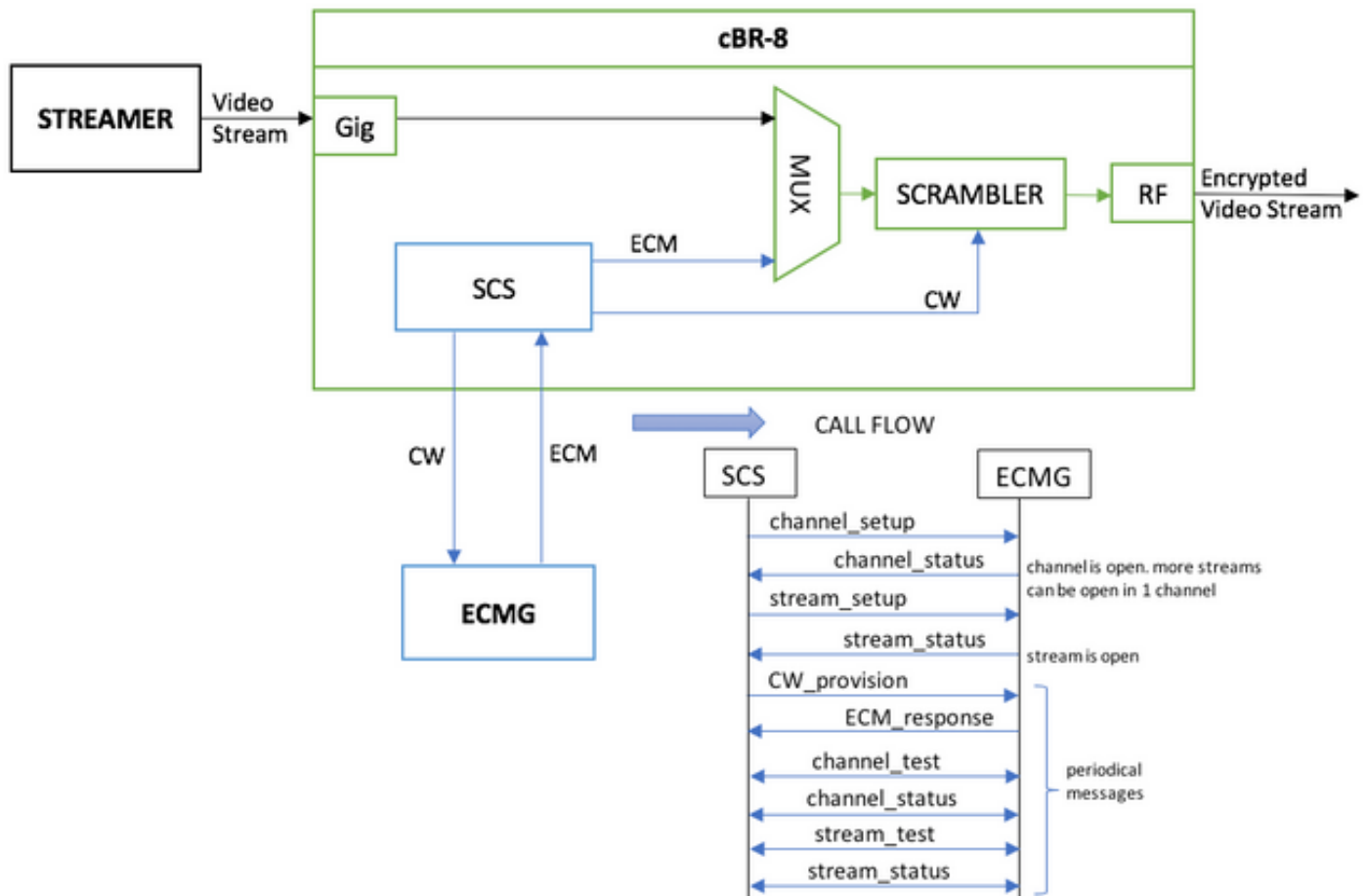
### Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

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. Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

# Antecedentes

El escenario presentado en este documento, ilustrado en la siguiente figura, involucra al cBR-8 como iCMTS, una máquina virtual de Linux (VM) utilizada como streaming de video con VLC y una máquina virtual de Linux con TSDuck. Se vuelve a crear el sistema de cifrado DVB-Symulcrypt, donde el cBR8 actúa como sincronizador de cifrado simultáneo (SCS), y la máquina virtual TSDuck desempeña la función de generador de mensajes de control de derechos (ECMG), ya que sería un servidor Nagra.



La máquina virtual que actúa como una secuencia de transmisión, envía simplemente un videoclip almacenado localmente, que hace bucles para simular una secuencia continua. El cBR-8 tiene una sesión basada en tabla (estática) configurada para esta simulación, y no hay un decodificador (STB) ni un módem que solicite la secuencia VoD; se inicia manualmente en el streaming.

Cuando se recibe la secuencia, el cBR-8 intenta comunicarse con el servidor ECMG configurado para cifrar la secuencia de vídeo e intercambia los mensajes descritos en el flujo de llamada en la figura anterior. Estos mensajes se intercambian claramente con TSDuck, lo que es bueno para analizar el contenido de los mensajes y depuraciones. También TSDuck responde a todas las solicitudes enviadas, sin comprobar la exactitud de los parámetros como ca-system-id, access-Criteria, etc.

Si el cBR-8 no se comunica con el ECMG, el flujo se envía en claro debido a la instrucción fail-to-clear.

En un escenario real, existe la necesidad de enviar a los STB un mensaje de administración de derechos (EMM), que autoriza al receptor a descifrar una palabra de control (CW) específica. Los

EMM se pueden enviar a través del cBR-8 o en un canal separado a los receptores, y TSDuck también tiene la función de simular el Generador EMM (EMMG)

## Configurar

### Sesiones de vídeo cBR-8

Este es un ejemplo de cómo configurar sesiones de vídeo DVB en cBR-8. El sistema de acceso condicional (CAS) proporciona normalmente los criterios de acceso, en este caso de simulación puede generar un número hexadecimal aleatorio, así como el ca-system-id.

El virtual-edge-input-ip es el destino IP de la secuencia, que en este caso no es un destino real, pero tiene que ser la misma IP utilizada para enviar la secuencia de vídeo desde el streaming.

```
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

En este dispositivo, simplemente puede instalar VLC desde la línea de comandos e iniciar una secuencia de un archivo de vídeo almacenado localmente.

Puede consultar la [Documentación](#) oficial.

Una vez instalado VLC, la línea de comandos a continuación muestra cómo iniciar una secuencia del archivo denominado cisco-tac-lab.mov, especificar la IP de destino y el puerto, el tsid y el puerto en el cBR-8, y realizar un loop del vídeo para simular un flujo continuo (`—repetir`):

```
cvlc cisco-tac-lab.mov —sout
'#duplicado{dst=udp{mux=ts,dst=10.10.10.10:65,tsid=42496,port=65}' —repetido y
```

## ECMG

Descargue TSDuck del sitio web oficial: [TSDuck](#), y consulte la documentación de la guía del usuario para instalar y encontrar información sobre las funciones.

Cuando se instala TSDuck, puede ejecutar la función ECMG en un puerto específico (`-p`), con la opción verbosa (`-v`) y el nivel deseado de depuración (`-d#`).

Ejemplo:

```
sudo tsecmg -p 3337 -v -d7
```

## Verificación

### En cBR-8

Después de configurar la sesión de vídeo en el cBR-8, puede verificar que se crea la sesión, ya que se trata de una configuración basada en tabla, la sesión siempre está presente y no muestra flujo de entrada:

```
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			

Una vez iniciado el flujo de vídeo, puede ver que se envía en claro, según la instrucción `fallo-a-clear` en el cBR-8 si el ECMG aún no está activo:

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

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

```
-----
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
-----
```

Quando usted inicia el ECMG también puede ver que la sesión de video está ahora cifrada:

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

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

```
-----
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
-----
```

La sesión cifrada en detalle:

```
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

Y el comando para mostrar el estado de la conexión ECMG:

```
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  
-----  
-----
```

**Nota:** Una vez que el cBR-8 recibe un ECM, se almacena en la memoria caché y, si se pierde la conexión con el ECMG, el ECM almacenado en memoria caché se utiliza para el cifrado hasta que se recibe uno nuevo.

## Sobre el ECMG

Gracias a las depuraciones habilitadas, puede ver todos los mensajes intercambiados entre ECMG y SCS (consulte el flujo de llamadas ilustrado en la figura inicial):

```
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 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

En el cBR-8, puede resolver problemas de cifrado con los seguimientos de la plataforma



supervisora correspondientes configurados en debug o nivel de ruido (no olvide restaurar el nivel de aviso al final):

```
set platform software trace sup-veman rp active scs debug
```

Un intercambio correcto de mensajes entre cBR-8 y ECMG es similar a esto:

```
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
```

## Información Relacionada

- Especificación técnica de DVB Simulcrypt, a más tardar en el momento de la creación de este artículo: [ETSI TS 103 197 V1.5.1 \(2008-10\)](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)