

# Configurazione dell'ambiente lab DVB-C con cBR-8, TSDuck e VLC

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Sessioni video cBR-8](#)

[Streamer](#)

[ECMG](#)

[Verifica](#)

[Su cBR-8](#)

[In ECMG](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

## Introduzione

Questo documento descrive come configurare uno scenario lab Digital Video Broadcasting - Cable (DVB-C) con il toolkit TSDuck, VLC e cBR-8.

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

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

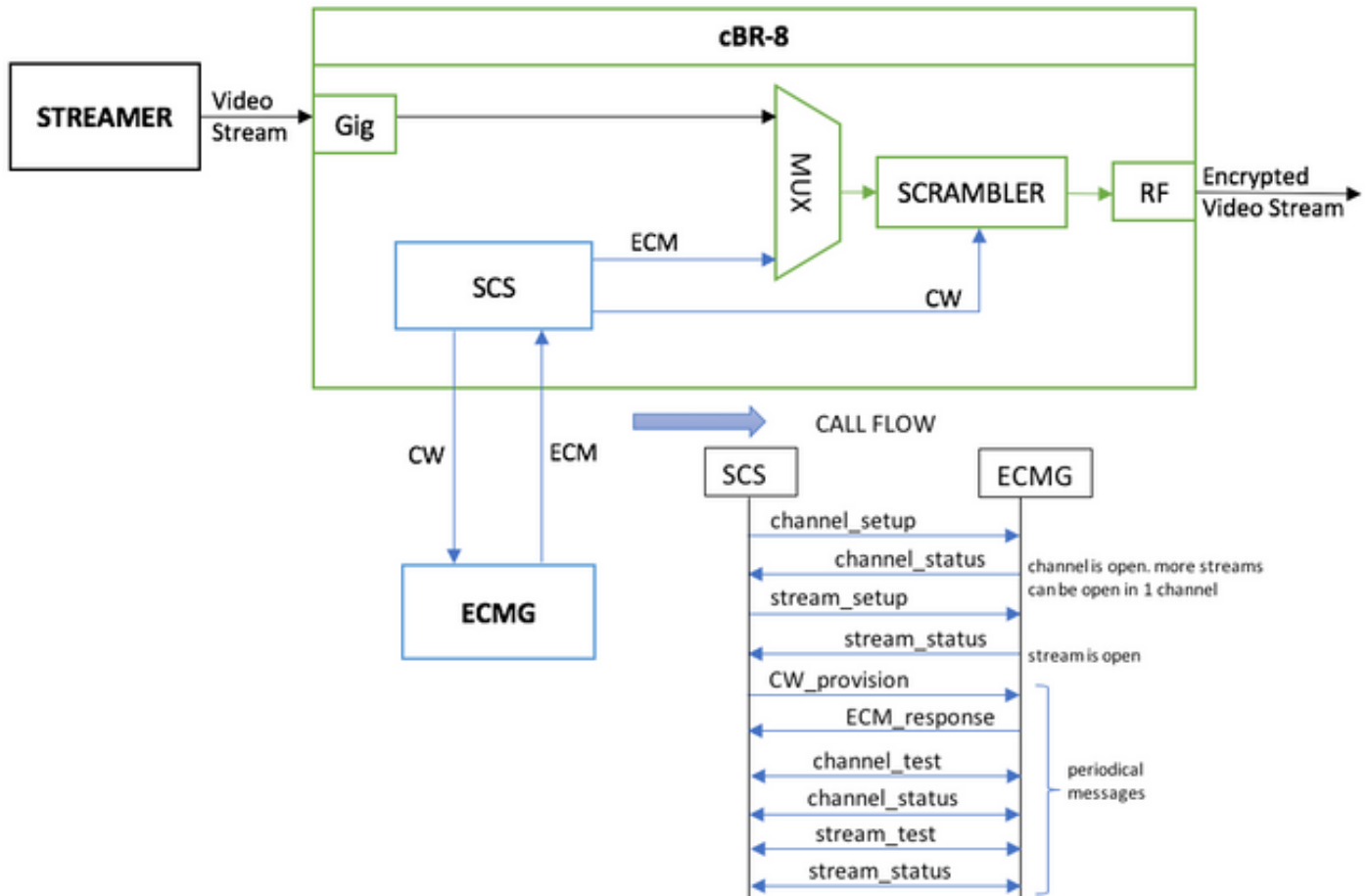
### Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

# Premesse

Lo scenario illustrato in questo documento, illustrato nella figura seguente, riguarda il cBR-8 come iCMTS, una macchina virtuale Linux (VM) utilizzata come video streamer con VLC e una VM Linux con TSDuck. Viene ricreato il sistema di crittografia DVB-Symulcrypt, in cui cBR8 funge da Sincronizzatore Simulcrypt (SCS) e la VM TSDuck svolge il ruolo di generatore di messaggi di controllo delle autorizzazioni (ECMG, Entitlement Control Message Generator) come se fosse un server Nagra.



La VM che funge da streamer invia semplicemente una clip video archiviata localmente, che viene ripetuta per simulare un flusso continuo. Per questa simulazione, cBR-8 dispone di una sessione (statica) basata su tabella e non è disponibile un set-top box (STB) o un modem che richieda il flusso VoD. Il flusso viene avviato manualmente nel flusso.

Quando il flusso viene ricevuto, il cBR-8 tenta di comunicare con il server ECMG configurato, al fine di crittografare il flusso video, e scambia i messaggi descritti nel flusso di chiamata nella figura precedente. Questi messaggi vengono scambiati in modo chiaro con TSDuck, che è utile per analizzare il contenuto dei messaggi e dei debug. Anche TSDuck risponde a tutte le richieste inviate, senza verificare la correttezza dei parametri come ca-system-id, access-criteria, ecc.

Se il cBR-8 non comunica con l'ECMG, il flusso viene inviato in chiaro a causa dell'istruzione fail-to-clear.

In uno scenario reale, è necessario inviare all'STB un messaggio di gestione delle autorizzazioni (EMM), che autorizza il destinatario a decrittografare una parola di controllo (CW) specifica. Gli EMM possono essere inviati attraverso il cBR-8 o su un canale separato ai ricevitori, e TSDuck ha anche la funzione di simulare l'EMM Generator (EMMG)

# Configurazione

## Sessioni video cBR-8

Di seguito è riportato un esempio su come configurare le sessioni video DVB su cBR-8. I criteri di accesso sono normalmente forniti dal sistema di accesso condizionale (CAS, Conditional Access System). In questo caso di simulazione è possibile generare un numero esadecimale casuale, nonché per il ca-system-id.

Virtual-Edge-input-ip è la destinazione IP dello streaming, che in questo caso non è una destinazione reale, ma deve essere la stessa IP utilizzata per inviare il video streaming dallo 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
```

## Streamer

Su questo dispositivo, è sufficiente installare il VLC dalla riga di comando e avviare lo streaming di un file video archiviato localmente.

È possibile fare riferimento alla [Documentazione](#) ufficiale.

Dopo l'installazione del VLC, la riga di comando seguente mostra come avviare lo streaming del file `cisco-tac-lab.mov`, specificare l'IP e la porta di destinazione, il `tsid` e la porta sul `cBR-8` e riprodurre in loop il video per simulare un flusso continuo (`—repeat`):

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

## ECMG

Scarica TSDuck dal sito ufficiale: [TSDuck](#) e consultare la documentazione della guida dell'utente per installare e trovare le informazioni sulle funzionalità.

Quando è installato TSDuck, è possibile eseguire la funzione ECMG su una porta specifica (`-p`), con l'opzione `verbose` (`-v`) e il livello di debug desiderato (`-d#`).

Esempio:

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

## Verifica

### Su cBR-8

Dopo aver configurato la sessione video su `cBR-8`, è possibile verificare che la sessione sia stata creata, poiché si tratta di una configurazione basata su tabella in cui la sessione è sempre presente e non mostra alcun flusso di input:

```
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 volta avviato il flusso video, è possibile vedere che è inviato in chiaro, come indicato nell'istruzione `fail-to-clear` su `cBR-8` se l'ECMG non è ancora attivo:

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

Session	Output	Frequency	Streaming	Sess	Session	Source	UDP	Output
---------	--------	-----------	-----------	------	---------	--------	-----	--------

Input Id	Output Port	Input Hz	Output Type	Encrypt Type	Encrypt Status	Low PMV Lat	Session IP (S,G)	Session Name	Port	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 si avvia anche ECMG, è possibile notare che la sessione video è ora crittografata:

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

Session Id	Output Port	Frequency Hz	Streaming Type	Sess Encrypt Type	Session Encrypt Status	Source Low PMV Lat	Session IP (S,G)	Session Name	Port	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 sessione crittografata in dettaglio:

```
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

E il comando per visualizzare lo stato della connessione 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 volta ricevuto da cBR-8, l'ECM viene memorizzato nella cache e, se la connessione con l'ECMG viene persa, l'ECM memorizzato nella cache viene utilizzato per la crittografia finché non ne viene ricevuto uno nuovo.

## In ECMG

Grazie ai debug attivati, è possibile visualizzare tutti i messaggi scambiati tra ECMG e SCS (fare riferimento al flusso di chiamate illustrato nella figura iniziale):

```

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
```

## Risoluzione dei problemi

Su cBR-8, è possibile risolvere i problemi di crittografia con le tracce della piattaforma supervisor corrispondente impostate su debug o livello di rumore (non dimenticare di ripristinare il livello di



avviso alla fine):

## **set platform software trace sup-veman rp active scs debug**

Uno scambio corretto di messaggi tra cBR-8 ed ECMG ha il seguente aspetto:

### **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
```

## **Informazioni correlate**

- Specifiche tecniche di DVB Simulcrypt, più recenti al momento della creazione di questo articolo: [ETSI TS 103 197 V1.5.1 \(2008-10\)](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)