

# Configuration de l'environnement de travaux pratiques DVB-C avec cBR-8, TSDuck et VLC

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Configuration](#)

[Sessions vidéo cBR-8](#)

[Flux](#)

[ECMG](#)

[Vérification](#)

[Sur cBR-8](#)

[Sur ECMG](#)

[Dépannage](#)

[Informations connexes](#)

## Introduction

Ce document décrit comment configurer un scénario de TP DVB-C (Digital Video Broadcasting - Cable) avec la boîte à outils TSDuck, VLC et cBR-8.

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- DVB-C
- Cryptage
- VoD
- cBR-8

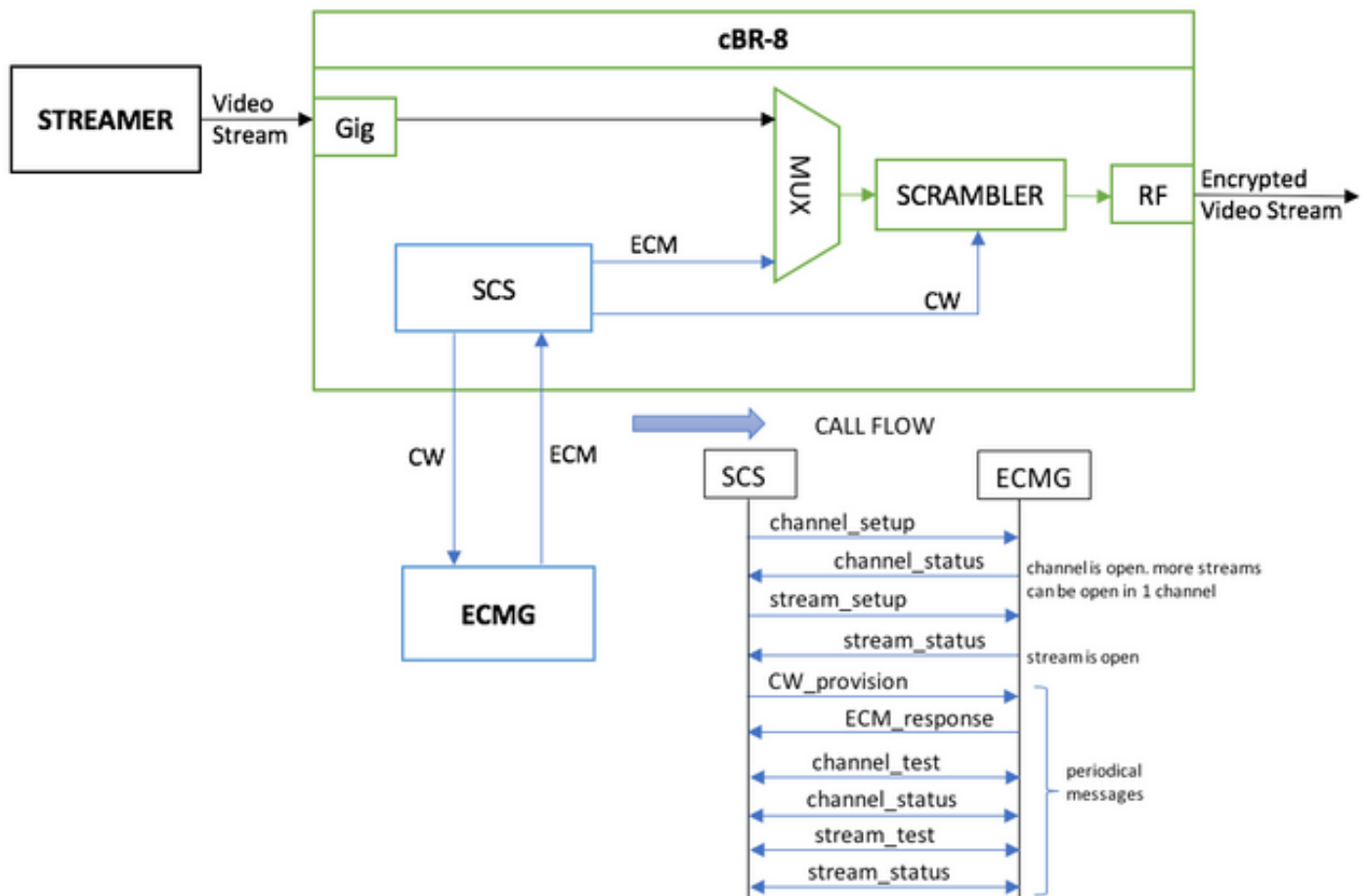
### Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

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 votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

# Informations générales

Le scénario présenté dans ce document, illustré dans la figure ci-dessous, implique le cBR-8 en tant qu'iCMTS, une machine virtuelle Linux (VM) utilisée comme flux vidéo avec VLC, et une machine virtuelle Linux avec TSDuck. Le système de cryptage DVB-Symulcrypt est recréé, où cBR8 agit comme synchroniseur Simulcrypt (SCS), et la machine virtuelle TSDuck joue le rôle de générateur de messages de contrôle d'autorisation (ECMG) comme s'il s'agissait d'un serveur Nagra.



La machine virtuelle qui agit en tant que diffuseur envoie simplement un vidéoclip stocké localement, qui bouge afin de simuler un flux continu. Le cBR-8 a une session (statique) basée sur une table configurée pour cette simulation, et aucun boîtier décodeur (STB) ou modem ne demande le flux VoD, il est lancé manuellement sur le flux.

Lorsque le flux est reçu, le cBR-8 tente de communiquer avec le serveur ECMG configuré, afin de chiffrer le flux vidéo, et échange les messages décrits dans le flux d'appels dans la figure ci-dessus. Ces messages sont échangés en clair avec TSDuck, qui est bon pour analyser le contenu des messages et des débogages. TSDuck répond également à toutes les demandes envoyées, sans vérifier l'exactitude des paramètres comme ca-system-id, access-criteret etc.

Si le cBR-8 ne parvient pas à communiquer avec l'ECMG, le flux est envoyé en clair en raison de l'échec de la suppression de l'instruction.

Dans un scénario réel, il est nécessaire d'envoyer aux STB un message de gestion des droits (EMM), qui autorise le destinataire à déchiffrer un mot de contrôle (CW) spécifique. Les modules EMM peuvent être envoyés par le cBR-8 ou sur un canal distinct aux récepteurs, et TSDuck a également la fonction de simuler le générateur EMM (EMMG)

# Configuration

## Sessions vidéo cBR-8

Voici un exemple de configuration des sessions vidéo DVB sur cBR-8. Les critères d'accès sont normalement fournis par le système d'accès conditionnel (CAS), dans ce cas de simulation, vous pouvez générer un nombre hexadécimal aléatoire, ainsi que pour l'id de système ca.

L'adresse ip de la périphérie virtuelle est la destination IP du flux, qui dans ce cas n'est pas une destination réelle, mais elle doit être la même que celle utilisée pour envoyer le flux vidéo à partir du routeur.

```
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
```

## Flux

Sur ce périphérique, vous pouvez simplement installer VLC à partir de la ligne de commande et démarrer un flux d'un fichier vidéo stocké localement.

Vous pouvez consulter la [documentation](#) officielle.

Une fois le VLC installé, la ligne de commande ci-dessous montre comment démarrer un flux du fichier nommé cisco-tac-lab.mov, spécifier l'adresse IP et le port de destination, le tsid et le port sur le cBR-8, et boucler la vidéo afin de simuler un flux continu (—répéter) :

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

## ECMG

Téléchargez TSDuck à partir du site officiel : [TSDuck](#), et reportez-vous à la documentation du guide de l'utilisateur afin d'installer et de trouver des informations sur les fonctionnalités.

Lorsque TSDuck est installé, vous pouvez exécuter la fonctionnalité ECMG sur un port spécifique (-p), avec l'option verbose (-v) et le niveau de débogage souhaité (-d#).

Exemple :

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

## Vérification

### Sur cBR-8

Après avoir configuré la session vidéo sur le cBR-8, vous pouvez vérifier que la session est créée, car il s'agit d'une configuration de table que la session est toujours présente et qu'elle n'affiche aucun flux d'entrée :

```
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			

Une fois que vous démarrez le flux vidéo, vous pouvez voir qu'il est envoyé en clair, comme le montre l'instruction fail-to-clear sur le cBR-8 si l'ECMG n'est pas encore actif :

```
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	

Lorsque vous démarrez également l'ECMG, vous pouvez voir que la session vidéo est désormais chiffrée :

```
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	

La session chiffrée en détail :

```
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

Et la commande pour afficher l'état de la connexion ECMG :

acdc-cbr8-2#show cable video encryption dvb ecmg id 1 connection

```
-----
```

ECMG ID	ECMG Name	Auto	Chan	Slot	ECMG Type	CA Sys ID	CA Subsys ID	PID Source	Lower limit	Upper limit	Streams/ECMG	Open ECMG
1	NAGRA_ELK	Enabled	RP	1	nagra Tier-Based	0x2775	0x3	auto	48	8190	1	1

ECMG Connections for ECMG ID = 1

```
-----
```

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

```
-----
```

**Note:** Une fois qu'un module ECM est reçu par le cBR-8, il est stocké dans le cache et si la connexion avec le module ECMG est perdue, le module ECM mis en cache est utilisé pour le chiffrement jusqu'à ce qu'un nouveau module soit reçu.

## Sur ECMG

Grâce aux débogages activés, vous pouvez voir tous les messages échangés entre ECMG et SCS (reportez-vous au flux d'appels illustré dans la figure initiale) :

```
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
```

## Dépannage

Sur le cBR-8, vous pouvez résoudre des problèmes de chiffrement avec les traces de plate-forme de superviseur correspondantes définies sur le niveau de débogage ou de bruit (n'oubliez pas de



restaurer le niveau de notification à la fin) :

**set platform software trace sup-ve-man rp active scs debug**

Un échange correct de messages entre cBR-8 et ECMG ressemble à ceci :

**show platform software trace message sup-ve-man 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
```

## Informations connexes

- Spécification technique DVB Simulcrypt, dernière au moment de la création de cet article : [ETSI TS 103 197 V1.5.1 \(2008-10\)](#)
- [Support et documentation techniques - Cisco Systems](#)