

cBR-8, TSDuck 및 VLC를 사용하여 DVB-C 랩 환경 구성

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[cBR-8 비디오 세션](#)

[스트리머](#)

[ECMG](#)

[다음을 확인합니다.](#)

[cBR-8](#)

[ECMG](#)

[문제 해결](#)

[관련 정보](#)

소개

이 문서에서는 TSDuck 툴킷, VLC 및 cBR-8을 사용하여 DVB-C(Digital Video Broadcasting - Cable) 랩 시나리오를 구성하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- DVB-C
- 시스템 암호화
- VoD
- cBR-8

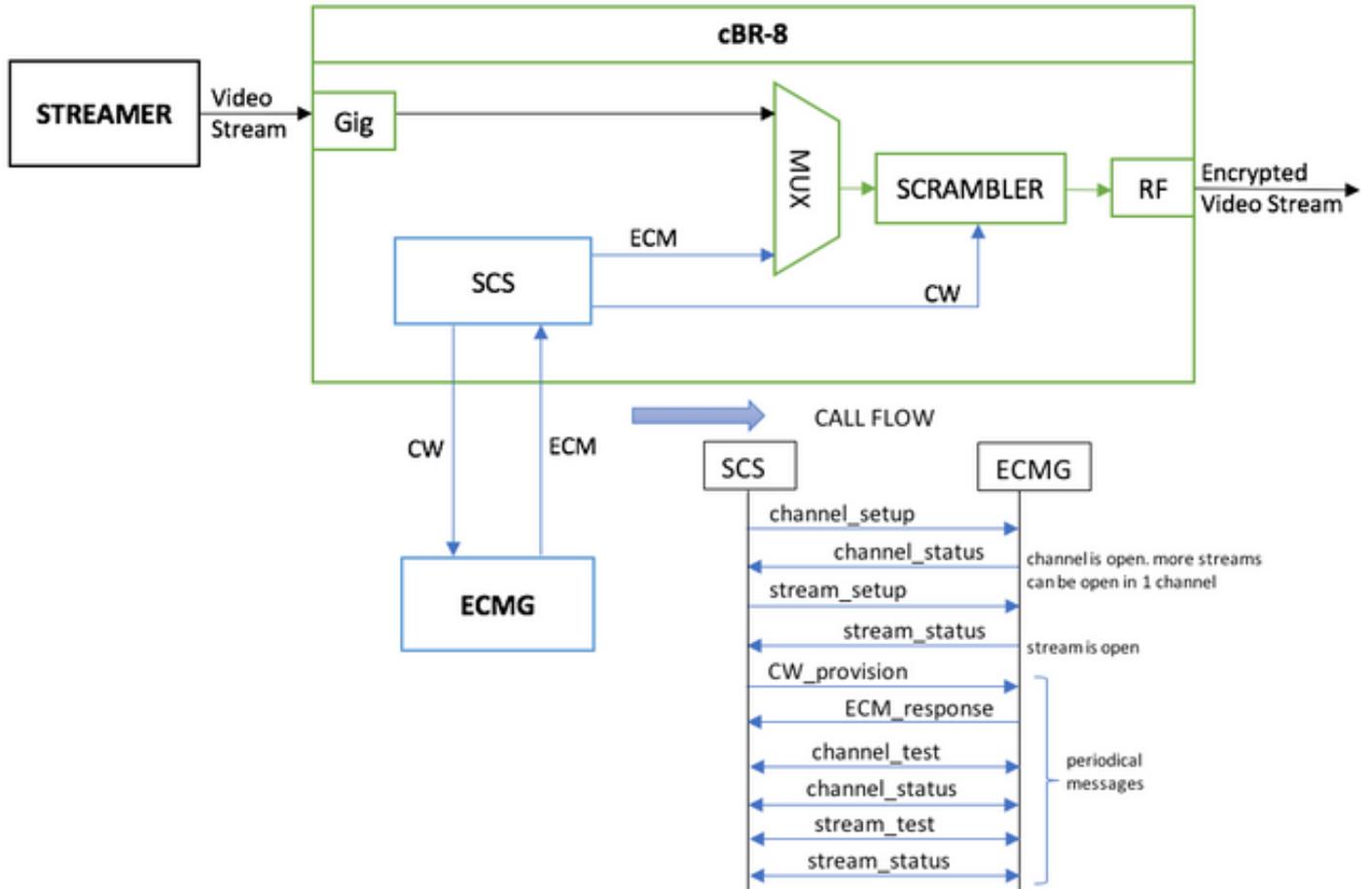
사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

아래 그림에 나와 있는 시나리오는 iCMTS로 cBR-8, VLC에서 비디오 스트리밍으로 사용되는 Linux VM(Virtual Machine) 및 TSDuck가 설치된 Linux VM과 관련이 있습니다. DVB-Symulcrypt 암호화 시스템이 다시 생성됩니다. 여기서 cBR8은 SCS(Simcrypt Synchronizer) 역할을 수행하고 TSDuck VM은 Nserver처럼 ECMG(Entitlement Control Message Generator) 역할을 수행합니다.



스트림 역할을 하는 VM은 단순히 로컬에 저장된 비디오 립을 전송하여 연속 스트림을 시뮬레이션 합니다. cBR-8에는 이 시뮬레이션에 대해 하나의 테이블 기반(정적) 세션이 구성되어 있으며, VoD 스트림을 요청하는 STB(Set-Top Box) 또는 모뎀이 없으므로 스트리밍에서 수동으로 시작됩니다.

스트림이 수신되면 cBR-8은 비디오 스트림을 암호화하고 위 그림의 통화 흐름에 설명된 메시지를 교환하기 위해 구성된 ECMG 서버와 통신을 시도합니다. 이러한 메시지는 TSDuck와 명확하게 교환되며, 이는 메시지 및 디버그의 내용을 분석하는 데 유용합니다. 또한 TSDuck는 매개 변수를 ca-system-id, access-criteria 등으로 정확하게 확인하지 않고 전송된 모든 요청에 응답합니다.

cBR-8이 ECMG와 통신하지 못하면 명령이 Fail-to-Clear로 인해 스트림이 지워집니다.

실제 시나리오에서는 수신자가 특정 Control Word(CW)의 암호를 해독할 수 있도록 권한을 부여하는 EMM(Entitlement Management Message)을 STB에 보내야 합니다. EMM은 cBR-8을 통해 또는 수신자에게 별도의 채널을 통해 전송할 수 있으며 TSDuck는 EMMG(EMM Generator)를 시뮬레이션하는 기능도 갖추고 있습니다

구성

cBR-8 비디오 세션

다음은 cBR-8에서 DVB 비디오 세션을 구성하는 방법에 대한 예입니다. 액세스 기준은 일반적으로 CAS(Conditional Access System)에서 제공하며 이 시뮬레이션 사례에서는 ca-system-id와 임의의 16진수 숫자를 생성할 수 있습니다.

virtual-edge-input-ip는 스트림의 IP 대상이며, 이 경우 실제 대상은 아니지만, 스트림에서 비디오 스트림을 전송하는 데 사용되는 IP와 동일해야 합니다.

```
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
```

스트리머

이 장치에서는 명령줄에서 VLC를 설치하고 로컬에 저장된 비디오 파일의 스트림을 시작할 수 있습니다.

공식 문서를 참조할 수 [있습니다](#).

VLC를 설치한 후 아래 명령줄은 cisco-tac-lab.mov 파일의 스트림을 시작하고 대상 IP 및 포트, cBR-8의 tsid 및 포트를 지정하고 비디오를 루프하여 연속 플로우를 시뮬레이션하는 방법을 보여줍니다(—repeat).

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

ECMG

공식 웹 사이트에서 TSDuck 다운로드:[TSDuck](#)를 설치하고 기능 정보를 찾으려면 사용 설명서 설명서를 참조하십시오.

TSDuck가 설치된 경우 자세한 정보 옵션(-v) 및 원하는 디버그 수준(-d#)을 사용하여 특정 포트(-p)에서 ECMG 기능을 실행할 수 있습니다.

예:

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

다음을 확인합니다.

cBR-8

cBR-8에서 비디오 세션을 구성한 후 세션이 생성되었는지 확인할 수 있습니다. 이는 테이블 기반 구성이며 세션이 항상 존재하며 입력 스트림이 표시되지 않기 때문입니다.

```
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		

비디오 스트림을 시작하면 cBR-8에서 지침에 따라 ECMG가 아직 가동되지 않은 경우 명령이 Fail-to-Clear로 전송되는 것을 확인할 수 있습니다.

```
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	

ECMG를 시작할 때 비디오 세션이 암호화되었음을 확인할 수 있습니다.

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

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

암호화된 세션의 세부 정보:

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

ECMG 연결 상태를 표시하는 명령입니다.

accdc-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
-----
-----
```

참고:cBR-8에서 ECM을 수신하면 캐시에 저장되며, ECMG와의 연결이 끊기면 새 ECM이 수신될 때까지 캐시에 저장된 ECM을 암호화에 사용합니다.

ECMG

활성화된 디버깅 덕분에 ECMG와 SCS 간에 교환되는 모든 메시지를 볼 수 있습니다(초기 그림에 표시된 통화 흐름 참조).

```
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
```

```
* 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
```

문제 해결

cBR-8에서는 디버그 또는 노이즈 레벨로 설정된 해당 슈퍼바이저 플랫폼 추적에 대한 암호화 문제를 해결할 수 있습니다(마지막에 알림 레벨을 복원하는 것을 잊지 마십시오).

플랫폼 소프트웨어 추적 sup veman rp 활성 scs 디버그 설정

cBR-8과 ECMG 간의 올바른 메시지 교환은 다음과 같습니다.

```
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

관련 정보

- DVB Simplicrypt 기술 사양, 이 문서 작성 시 최신 정보:[ETSI TS 103 197 V1.5.1\(2008-10\)](#)
- [기술 지원 및 문서 - Cisco Systems](#)