PowerVu with CSA instead of DES

On some transponders PowerVu is used in combination with CSA instead of DES. This document describes how CSA key calculation differs from DES key calculation.

23.12.2014 (Version 1.0) up-to-date version http://colibri-dvb.info => PowerVu
Colibri <colibri.dvb@gmail.com>

Since years I know that the newer PowerVu IRDs can also use CSA instead of DES because I have partially disassembled the IRD firmware, but since a few days I saw that CSA not only possible, but is really used.

There is an unencrypted flag in the ECM that indicates if DES or CSA must be used to decrypt the video/audio streams.

Each ECM contain nanos. Starting with a two byte long length information (must be masked with FFFh). The next byte is the tag (the type of the nano). Then the data bytes of the nano will follow.

Data byte 0 and 1 of every nano must have the value 0E00h.

Data byte 2 and 3 of every nano must have the value 0000h.

In nano 20h the two msbits of data byte 7 indicates the if DES or CSA must be used to decrypt the video/audio streams:

```
00xxxxxxb = DES
10xxxxxxb = CSA
```

Here is an example of an ECM that shows that **DES** must be used to decode the video/audio stream:

```
47 41 F4 1B <= TS header
00 81
30 3D <= 303Dh & FFFh = 3Dh section length

30 37 <= 3037h & FFFh = 37h length of the first nano
20 <= tag
0E 00 <= must be 0E00h
00 00 <= must be 0000h
00
8E <= continues counter
A0
00 <= 00000000b the two msbits are relevant (00b means DES / 10b means CSA)
39 09 9C E8 53 75 06 02 9F 4C F0 EF 72 8C 00 00 90 00 00 B9 BD 34 AD A5 02 D6 B5 EE 8A 4D
2B D3 4A EA 67 8E 23 7B 28 A0 8F 95 37 EC 86 B1 <= remaining data of the first nano

5A 9D 67 9D <= CRC32
```

Here is an example of an ECM that shows that **CSA** must be used to decode the video/audio stream:

```
47 57 70 1E <= TS header
00 80
30 61 <= 3061h & FFFh = 61h section length

30 37 <= 3037h & FFFh = 37h length of the first nano
20 <= tag
0E 00
00 00
00
A5 <= continues counter
A0
80 <= 10000000b the two msbits are relevant (00b means DES / 10b means CSA)
94 8A 13 91 B8 1C 89 73 2F FC FD 2D 16 18 00 00 10 00 00 26 18 A5 0F DC EE 2F E4 F5 BA 62
71 88 55 F0 C2 06 D3 53 31 FE 2E 1A 8B 6F 0C 3C <= remaining data of the first nano

30 10 <= 3010h & FFFh = 10h length of the second nano
27 <= tag
0E 00 <= must be 0E00h
00 00 <= must be 0E00h
80 <= key type (e.g. 80h is VID key)
00
4D 4A 7D 14 AA 58 69 6B <= convolved CW
82 <= check-sum

30 10 <= 3010h & FFFh = 10h length of the third nano
27 <= tag
0E 00 <= must be 0E00h
00 00 <= must be 0E00h
```

```
20 <= key type (e.g. 20h is A1 key)
00
D0 C5 C5 5A 6C 55 81 42 <= convolved CW
A8 <= check-sum

A1 4D FE 9C <= CRC32
```

You can see that for CSA decryption nano 27h is present (at least one, max. 8).

In the IRD firmware the 8 bytes are named "Convolved CW".

In my other PowerVu papers you can find the info how the DES key is calculated.

During the calculation the 7 byte long seed7 value is used as DES key without parity.

For CSA the seed7 value gets XORed with the convolved CW to get the CW.

Cw[0] = seed7[0] XOR ConvolvedCw[0]
Cw[1] = seed7[1] XOR ConvolvedCw[1]
Cw[2] = seed7[2] XOR ConvolvedCw[2]
Cw[3] = seed7[3] XOR ConvolvedCw[3]
Cw[4] = seed7[3] XOR ConvolvedCw[4] //seed7[3] is used a second time!
Cw[5] = seed7[4] XOR ConvolvedCw[5]
Cw[6] = seed7[5] XOR ConvolvedCw[6]
Cw[7] = seed7[6] XOR ConvolvedCw[7]

After the following calculation you have the final CW that can be used for decryption.
Cw[3] = Cw[0] + Cw[1] + Cw[2];
Cw[7] = Cw[4] + Cw[5] + Cw[6];

For every nano that contains a convolved CW a key type byte is present that indicates for what the final CW can be used. Here are all possible values:

```
80h = VID (VIDeo)
40h = HSD (High Speed Data)
20h = A1  (Audio 1)
10h = A2  (Audio 2)
08h = A3  (Audio 3)
04h = A4  (Audio 4)
02h = UTL (UTiLity)
01h = VBI (Vertical Blanking Interval)


80 <= key type (e.g. 80h is VID key)
00
4D 4A 7D 14 AA 58 69 6B <= convolved CW
82 <= check-sum
```

When adding all bytes from key type to incl. check-sum byte itself you should get a 00h byte.
`80 + 00 + 4D + 4A + 7D + 14 + AA + 58 + 69 + 6B + 82 = (4)00`

From the Ku band transponders that I can receive I have found that CSA is used for the following transport streams:
30W 12092V
4W 10842H

If you want try a **HSD** instead of a normal VID/A1 DES key you can use 9E 11823H sid 9.
I was able to get the Video after removing the IP UDP protocol.