

gbox FAQ

Version 0.1.4 für gbox ab V2.01

Vorwort:

Für dieses FAQ hab ich die originale Readme der gbox als Grundlage verwendet und leicht um die neuen Änderungen der gbox erweitert. Danach gehe ich auf die verschiedenen Einstellmöglichkeiten der gbox in den config Dateien sowie zu den Keydateien ein und versuche diese zu dokumentieren.

Vorwort 2:

Aus gegebenem Anlass ein Zitat aus der readme.1st was jedem zu denken geben sollte

In the past Weeks things are moving in bad directions.

I had never anything against it, that anyone uses the gbox. The only thing I am asking is not to spread it on upload centers, in images, and not to discuss it in public Board areas. It is only about security, nothing else. And security has a higher priority then the freedom of speech.

Money was NEVER a subject. I and my Friends have NEVER taken a Penny for the gbox, a nds keyfile, or for anything else ! Everyone claiming different is simply NOT telling the true.

I am very sad that people do not respect my/our wishes. The situation is bad, and it will become even worse. The nds security officers and lawyers have begone knocking on our doors, and The Hydra Team was already forces to quit. And more others will follow.

This is not a game any more, if you are not cooperative they are threatening with a Multimillion US\$ lawsuit, and even there is a chance that you could win the case, the lawyer and court costs will ruin everyone. The EU (and other) laws are very restrictive against hacking, only the mentioning of the word Emu/Softcam is seen as illegal, or only the pmt.tmp implementation can be seen as a crime animation. So, a lot of people that have made a big (NONPROFIT) contribution to the DVB Scene will become big trouble, people that have spent a lot of theyre free time to develop Software for all of you, time that they could have spend with theyre wives and kids.

And all of this is happening because some guys did not respect these developers, people like adas that are uploading the gbox on upload centers, people like Sandali (owner of <http://opencam.altervista.org>) that has cracked the gbox nds keyfile, people like Goran, that are animating discussions about the gbox, and Boards like Satdudez that are tollerating and not banning such discussions.

You are bringing us all into danger, you are making us to rethink if we should spend our free time in a different way. I simply do not understand ! What motive could Sandali have, to publish in his gbox crack, my eeprom key, he is directly threatening my Friend that has given me the key from his box, he can now be tracked and it is to expect that legal action against him will follow.

Thank you Sandali, you know why this nds eeprom keyfile was there, to protect that eeprom key that you have published. In the nds key file the card serial was crypted together with the eeprom key, and if anyone had published the eeprom key, he would publish also his cards serial. That was the ONLY reason !!! Everyone that has requested such a keyfile, had got one from me, EVERYONE, free of charge, money was NEVER taken. For this Subject I don't blame only Sandali, I also blame all other that were informed of what he intended to do, and have take no action to convince him to stop, in order "to avoid trouble" !

And now explain me Sandali why you have published the private eeprom key ? You are not a Hero Sandali, you are the opposite. You have harmed a lot of people with your action, people that have done more for the community that you have, people that have never done anything against you, people that would have helped you for anything, if you have asked them.

Alea iacta ennt

I have been asked from my Friend that have helped my with the nds camcrypt, to remove it from the gbox. I am loyal to my Friends, and I have done so. The gbox is NOT supporting nds cards any more. I hope this will help my Friend to ease the preasure on him.

If things continue to move in the wrong direction, I will consider to stop developing this Software for the wider public. I will just make it just for me and several real Friends I have.

It's up to you now.

Die UDP Befehle:

UDP port 8003: gbox_out (CW_write, Need_pids)
UDP port 8004: gbox_in (PMT, CAT)
UDP port 8005: gbox_in (pid1 von der Need_pids liste)
UDP port 8006: gbox_in (pid2 von der Need_pids liste)
UDP port 8007: gbox_in (pid3 von der Need_pids liste)
usw.

CW_Write : 89 CW1 CW2

Beispiel : 89 01 02 03 04 05 06 07 08 10 20 30 40 50 60 70 80

Need_pids : 8A pid_anzahl(1 byte) pid1(2 bytes) pid2(2 bytes) usw ...

Beispiel : 8A 02 10 00 15 00

CAT : 86 01

PMT : 87 02

pidx : 88 pid(2 bytes) len(1 byte) data

Beispiel : 88 15 01 48 81 70 ...

Bei jedem Need_pids muss der Server alle pids \geq 8005 schließen, und nach der erhaltenen Liste alle neu öffnen.

Irdeto AU

- Single Key update
- multiple Key update - 2 keys
- multiple Key update - 2 single key update in ein Packet
- PMK update

ein multiple Key update - 4 keys konnte ich noch nicht erwischen, sollte aber auch klappen.

Es können beliebig viele Prov/PMK's eingegeben werden. Mit debug Level 2 werden alle keys geschrieben, unabhängig ob diese neu oder alt sind. So können alle Prov/PMK auf Gültigkeit überprüft werden.

Falls eine Provider ID als 00 00 00 angegeben wird. Wird diese mit jedem EMM überprüft, bei gültigen Signatur wird ein update ausgeführt.

Seca AU

- multiple Key update - 3 keys
- AU über primary MK oder primary & secondary MK

Falls ein key bekannt ist, nicht aber die PPUA und/oder key Nummer, wird die PPUA als 00 00 00 00 angegeben. In diesem fall wird dieser Key für alle EMM's überprüft, bei gültigem hash&signatur wird dann das update ausgeführt, und somit wird auch die PPUA/KeyNr ermittelt.

Wie viele 00 00 00 00 keys gleichzeitig bearbeitet werden können, hängt von der Geschwindigkeit der EMM Daten ab, einer geht aber immer.

Es können beliebig viele PPUA/MK's pro Provider benutzt werden. Mit debug Level 2 werden alle keys geschrieben, unabhängig ob diese neu oder alt sind. So können alle PPUA/MK auf Gültigkeit überprüft werden.

Für Seca2 werden noch die Dateien s2issuer.b und s2provid.b für AU benötigt.

Via AU

- single Key update
- multiple key update 2 keys
- AU über SA (SRG, SloTV, ...)
- AU über UA (SVT, HRT, ...)
- AU über issuer

Es können beliebig viele Prov/PMK's eingegeben werden. Mit debug Level 2 werden alle keys geschrieben, unabhängig ob diese neu oder alt sind. So können alle Prov/PMK auf Gültigkeit überprüft werden.

Falls eine PPUA als 00 00 00 00 angegeben wird, wird dieser MK mit allen EMM's überprüft, im falle einer gültigen Signatur wird das update ausgeführt.

Nagra AU

implementiert für ROM3, ROM7, ROM10 und ROM11 Kartentypen.

0001 - Dish Networks
0801 - Bell Express VU
2C01 - SABC
4001 - Via Digital
4601 - Euskatel
4801 - TV Cabo
4A01 - MMBN (Asia)
4E01 - Dream TV (Philippines)
5401 - NTL Cable
5A01 - TeleWest Cable
5C01 - C&W/NTL Cable
7001 - Polsat
7401 - Star Digital
9401 - Hong Kong Cable

Hier gibt es nicht viel zu sagen, ist komplett plug and play. Einfach in Autoupdate_mode=Auto laufen lassen. Alle rom's, eeprom's, ram's, keys, usw. sind in der gbox integriert. Seit der Version 1.28 sind die Roms extern, bezeichnet sind diese mit z.b *rom02.b*, *rom02.eep.b* und *rom02.ram.b*

Benötigt werden sie alledings nicht unbedingt, da schon seit Monaten das au auch ohne die Roms läuft.

Desweiteren gibt es eine neue Key-Datei namens Nagra.txt in der die Nagra EMM keys abgelegt sind. Worüber das AU läuft.

Conax AU

Es können beliebig viele Prov/PMK's eingegeben werden. Mit debug Level 2 werden alle keys geschrieben, unabhängig ob diese neu oder alt sind. So können alle Prov/PMK auf Gültigkeit überprüft werden.

Das AU ist implementiert für komplette updates (exp+mod) sowie für partielle updates (nur exp oder nur mod)

Gbox Multisystem Softcam

In der gbox ist ein Multisystem Softcam integriert. Um es zu nutzen muss ein Chipkarten Lesegerät am com1 oder com2 (Modem slot bei dbox2) angeschlossen werden.

Es kann eine Smartmouse/Phoenix am com1 oder com2 angeschlossen werden. Oder ein sc8in1 für bis zu 8 Karten an einem com port. Damit ist es möglich gleichzeitig bis zu 8 Chipkarten über com1 oder com2 anzusprechen. Dabei ist es egal um welches Kartensystem es sich handelt, es können alle Karten und Systeme kombiniert werden. Dies entspricht 8 Magiccam's

Der com2 port an der dbox2 ist TTL Level, da kann ein sc8in1 direkt angeschlossen werden, eine Smartmouse jedoch nicht. Hierfür wird z.b. ein multicam benötigt. Wie ein Multicam am com2 der dbox2 anzuschließen ist, könnt ihr in diversen Faq´s nachlesen.

Zur Zeit werden folgende Cam Systeme unterstützt:

Irdeto allcam	A0x/K0x/S0x/C/D/F/1.x/382/383/384/Z/nonZ/DS9/Xin1
Seca/Mediagard	Original Seca1+Seca2, GW, Fun
Viaccess	Original, GW, Fun
Nagravision	Original, Fun
Cryptoworks	Original
Conax	Original
NDS	Original

Alle Systeme mit vollem Autoupdate Support.

Um eine NDS Karte zu benutzen wird ein Reciever Key benötigt der in der Firmware gecrypted ist. Dieser kann so plain nicht weitergegeben, noch in der gbox abgelegt werden.

Jeder der ihn haben will muss dem Entwickler der gbox seine Karten-Serial (in Hex) zuschicken, danach bekommt er von ihm einen gecrypteten Key in einem NDS file das dann nach /var/keys kopiert werden muss. Dieser wird dann nur mit dieser Karte laufen, auch in künftigen gbox Versionen.

Um die Karten-Serial zu ermitteln, startet die gbox mit der NDS Karte und die wird die hex Serial im log anzeigen. Mit diesem wird eine Karte nicht eindeutig identifiziert, weil die letzte Ziffer der Serial nicht im Hex drinn ist.

Getestet wurden bisher 091f (viasat), 0919 (skyitalia) und 0961 (bskyb)

0919 und 0961 haben hohe Baudraten und laufen nur am externen com port der STBx45xx Boxen. Die 091f auch im Internen Slot der STBx45xx Boxen.

In der Dbox2 laufen sie auch im internen Multicam, getestet mit SC8in1 und Friedels Multicam rev.1.2

Ab der Version 2.01 wird kein NDS mehr unterstützt „*The gbox is NOT supporting nds cards any more.*“

Den Grund könnt ihr im Vorwort 2 lesen.

Getestet auf dbox2(PPC), VDR(intel x86) und Win32(intel x86)

	6MHz test	ECM test	EMM test
Irdeto 1.x GW	OK	OK	OK
Irdeto 6in1 Fun	OK	OK	OK
Irdeto DS9 Piccard2	OK	OK	-
Irdeto Fun C/D383Z	OK	OK	OK
Irdeto Fun C/D/F384Z	OK	OK	OK
Irdeto original C/D/F383Z	OK	OK	OK
Irdeto original C/D/F384	OK	OK	OK
Irdeto original A01/K01/S01	OK	OK	OK
Irdeto original A02/K02/S02	OK	OK	OK
Irdeto2 original	OK	OK	OK
Seca Fun	OK	OK	OK
Seca GW	OK	OK	OK
Seca original	OK	OK	OK
Seca2 original	OK	OK	OK
Viaccess Fun	OK	OK	OK
Viaccess GW	OK	OK	OK
Viaccess original	OK	OK	OK
Viaccess2 original	OK	OK	OK
Nagravision Fun	OK	OK	OK
Nagravision original ROM2	OK	OK	OK
Nagravision original ROM3	OK	OK	OK
Nagravision original ROM7	OK	OK	OK
Nagravision original ROM10	OK	OK	OK
Nagravision original ROM11	-	OK	OK
Cryptoworks original V1	OK	OK	OK
Cryptoworks original V2	OK	OK	OK
Cryptoworks original V3	OK	OK	OK
Conaxis Fun-NagraInConax	OK	OK	-
Conax original	OK	OK	OK
NDS original	OK	OK	OK

(*) Bei der dbox2 können 3.579MHz und 6.000MHz Oszillatoren für alle systeme benutzt werden. Die Software ermittelt die Baudrate, den Kartentyp, und führt alle Einstellungen automatisch durch. Ein 6MHz ist zu bevorzugen, weil die Karte damit 68% schneller arbeitet, schnelleres umschalten, etc. Fast alle Karten (außer Nagra ROM11) laufen auf 6MHz absolut stabil.

Beim PC mit VDR ist es zu empfehlen am sc8in1 mehrer Quarze nach Bedarf zu kombinieren. Im Auslieferungszustand sind vorhanden
2*6.000MHz, 2*3.686 und 4*3.579MHz

Folgende systeme arbeiten mit den daneben stehende Quarze.

Irdeto	6.0000 MHz
Seca	3.5795 MHz
Viaccess	3.5795 MHz
Nagravision	3.6864 MHz
Cryptoworks	3.5795 MHz
Conax	3.5795 MHz
NDS	3.5795 MHz

In der gbox ist ein Software-Blocker integriert, der es ermöglicht EMM's zu blocken. Es können nach adressierung EMM Gruppen geblockt werden, EMM's für alle Karten, für SharedAddress oder UniqueAddress. Hierfür ist die softcam.cfg zu konfigurieren.

Kill gbox

Um die gbox zu beenden, eine datei /var/tmp/gbox.kill erstellen, danach beendet die gbox sich selbst.

EMM's manuell zum Multicam schicken

Um ein EMM zu irgendeiner karte zu schicken, wird dieses in einer Date /var/tmp/gboxemm.txt geschrieben, im Format:

nr>emm ...

nr ist in Hex, also z.b card12 = 0C

gezählt wird in folgender Reihenfolge:

slot1=1 in hex also 01

slot2=2 in hex also 02

com1-card1=3, in hex also 03

...

com1-card8=10, in hex also 0A

com2-card1=11, in hex also 0B

...

com2-card8=18, in hex also 12

Beispiel (löscht chid EMM für PW Karte in com2 card01):

```
0B>01 01 00 00 00 6B D0 00 00 00 04 65 05 01 82 00 98 DE DF D3 7A 73 BB D2 B9 21 6D D3 0E EF A2 86
B8 AC DB A9 28 D5 F5 11 63 9A BD 8E DB FB 2F C9 8C 07 53 93 38 79 0B 75 C4 4D 7A EC 16 36 A7 E8
7F 21 3B 16 1E A8 50 21 B1 C1 70 6A 05 6D B5 55 54 14 34 4C 7D 76 A7 CB E8 FA A2 48 E9 E3 97 F3 86
18 65 3B C4 E2 FB 60 5B 93 EC 36 AD 64 FB 50 3D
```


NET Mode

Der NET-mode ist gedacht für:

1. Entwicklung/Forschung (implementiert)

Die dbox2 schickt jedes ECM über das Netzwerk zum PC, dieser führt die Entschlüsselung durch, schickt die CW's zur dbox2, die diese für die Decodierung verwendet.

* Im subdir /net-mode/ befindet sich eine kleine demo soft, c source und exeutable (cygwin)

2. dbox2 Vernetzung (implementiert)

Vernetzung mehrerer dbox2 (Wohnzimmer, Schlafzimmer, Nachbarin, ...) um mit einer Karte alle Boxen zu betreiben, bzw. das alle dboxen alle verfügbare Karten (Irdeto2, Seca2, ...) benutzen können.

3. CW&Card Sharing (in developement)

Wie punkt1, nur anstatt ein PC wird ein Server benutzt, mit dem übers Internet kommuniziert wird.

Für der NET mode muss UDP initialisiert werden L: { 01 } PC_IP

Server mode: Decode ECM

Decodiert ein ECM mit dem emu, antwortet mit den CW's.

Befehl: 0x42 0x15 Client_IP Len CaID PID ECM_data ...

Antwort(succes) 0x41 0x15 0x10 CW0 ... CW1 ...

Antwort(failed) 0x41 0x15 0x00

Client mode: Init NET mode

NET mode Init, EMU wird ausgeschaltet, dbox2 wird zum client, die ECM's gehen zur angegebenen IP (Server).

Befehl: 0x42 0x12 0x04 Server_IP

Antwort(succes) 0x41 0x12 0x00

Um nach jedem umschalten in den Net mode zu gelangen, muss in der gbox_cfg:

G { 02 } // Pure NET Client

eingetragen werden.

Multipid logger

Im zip befindet sich der MultiCaID/MultiPID Logger von trilu. Um ihn benutzen zu können muss im gbox_cfg die Einstellung L: { 01 } PC_IP eingetragen sein, um den UDP Port zu initialisieren.

Die implementierten UDP (port:8017) Befehle sind:

Befehl: "B" 0x42
Antwort: "A" 0x41
LogDaten: L CaID (2bytes) Pid(2bytes) Len(2bytes) ... Daten n=Len

Struktur

Befehl	Type	Länge	Daten=Länge
0x42	0x??	0xlen	Daten

Get cat

Befehl: 0x42 0x01 0x00
Antwort: 0x41 0x01 0xLN
LN Anzahl der Bytes im CAT.

Get pmt

Befehl: 0x42 0x02 0x00
Antwort: 0x41 0x02 0xLN
LN Anzahl der Bytes im pmt.

Get selected pids for logging

Befehl: 0x42 0x03 0x00
Antwort: 0x41 0x03 0xLN 0xA1 0xA2 0xB1 0xB2

LN/2 Anzahl der pids.

A1A2 first pid

B1B2 second pid

...

..

Falls keine pids selektiert sind: 0x41 0x03 0x00

Selected pids for logging

Befehl: 0x42 0x04 0xLN 0xA1 0xA2 0xB1 0xB2

Antwort: 0x41 0x04 0x00 (OK)

LN/2 Anzahl der pids.

A1A2 first pid

B1B2 second pid

...

..

Get available pids for logging

Befehl: 0x42 0x05 0x00

Antwort: 0x41 0x05 0xLN 0xA1 0xA2 0xA3 0xA4 0xA5 0xB1 0xB2 ...

LN/5 Anzahl der pids.

A1A2 first CaID

A3A4 first pid

A5 Type EMM=0 ; ECM=1

..

Falls keine pids da sind (FTA): 0x41 0x05 0x00

Start Logging

Befehl: 0x42 0xFE 0x00

Antwort: 0x41 0xFE 0x00

Stop Logging

Befehl: 0x42 0xFF 0x00

Antwort: 0x41 0xFF 0x00

Es können nur die pids geloggt werden die ich ausgelesen und gemapped habe, also cat und pmt sind informativ, benutze die 0x05 um auszulesen was da ist.

Nur bei log_off Befehle schicken !

Nach jeden Befehl soll eine Antwort kommen. Beim Log_off Befehl muss ich immer zwei log_off's schicken um eine Antwort zu bekommen, könnte aber auch an meiner test Software liegen.

Im Verzeichnis befindet sich der Logger von trilu, dieser kann einzelne pids, oder multiple pids (ecm und/oder emm) loggen.

Card sharing

Das cwsharing supportet das lokale Netz sowie das Internet. Es wird ein peer-to peer netz aufgebaut, wobei alle dbox-en direkt miteinander kommunizieren. Dies ist aus Gründen der Sicherheit und Schnelligkeit so gewählt worden.

Beim cwsharing ist es egal welche dbox2 welche Karte(n) hat, und wer welchen Sender zur Zeit benutzt.

Das cwsharing ist ziemlich optimiert, und pro Karte sollten auch mehr als 100 Benutzer kein Problem sein.

Alle Kartentypen werden beim sharing unterstützt.

Einige Eigenschaften:

1. Der Eigentümer hat absolute Priorität, egal wie viele Benutzer an seiner Karte(n) dranhängen, das zappen sollte nicht beeinträchtigt werden.
2. Jede dbox2 kommuniziert direkt nur mit den dbox-en die in der cwshare.cfg eingetragen sind.
3. nach der eingabe der IP in cwshare.cfg, folgen 2 codes
code1 - diese IP darf meine Kartendaten an x Freundeskreise weitergeben.
code2 - diese IP darf meine Fremde Kartendaten an x Freundeskreise weitergeben.

Wenn der User A eine Cryptokarte hat, und der User B sein Freund ist (und in der cwshare.cfg eingetragen ist), dann kann User B auch die CW karte benutzen. Wenn jetzt User C ein Freund von User B ist, darf er auch die Karte von A benutzen, falls dieser das erlaubt hat $code1 > 1$. Wenn jetzt User D ein Freund von User C ist, darf der auch die Karte von A benutzen (wenn dieser das erlaubt hat $= code1 > 2$) und so weiter.

Wichtig ist: das jede dbox2 NUR mit den befreundeten kommuniziert, die ecm's/cw's gehen von dbox2 zu dbox2 bis die das Ziel erreicht haben.

4. Alle Daten sind vollständig verschlüsselt, nicht einmal ein Header ist zu erkennen.

Install:

- a. Wer ins Internet will, soll mal bei dyndns.org vorbeischaun und sich ein Account einrichten. Im Router dyndns aktivieren, damit sich der Router immer bei dnydns.org meldet wenn der online geht.
- b. Unbedingt im Netzwerk config der dbox den Gateway und Namensserver eintragen.
- c. Am Router müssen die entsprechenden Ports zur dbox2 umgeleitet werden, damit diese die eingehenden Daten auch empfangen kann. Das nennt man "port forwarding". Die ist im Lokalen Netz nicht nötig ! Nur fürs Internet!
- d. Im lokalen Netz, müssen in jeder cwshare.cfg alle Boxen eingetragen werden, damit auch gewährleistet wird das sich diese auch immer finden. Danach ist es egal welche zuerst hochfährt, die nächste findet immer die, die schon da sind.
- e. Unbedingt im MIX_MODE und (NON_AU oder AUTO_AU) betreiben !!! Wenn das AU auf ON ist, werden zu viele EMM's zur Karte geschickt, was das sharing belastet. Auto AU schaltet das AU bei Bedarf an, und schaltet es nach einem erfolgreichen decode wieder aus. Es wird jedoch empfohlen das CW sharing mit NON_AU zu betreiben, weil sich das AU auch einschalten wird falls zum Beispiel auf PW direkt gezappt wird, weil dort nicht zu erkennen ist ob ein AU erforderlich ist oder nicht.

Hier mal ein Beispiel für 3 Boxen im lokalen Netz (zu hause) wobei eine noch am Internet dran ist.

```
# cwshare.cfg --- dbox1 --- Internet und Lokales Netz
#
# My dbox2 (local IP =192.168.0.6)
#           password
M: { mydbox2.homeip.net { AA242456 }}
#
#   Internet Friends      RX TX_Port password code
D: { friend1.homeip.net  { 8200 8200 { B142AB11 { 5 5 }}}}
D: { friend2.homelinux.net { 8200 8200 { 81BFF901 { 5 5 }}}}
#
#   other local boxes
D: { 192.168.0.51        { 8200 8200 { AB333441 { 5 5 }}}}
D: { 192.168.0.52        { 8200 8200 { BA334B24 { 5 5 }}}
```

```
# cwshare.cfg --- dbox2 --- Lokales Netz
#           password
M: { 192.168.0.51        { AB333441 }}
#
#   other local boxes
D: { 192.168.0.6         { 8200 8200 { AA242456 { 5 5 }}}}
D: { 192.168.0.52        { 8200 8200 { BA334B24 { 5 5 }}}
```

```
# cwshare.cfg --- dbox3 --- Lokales Netz
#           password
M: { 192.168.0.52        { BA334B24 }}
#
#   other local boxes
D: { 192.168.0.6         { 8200 8200 { AA242456 { 5 5 }}}}
D: { 192.168.0.51        { 8200 8200 { AB333441 { 5 5 }}}
```

CWserver

Falls die gbox nicht auf einem Reciever läuft, und die /var/tmp/pmt.tmp nicht gefunden wird, arbeitet diese als CW Server. Benutzt wird ein Phoenix am com1, oder ein sc8in1 für bis zu 8 Karten am com1.

Off-Air Auswertung

Um geloggte ECM's und/oder EMM's auszuwerten oder zu eine Karte zu schicken, wird die gbox mit 2 argumenten aufgerufen:

```
gbox file caid
```

```
Beispiel: gbox seca-ecm.txt 0100
```

```
Beispiel: gbox seca-ecm.bin 0100
```

In der gbox_cfg kann bestimmt werden ob durch den Emu eine Auswertung erfolgen soll (Mode=emu) oder die ECM's und/oder EMM's zu Karte geschickt werden sollen (Mode=Softcam)

Die ecm's und emm's können im text/bin file gemischt sein, falls der log im gbox multipid mode erstellt wurde, können auch ecm's und emm's verschiedener systeme im log file vorhanden sein. Falls am com1 ein sc8in1 vorhanden ist, werden die ecm's und emm's den entsprechenden Karten zugeleitet.

Bei Nagra EMM auswertung, muss sich in den log file auch ein ecm befinden, weil dieses zur überprüfung der keys benutzt wird.

GSMS

gsms (gbox shot messaging service) ermöglicht Nachrichten zu einzelnen oder allen boxen im Netz zu verschicken. Dazu wird nur eine Datei gsms.txt nach /var/tmp kopiert, diese wird danach eingelesen, die Daten verschickt und die datei gelöscht.

Falls die ip als 0.0.0.0 eingegeben wird, wird die Nachricht an alle dbox-en die in cwshare.txt angegeben sind, verschickt. Das automatische darstellen der gsms Nachrichten auf dem Bildschirm ist nur unter Neutrino und Enigma möglich. Unter VDR müssen diese mit einem plugin oder was anderes (was folgen wird) dargestellt werden.

```
192.168.0.51 0 this is a normal test message/this is the second line//and this the fourth
someip.homelinux.net 0 this is also e normal message
192.168.0.52 1 this will display a popup that will require a/OK pressing, to disappear
```

Es ist auch möglich gsms an Boxid's zu schicken.
ABCD 1 dies ist ein test

Install

Die Dateien von /keys/ (irdeto, seca, via, ...) müssen in /var/keys/ kopiert werden.

Ein Verzeichnis /var/tmp muss auch vorhanden sein, dort werden temporäre Dateien abgespeichert.

Für verschiedene Einstellungen bitte dis gbox_cfg und die softcam.cfg ansehen

VDR:

Es werden die üblichen Treiber benutzt, mit gepatchter Root.

VDR mit dem patch /x86/linux/gbox_vdr.diff patchen

Aufruf der gbox:

```
./vdr
./gboxx86
```

Bei multiple Karten muss die gbox mehrmals aufgerufen werden. Es muss für jede Karte eine eigene Kopie der gbox (inclusive keys und config files) erstellt werden.

```
/var/keys/gbox0/gboxX86
/var/keys/gbox1/gboxX86
/var/keys/gbox2/gboxX86
etc ...
```

Eine gbox kann ein com port (mit Multicam oder sc8in1) verwalten. Falls 2 oder mehr Karten im System sind, können alle ein eigenes Multicam oder sc8in1 zugewiesen bekommen, die Smartcards werden dann von allen dvb-s karten benutzt.

Nur eine gbox kann das cwshare managen. Nennen wir diese master gbox, die anderen slave gbox.

Im Master cwshare.cfg alle anderen Freunde eintragen, und auch die restlichen dvb-s Karten im system. Dabei sollen alle slave gboxen einen eigenen RX UDP port haben:

Beim Master (4 Karten Beispiel):

```
M: { 192.168.0.10 { 12345678 }  
D: { 192.168.0.10 { 8000 8001 { 12345670 { 5 5 } } } }  
D: { 192.168.0.10 { 8000 8002 { 12345671 { 5 5 } } } }  
D: { 192.168.0.10 { 8000 8003 { 12345672 { 5 5 } } } }
```

Bei den slaves:

```
M: { 192.168.0.10 { 12345670 }  
D: { 192.168.0.10 { 8001 8000 { 12345678 { 5 5 } } } }
```

```
M: { 192.168.0.10 { 12345671 }  
D: { 192.168.0.10 { 8002 8000 { 12345678 { 5 5 } } } }
```

```
M: { 192.168.0.10 { 12345672 }  
D: { 192.168.0.10 { 8003 8000 { 12345678 { 5 5 } } } }
```

Das softcsa ist nicht implementiert. Ich habe keine skystar und kann das somit auch nicht implementieren/testen.
Vom

/dev/dvb/adapter0/ca0

benutze ich nur

```
ioctl(ca_handle,CA_SET_DESCR,&ca_descr);
```

Wenn jemand interesse hat, kann er ja ein modul schreiben das ein device /dev/dvb/adapter0/ca0 öffnet und das softcwwrite dort implementieren.

dbox2

Die gbox ist ein daemon,sie darf NICHT beim umschalten gekilled werden. Sie wird einmal aufgerufen, und das Kanal wechsel wird getriggert durch das schreiben der PMT nach /var/tmp/pmt.tmp

Windows

Die dbox arbeitet unter Windows nur im indirekten Modus. Als bindeglied zwischen DVBCore Anwendungen und der gbox wird das plugin gboxsfriend.dll benutzt, das dvbcore kompatibel ist. Dieses Schickt beim Kanalwechsel die PMT&CAT zur gbox, und später auch die ECM's und EMM's

Für Multidec compatible Anwendungen muss noch der wrapper DVBCore.dll benutzt werden, der es ermöglicht ein dvbcode kompatibles Plugin mit einer Multidec Api Anwendung zu benutzen.

DVBCore.dll wird in dem Multidec Verzeichniss kopiert, und gboxsfriend.dll in das Verzeichniss \Plugins das im Multidec Verzeichniss erstellt werden soll.

Das zappen ist unter Multidec/ProgDVB ist langsamer, weil die pids von Multidec/ProgDVB zuerst zum wrapper gehen, dann zu gboxsfriend und erst danach zur gbox. Um das zu beschleunigen müsste eine multidec kompatible gboxsfriend.dll geschrieben werden.

Season emulator

Die gbox kann auch als Season Emulator benutzt werden. Dazu muss in der gbox_cfg W: { 01 xx } eingetragen werden, xx ist der com port an dem das Season angeschlossen ist. Zur Zeit wird nur eine Emulation aller Systeme im Irdeto nonZ allcam unterstützt.

Die Server kann ein PC (Windows/Linux) oder eine dbox (ppc) sein. Alle Karten die am System angeschlossen sind (auch ein sc8in1) können durch das Season Interface benutzt werden, auch cwsharen usw ...

Der Season emulator benötigt einen eigenen gbox task, also müssen für jedes Season eine eigene gbox gestartet werden. Die interne gbox kommunikation erfolgt wie bei mehreren dvb-s karten über UDP.

Die Konfiguration der gbox

Nachdem ihr jetzt die grundsätzliche Funktion der gbox kennt, möchte ich jetzt mal auf die Konfiguration und Einstellmöglichkeiten eingehen

Gbox_cfg

```
# Trace/Debug
# xx yz ; xx=00 no konsole output
# xx yz ; xx=01 konsole output
# xx yz ; y=0 debug output (don't use)
# xx yz ; y=1 no debug output
# xx yz ; z=0 ouput to konsole
# xx yz ; z=1 output to /var/tmp/debug.txt
# xx yz ; z=2 Output to UDP (to capture with gboxt)
```

Z: { 01 12 } 192.168.1.29 8024

In diesem Eintrag kann die log Ausgabe der gbox eingestellt werden, die dann mittels tracen (z.b. gboxt.exe) auf dem Rechner dargestellt werden kann. Dabei ist zu beachten das die IP, die vom Rechner sein muß wo das Traceprogramm läuft. Nach der IP wird der Port angegeben auf dem die Daten gesendet werden. Standartmäßig ist hier 8024 angegeben, den man auch ändern kann fals dies nötig ist.

```
# Task type
# 00 ** normal mode gbox mode (no season, use Multicam instead)
# 01 ** Season Emulation in a Irdeto allcam chid 0602 nonZ
# 10 ** Season Emulation in a Irdeto allcam chid 1702 nonZ
# 11 ** Season Emulation in a Irdeto allcam chid 1702 Z
# 12 ** Season Emulation in a Irdeto allcam chid 1722 Z
# 13 ** Season Emulation in a Irdeto allcam chid 1762 Z
# ** *1 Com port to use for season emulatiom (1 or 2)
# ** 0* Use CTS f r RST detection
# ** 1* Use DCD f r RST detection
# ** ** 9600 = Baudrate to be used (working only on dbox2)
```

W: { 00 00 } 9600

Mit diesem Eintrag kann die gbox ein Season-Interface ansteuern. Du kannst dann mittels Season auch cw's zu anderen Boxen weiter leiten die keinen Netzwerkzugang besitzen. Beim betreiben des Season über eine dbox2 ist zu beachten das die gbox 2 mal gestartet werden muß. Einmal wie üblich für die dbox2 und ein zusätzliches mal für das Season. Hierfür erstellt ein zweites Verzeichnis

z.b. /var/keys/season

in diesem Verzeichnis werden werden alle Dateien die auch unter /var/keys liegen verlinkt, bis auf die gbox_cfg und die cwshare.cfg die in beiden beiden verzeichnissen getrennt voneinander existieren müssen. Hier mal ein Beispiel wie man die links zu erstellt

```
cd /var/keys/season
```

```
ln -s /var/keys/gbox gbox
```

```
ln -s /var/keys/irdeto irdeto
```

```
ln -s /var/keys/seca seca
```

```
...
```

```
..
```


Desweiteren ist anzumerken, das die zweite Instanz der gbox nur durch einen Eintrag in der cwshare.cfg bei der dbox2 angesprochen wird. Daher muß diese auch in beiden Verzeichnissen vorhanden sein und darf nicht verlinkt werden. Dabei ist zu beachten, das auch zwei unterschiedliche ports verwendet werden müssen. Hier mal ein Beispiel:

```
/var/key/cwshare.cfg
```

```
# My dbox2 password
M: { 192.168.0.51 { F37764E1 }}

# Friends password code
D: { 192.168.0.51 { 2499 2299 { 30030002 { 1 1 }}} }
# here all other boxes, local and internet
```

```
/var/key/season/cwshare.cfg
```

```
# My dbox2 password
M: { 192.168.0.51 { 30030002 }}
D: { 192.168.0.51 { 2299 2499 { F37764E1 { 0 0 }}} }
```

```
# gbox working mode
```

```
# 00 pure Emu
# 01 pure SoftCam (when card inside), Emu (when no card inside)
# 02 pure Net Client
# 03 Mixed mode !!!!! always use this !!!!!
```

```
G: { 03 }
```

Hier kann man einstellen in welchem Modus die gbox arbeiten soll,

```
# Display ECM messages on Konsole
```

```
# 00 don't show any ECM's
# 01 show only valid ECM's
# 02 show valid and bad ECM's with complete data display
```

```
C: { 01 }
```

Hier kann man einstellen ob die ECM's im log angezeigt werden sollen

```
# Display EMM messages on Konsole
```

```
# *0 don't show any EMM's
# *1 show only valid EMM's
# *2 show valid and bad EMM's with complete data display
# *1* Log any tier change on PW-Nagra-17xx cards into /var/keys/pwupdate.log
```

```
M: { 00 }
```

Hier kann eingestellt werden ob EMM's im log angezeigt werden sollen. Desweiter besteht hier die möglichkeit bei PW-Nagra Karten eine änderung der Tiers mit zu loggen, dies wird dann in /var/keys/pwupdate.log geschrieben.

```
# AutoUpdate
```

```
# 00 is ignored (default)
# 01 enable AU
# 02 Auto AU !!!!! always use this !!!!!
```

```
A: { 02 }
```

Hier kann eingestellt werden ob Autoupdate gemacht werden soll oder nicht. Je nachdem was eingestellt wird, werden EMM's zur Karte geschickt oder auch nicht. Man sollte am besten 02 eintragen da hier das AU dann

Automatisch erfolgt wenn nichts mehr entschlüsselt wird. Wenn ihr Card sharing betreibt ist hier aber das zu beachten was schon im Kapitel Card Sharing beschrieben ist

```
# Key update
# 01 update only new keys, default setting
# 02 update all keys (used for valid PMK checking)
```

U: { 01 }

Hier kann eingestellt werden ob in den Key-dateien nach AU nur Keys eingetragen werden die neu und noch nicht vorhanden sind, oder ob alle Keys geschrieben werden sollen z.b. um die Gültigkeit der Masterkeys zu überprüfen.

```
# Hash pids
# 00 hash all pids, DO NOT USE THIS OPTION, IT IS ONLY FOR TESTING STUFF
# 01 hash only pids with implemented algos and available keys !!!!! always use this !!!!!
```

H: { 01 }

Last die Standarteinstellung. Dieser Eintrag ist nur für Testzwecke

```
# Key files folder
# 00 keyfiles in /var/keys
# 01 keyfiles in /var/tmp
```

T: { 01 }

Hier kann angegeben werden in welchem die Keyfiles zu finden sind. Hier sollte als Arbeitsverzeichnis bei der dbox2/dreambox tmp angegeben werden. Da sonst bei jedem AU der Keyfiles ins Flash geschrieben wird. Eine saubere Lösung ist wenn die Keyfiles beim Start ins tmp kopiert werden und beim Runter fahren verglichen werden und gegeben falls bei Änderung zurück kopiert werden. Somit wird ein unnötiges schreiben ins Flash verhindert.

```
# On Screen Display for the Multysystemcam on com1/com2
# 00 No OSD (default)
# 01 display Smartcard messages in Neutrino
# 02 display Smartcard messages in Neutrino new nhttpd format
# 03 display Smartcard messages in Enigma
#                               mypass (use user=root and pass=mypass for nhttpd)
```

O: { 02 } 192.168.1.202

Für das OSD der gbox bitte hier den entsprechenden Modus auswählen und die IP eurer Box eintragen. Wenn in eurem Image der nhttpd Pass geschützt ist muß hier auch noch ein Passwort vom User root angehanden werden

```
# Logger and NET-mode UDP port Init
# 00 do not init UDP port
# 01 Init UDP port for IP, port, UDP
```

L: { 00 } 192.168.1.29 8017

Hier gelten die Einstellungen die schon unter NET Mode und Multipidlogger beschrieben sind

```

# demux, multicam/comport configuration
# *x ** ; x = device nummber to be used (0= default)
# D* ** ; force use of /proc/bus/gtx for cw write operations
      (in the dbox2 the gbox writes directly the CW's, 00 is the
      recommended mode ! No drivers are required) use D* ** ONLY if
      you have Problems. 0* ** should be prefered !!!
# F* ** ; write CW's in both devices (used for PVR in STB04xxx devices)
# ** y* ; y=0001b (0x1*) use slot1 as Multicam (only on STB04xxx)
# ** y* ; y=0010b (0x2*) use slot2 as Multicam (only on STB04xxx)
# ** y* ; y=0011b (0x3*) use slot1 and slot2 as Multicams (only on STB04xxx)
# ** *z ; z=0001b (0x*1) use com1 as Multicam
# ** *z ; z=0010b (0x*2) use com2 as Multicam
# ** *z ; z=0011b (0x*3) use com1 and com2 as Multicams
# ** yz ; yz=00110011b (0x33) use slot1+slot2+com1+com2 as Multicams
# ** *z ; z=0100b (0x*4) invert CTS detection on com1
# ** *z ; z=1000b (0x*8) invert CTS detection on com2
# example: yz = 0x0A = 00001010 (use com2 with inverted CTS detection line)

```

V: { 00 02 }

In diesem Eintrag kann im ersten byte angegeben werden wohin die cw's geschrieben werden sollen. Im Zweiten byte wird die Konfiguration/Aktivierung für das Multicam bzw. die internen Slots der Dreambox vorgenommen. Hier ist an zu merken das die angaben in hex gemacht werden und somit eine kombination aus mehreren Funktionen möglich sind. z.b. 02 = com2 als Multicam und 08 invertiert CTS detection. Dann ergibt 02 + 08 = 0A

```

# =====
# extra options to override default seting in case you experiance Problems,
# normaly these setings are not required.
#

```

Die folgenden Funktionen werden im Normalfall nicht benötigt und sind nur bei Problemen zu benutzen um die von der gbox vorgegeben standart Einstellungen zu umgehen, daher sind sie auch mit # auskommentiert. Und stehen nur der Vollständigkeits halber hier und ich werde versuchen diese so gut es mir möglich ist zu dokumentieren. Wer hier fummelt sollte eh wissen was er macht. * smile *

```

# cpu, api, cw write method
#
# 00 auto
# 01 80X86
# 02 MPC823
# 03 IBM STB04xxx
# 00 auto
# 01 ibm
# 02 api2
# 03 api3
# 00 cw write style = use internal routine (only dbox2)
# 01 cw write style = CA_SET_DESCR type1 (dbox2, vdr)
# 02 cw write style = CA_SET_DESCR type2
# 03 cw write style = CA_SET_DESCR type3
# 04 cw write style = 16 byte /proc/bus/gtx
# 05 cw write style = 4096 byte /proc/bus/gtx, cw offset=0x04D0
# xx write cw in 1st device with index=xx
# yy write cw in 2nd device with index=yy
# 01 means with .index=0 (default)
# 02 means with .index=1
# 03 means with .index=0 and .index=1
# .. any bitwise combination ...
# FF means with all 8 indexes (0-7)

```

#J: { 01 00 00 01 01 }

Mit dem ersten byte kann die CPU eingestellt werden die verwendet wird.
Mit dem 2byte kann die api bestimmt werden.
Mit dem dritten, vierten und fünften byte kann der cw_write style geändert werden.

```
# Files, folders, devices
#      folder/device          comment
#
#F: { 01 } /var/tmp/pmt.tmp    // pmt file
#F: { 02 } /var/tmp/emm.info   // key update log file
#F: { 03 } /var/tmp/ecm.info   // ecm info file
#F: { 04 } /var/tmp/atack.txt  // atack info file
#F: { 11 } /var/keys          // folder for config files
#F: { 12 } /var/tmp           // folder for temporary file
#F: { 13 } /var/tmp           // folder for keyfiles
#F: { 21 } /dev/dvb/adapater0/ca0 // ca device
#F: { 22 } /dev/dvb/adapater0/dvr0 // dvr device
#F: { 23 } /dev/dvb/adapater0/demux0 // demux device
#F: { 31 } /dev/dvb/adapater0/ca1 // second ca device to write cw's (recording)
#F: { 41 } /dev/sci0          // STB04xxx slot1
#F: { 42 } /dev/sci1          // STB04xxx slot1
#F: { 43 } /dev/tts/0         // rs232 com1
#F: { 44 } /dev/tts/1         // rs232 com2
```

mit diesen Einträgen können die Verzeichnisse die von der gbox benötigt werden angepaßt/geändert werden.

```
# For overclocking freaks, Clocking internal STB04xxx slots with:
# 1=7.80MHz 2=6.30MHz 3=5.25MHz 4=4.50MHz 5=3.94MHz 6=3.50MHz 7=3.15MHz
# Irdeto works fine on 7.8MHz, all other fine on 6.30MHz or 5.25MHz
# The Dream has not a clean clock signal, some cards that are sensitive will
# refuse to work even on 3.5MHz, thats why 3.15MHz is the default value there.
# On other STB04xxx recievers the same cards work with 6.30MHz
#
#F: { 51 } 02                // irdeto 6.30MHz
#F: { 52 } 03                // seca 5.25MHz
#F: { 53 } 03                // viaccess 5.25MHz
#F: { 54 } 03                // nagra 5.25MHz
#F: { 55 } 03                // crypto 5.25MHz
#F: { 57 } 03                // conax 5.25MHz
#F: { 58 } 03                // nds 5.25MHz
```

mit diesen Einträgen können die Taktraten der internen Slots der Dreambox eingestellt/übertaktet werden.

Softcam.cfg

```
# SoftCam EMM Log configuration
#
# Save EMM data
# 00 Don't save anything (default)
# 01 Save processed EMMs (/var/tmp/emmproc.bin)
# 02 Save rejected EMMs (/var/tmp/emmblock.bin)
# 03 Save both EMMs
```

L: { 00 }

Hier kann man Einstellen ob die EMM's die zur Karte gehen und/oder von der Karte abgewiesen werden in einer Datei gespeichert werden sollen

```
# Check additional OSC Frequency !!! ONLY FOR DBOX2 !!!
# 00 No Check (default)
# 01 Check on, Value in KHz , example 8MHz=08000
```

F: { 00 } 00000

Überprüfung der Freq für die Karte

```
# Check additional Baurate for ATR !!! ONLY FOR DBOX2 !!!
# 00 No Check (default)
# 01 Check on, Value in bit/s, example 81861
```

B: { 00 } 0000

Überprüfung der Baudrate für die Karte

```
# Enable TPScript Support
# *0 TPScript off
# *1 TPScript on (default)
# 1* don't use PTS on Viaccess cards
# 2* don't use PTS on Cryptoworks cards
# 3* don't use PTS on Viaccess and Cryptoworks cards
```

Y: { 01 }

Hier kann die Unterstützung für TPScript Karten aktiviert werden.

```
# Empty RS232 buffer before writing
# 00 No reset
# 01 Reset Buffer (default)
```

E: { 01 }

RS232 Puffer leeren vor dem schreiben, sollte beim Sharen unbedingt eingeschaltet sein.

```

# SoftCam Blocker/Filter definition
#
# 00 = Allow EMM (default for all)
# 01 = Block EMM addressed to:
#   all card
#   | shared address/provider
#   | | unique address/hex_serial
# -----

```

```

I: { 00 00 00 } Irdeto   All SA UA
V: { 00 00 00 } Via     - SA UA
S: { 00 00 00 } Seca    - SA UA
N: { 00 00 00 } Nagra   All SA UA
C: { 00 00 00 } Crypto  All SA UA
O: { 00 00 00 } Conax   - SA UA
D: { 00 00 00 } NDS     All SA UA

```

Hier kann der Blocker aktiviert werden. Das erste byte steht für all, also EMM's die an alle Karten gehen. Das zweite byte steht für Shared address was soviel bedeutet wie Karten Gruppen und das dritte byte für unique address was für an eine spezielle Karte steht in unserem fall also EMM's die speziell an unsere karte in der box gerichtet sind. Als Beispiel mal unsere PW Karten: Das erste byte würde die D0, das zweite Byte die D2 und das dritte byte die D3 blocken.

Timeout Definitions

```

#
T: { 00 02 01 } ATR timeout (normal*X), instruction timeout (normal*X)

```

Hier kann der timeout für den ATR eingestellt werden. Am besten so lassen. Wenn ihr allerdings Probleme mit der Erkennung eurer Karte im Multicam habt. Kann es euch helfen, wenn ihr diesen Eintrag etwas abändert z.B so T: { 00 06 01 } damit wurden dann auch Karten erkannt die vorher nicht wollten. Also bei Problemen einfach etwas mit diesem Eintrag probieren.

```

# com1/com2 delays, all delays to be entered in decimal !
#

```

```

# Do NOT touch !!!!!!!!!!!!!!!!!!!!!

```

```

# all timings should be 0, change only if you have problems.

```

```

T: { 01 00000 000 } Irdeto   : byte delay (1e-6 s), instruction delay (1e-3 s)
T: { 02 00000 000 } Seca
T: { 03 00000 000 } Viaccess
T: { 04 00000 000 } Nagra
T: { 05 00000 000 } CryptoWorks
T: { 06 00000 000 } Geldkarte
T: { 07 00000 000 } Conax
T: { 08 00000 000 } NDS

```

Hier können die delays für den Comport eingestellt werden. Am besten so lassen

```

# Use Crypto card bug to decode any CW, no matter of chid
#
# Cxxx configuration (is aktiv for the provider only if a config line is present)
# (ECM sample to be used to get the cw's with Cxxx)
#
# Provider                provider    chid - that is present on the card for that provider
# ||                      ||          || ||
# UPC-Astra
8: { 8D } A4 4C 00 00 2E 20 00 2B 83 01 8D 8C 03 20 01 00 8E 02 00 00 C5 01 00 DB 10 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 DF 08 00 00 00 00 00 00 00 00
# Digiturk
8: { C1 } A4 4C 00 00 31 28 00 2E 83 01 C1 8C 03 89 19 19 8E 02 00 00 8F 01 A9 91 01 55 DB 10 00 00 00 00
00 00 00 00 00 00 00 00 00 DF 08 00 00 00 00 00 00 00 00

```

mit dieser Einstellung kann der Bug der Cryptokarten benutzt werden

```

# Use constant CW's for
#   caid  pid  CW0 ...          CW1 ...
0: { 01 00 56 00 { 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF }

```

hier können Konstante cw's eingetragen werden die, die Box auf bestimmten Kanälen senden soll

cwshare.cfg

Ignore user ID in case you want to ignore some bad cards (max 255 Y: entrys/lines)

Y: { 12 34 }

Mit diesem Eintrag könnt ihr spezielle ID's bei euch auf der Box sperren, wenn sie bei euch Probleme verursachen z.B Emukarten. Es können insgesamt 255 Karten hier gesperrt werden. Dafür einfach den Eintrag mit der nächsten ID, die ihr sperren wollt wiederholen.

allow maximum 06 share level for recieved card data

I: { 04 }

Hier könnt ihr die max. Sharetiefe einstellen die ihr empfangen wollt. In diesem Fall z.B. kommen nur Karten mit einer Distanz von 4 bei euch an. Alles was weiter weg ist wird ignoriert, selbst wenn sie bis zu euch kommen würden. Dies ist von Nöten wenn ihr z.B. die maximal von der gbox verwaltbaren Karten im share habt. Da sonst die stabilität leiden kann.

```
# 01 repeat EMM's
# 00 consider every EMM only once
# 01 restart pid on overflow
# 1* reset ENX on every channel change
# *1 check/reset ENX freezes on FTA
# *2 check/reset ENX freezes on PayTV
# *3 check/reset ENX freezes on FTA and PayTV
# 00 write nothing in atack.txt
# 01 write into atack.txt: password is wrong
# 02 write into atack.txt: ID unknown
# 04 write into atack.txt: IP is wrong
# 08 write into atack.txt: port is wrong
# 10 write into atack.txt: share.stat
# 20 create online.log for online/offline logging
# 1F schreibt alles ohne comeOnline/Offline meldungen
# 3F Schreibt alles. (kombiniert Bits für andere Kombinationen)
# 4000 send ecm again after 4 Seconds if no reply.
# 6000 resync decode after 6 Seconds if net decode failed
```

N: { 00 01 03 1F 4000 6000 }

Mit dem ersten byte kann das EMM Handling eingestellt werden. Ob nur neue EMM's verwendet werden sollen oder ob sie wiederholt werden sollen. Mit dem zweiten byte kann eingestellt werden ob der pid nach einem Überlauf neu gestartet werden soll. Das dritte byte ist für ENX geplagte bei der dbox sehr interessant, die gbox bitte die Möglichkeit Freezer zu erkennen und automatisch einen ENX-Reset aus zu führen. Mittels des vierten bytes wir eingestellt was in die atack.txt geschrieben wird. Nach dem vierten byte wird die Zeit in ms angegeben, wann ein ECM erneut ins Netz geschickt wird wenn, kein gültiges cw zurück kommt Die zweite eingestellte Zeit sagt aus wann neu synchronisiert werden soll, wenn kein gültiges cw aus dem Netz zurück kommt. Hier auch in ms angegeben.

Send ECM's at maximum 5 cards (please use this as default)

X: { 05 }.

Hier kann eingestellt werden wieviele Karten für ein CW angefragt werden. Bitte benutzt dies als Standard einstellung.

Send ECM's in any case to these card ID's, even they are more then X:
some examples ...

G: { 17020000 1234 }
G: { 0B000000 ABCD }
G: { 05021510 AB34 }
G: { 0D03A000 A234 }

Hier könnt ihr spezielle Karten angeben, z.B. wenn ein Freund von euch BB aboniert hat oder öfters Direkt bestellt. Eingetragen wird die Card gefolgt von der BoxID, welche ihr euch aus der share.info suchen könnt. Das G: hat vor dem X: vorrang d.h wenn ihr von einem Provider z.b mehr Karten unter G: eintragt als ihr unter X: angegeben habt, dann sind auch diese alle in eurem share ohne das ihr das X: erhöhen müßt. Dadurch wird die Netzauslastung wesentlich reduziert und alles sollte stabiler laufen.

For W: please read the cwshare.txt, here only some examples
use card 1 only for the following pids
#W: { 01 02 02 } 1022 100A 100B 102B 1009 101D 1029 1014 1011 101B
don't use card 2 for the following pids
#W: { 02 03 03 } 1008 1016

0x=Slot Nummer (von 1 bis 8) auf der sich W: bezieht
0x=Level für Eigentümer
0=Karte nicht benutzen
1=Karte für alle pids benutzen
2=Karte nur für die folgenden pids benutzen
0x=Level für Netzwerk
0=Karte nicht freigeben
1=Karte für alle pids freigeben
2=Karte nur für die folgenden pids freigeben

W: { 0B 02 02 } 1008 100A 100B 102B 1009 101D 1029 1014

Dieser Eintrag ist besonders Wichtig. Hier wird eingestellt welche Pakete die Karte entschlüsseln kann und wie sie intern und extern verwendet werden. Das erste byte bezeichnet den Slot des Multicams (oben im Beispiel Com2 – Card1 also ein Standard Multicam), der in hex angegeben wird. Gezählt wird in folgender Reihenfolge:

slot1=1 in hex also 01
slot2=2 in hex also 02
com1-card1=3, in hex also 03
...
com1-card8=10, in hex also 0A
com2-card1=11, in hex also 0B
...
com2-card8=18, in hex also 12

mit dem zweiten byte kann der Level für den Eigentümer und mit dem dritten byte die für das Netzwerk eingestellt werden. Hiernach werden die Pids angegeben die auf der Karte freigeschaltet sind.

Nimmt euch mal ein wenig Zeit und konfiguriert mal das W: , eure Karten, eure dbox, und auch die Freunde im Netz werden euch dankbar sein da hierdurch das Netz entlastet wird.

Hierbei ist zu beachten, das die Sport Unterkanäle auf den Direkt Unterkanälen gesendet werden. Also sollten am besten auch die Direkt Unterkanäle mit in das W: aufgenommen werden wenn ihr Sport im Abo habt. Sonst werden die Unterkanäle nicht geshart bzw. werden sie nicht von eurer Karte beantwortet und kommen aus dem Netz.

Für die Leute die besonders viel Direkt bestellen wäre es vielleicht interessant nur die pids für Direkt in das W: aufzunehmen, somit würde die Karte von den normalen Anfragen entlastet und beantwortet nur die Anfragen für Direkt. Somit sollte das Sharen von Direkt stabiler laufen.

hier mal eine Liste der Pids die z.Z für PW aktuell und mir bekannt sind. Das C hinter den Einträgen bezeichnet Sat und das D Kabel. Die Liste enthält keinen Anspruch auf Vollständigkeit, da ich hier nur pids aufgelistet habe die mir genannt wurden, oder ich bisher gefunden habe.

1FFF alle Pids dieser Karte	1008 Premiere Start	100A Premiere 1	100B Premiere 2
102B Premiere 3	1009 Premiere 4	101D Premiere 5	1029 Premiere 6
1014 Premiere 7	1204 Prem Nostalgie C	1704 Prem Nostalgie D	102A 13 th Street C
152A 13 th Street D	100E Discovery Cannel C	150E Discovery Cannel D	1203 MGM C
1703 MGM D	1024 SciFi C	1524 SciFi D	1011 Sport 1 C
1511 Sport 1 D	101B Sport 2 C	151B Sport 2 D	1015 Beate Uhse TV
1017 Krimi C	1517 Krimi D	1010 Serie C	1510 Serie D
1035 Prem Austria	1206 Goldstar	1022 Disney Channel C	1522 Disney Channel D
1032 BigBrother C	101C FoxKids C	151C FoxKids D	1013 Junior C
1513 Junior D	1016 Heimatkanal	100F Classica	100C Animal Planet C
150C Animal Planet D	10A8 Hit 24 C	15A8 Hit 24 D	

10D0 10D1 10D2 10D3 10D4 10F0 10F1 10F2 10F3 10F4 10F5 Premiere Direkt C (D?)
 1201 1309 130A 130B Premiere Erotik C (D?)
 1298 1299 129A Blue Movie C (D?)

S: is the same as W:, just using the SID instead of the ECMPID

```
# 0x=Slot Nummer (von 1 bis 8) auf der sich W: bezieht
# 0x=Level für Eigentümer
# 0=Karte nicht benutzen
# 1=Karte für alle pids benutzen
# 2=Karte nur für die folgenden pids benutzen
# 0x=Level für Netzwerk
# 0=Karte nicht freigeben
# 1=Karte für alle pids freigeben
# 2=Karte nur für die folgenden pids freigeben
```

S: { 0B 02 02 } 0008 000A 000B 002B 0009 001D 0029 0014

Dies ist der selbe Eintrag wie bei W: nur das hier anstelle der PID die SID angegeben wird. Unter Neutrino Könt ihr euch die SID eines Kanals anzeigen lassen, indem ihr rot/blau drückt. In der Technischen Info könnt ihr dann die SID finden. Leider fehlen mir hier genauere Infos ob W: und S: angegeben werden müssen oder ob nur einer reicht. Von daher tragt mal eure Karte unter beidem ein. Schaden kann es nicht

cwshare.cfg --- dbox1 --- Internet und Lokales Netzt

```
#
# password
M: { mydbox2.homeip.net { AA242456 } }
```

Das ist eure eigene Adresse gefolgt von eurem Passwort

```
# Internet Friends port range password cod
D: { friend1.homeip.net { 8010 8010 { B142AB11 { 5 5 } } } }
D: { friend2.homelinux.net { 8010 8010 { 81BFF901 { 5 5 } } } }

# other local boxes
D: { 192.168.0.51 { 8020 8020 { AB333441 { 5 5 } } } }
D: { 192.168.0.52 { 8020 8020 { BA334B24 { 5 5 } } } }
```

Mittels dem D: Könt ihr eure Freunde mit denen ihr Sharen wollt eintragen. Als erstes kommt die IP oder Domain gefolgt von dem RX Port danach der TX Port worüber die Kommunikation laufen soll. Hiernach folgt das Passwort und schlußendlich noch die zwei codes für :
 code1 - diese IP darf meine Kartendaten an x Freundeskreise weitergeben.
 code2 - diese IP darf meine Fremde Kartendaten an x Freundeskreise weitergeben.

ignore.list

```
#  
# G: Card records will be ignored in the share.info and will reduce  
# the total cards nummber. (maximum 128 G: records are allowed)  
#  
#  
# -----  
# !!! BLOCK VALID CARD RECORDS ONLY ON BOXES NOT USED FOR FORWARDING !!!  
# To reduce more cards you can spezify here all Providers that you do  
# not use, like conax if you don't have Thor, or providers from all other  
# sattelites if you only have Astra&Horbird. If you still want reduce more  
# you can for example specify 17220000 to ignore all PW Cable cards.  
#  
# But keep in mind that this will totaly discard these card records, and  
# you will also not be able to forward them.  
# -----
```

Ignore Viaccess1 (suppoerted in emu) and not used Viaccess2 Providers

```
G: { 05007800 }  
G: { 05007C00 }  
G: { 05019420 }  
G: { 05019430 }  
G: { 05019440 }  
G: { 05020A40 }
```

Ignore seca1 and seca2 Providers that are suppoerted in emu

```
G: { 01000000 }  
G: { 01000012 }  
G: { 0100001C }  
G: { 01000025 }  
G: { 0100002A }  
G: { 01000030 }  
G: { 01000035 }  
G: { 01000036 }  
G: { 01000066 }  
G: { 01000067 }  
G: { 01000068 }  
G: { 01000071 }  
G: { 01000072 }  
G: { 01000073 }  
G: { 01000082 }  
G: { 01000083 }  
G: { 010001D5 }  
G: { 01000300 }
```

ignore unused cryptoworks provider

```
G: { 0D028800 }  
G: { 0D02A000 }  
G: { 0D02A400 }  
G: { 0D02A800 }  
G: { 0D06C400 }
```

Ignore Nagra1 (suppoerted in emu)

```
G: { 18004001 }  
G: { 18004801 }  
G: { 18007001 }  
G: { 18007101 }
```

Hier können Caid's eingetragen werden die beim Sharen ignoriert werden sollen, z.B. welche die über Emu laufen, Um somit das Netz zu entlasten. Überlegt euch gut welche Ihr hier eintragt, da diese Caid's dann von eurer gbox aus dem Netz dann komplett ignoriert werden und ihr die Caid dann auch nicht mehr weiter leitet.

Die Key Dateien

Als nächstes möchte ich mal auf die Keydateien eingehen.

Irdeto

Der Plainmasterkey wird folgendermaßen eingetragen.

M: { xx xx xx { aa aa aa aa aa aa aa }}

xx xx xx = Provid

aa aa aa aa aa aa aa aa = Plainmasterkey

Der Plainkey wird folgendermaßen eingetragen

I: { zz { aa aa aa aa aa aa aa }}

zz = Key Nr.

aa aa aa aa aa aa aa aa = Plainkey

Seca

Der Masterkey wird folgendermaßen eingetragen. Für Seca2 werden noch die Dateien s2issuer.b und s2provid.b für AU benötigt.

M: { xx xx yy yy yy yy zz { aa aa aa aa aa aa aa }}

xx xx = Provider

yy yy yy yy = PPUA unique address

yy yy yy 00 = PPUA shared address

zz = Masterkey Nr.

aa aa aa aa aa aa aa aa = Masterkey

Der Plainkey wird folgendermaßen eingetragen

S: { xx xx zz { aa aa aa aa aa aa aa }}

xx xx = Provider

zz = Key Nr.

aa aa aa aa aa aa aa aa = Plainkey

Conax

Der Masterkey wird folgendermaßen eingetragen

M: { 00 00 00 yy yy yy yy { zz { aa aa aa..... aa aa aa { bb bb bb.....bb bb bb } } } }

yy yy yy yy = UA oder SA

zz = Masterkey Nr

aa aa.....aa aa = Exp 64 bytes RSA key

bb bb.....bb bb = Mod 64 bytes RSA key

Der Plainkey wird folgendermaßen eingetragen

C: { zz { aa aa aa..... aa aa aa { bb bb bb.....bb bb bb } } }

zz = Key Nr.

aa aa.....aa aa = Exp 64 bytes RSA key

bb bb.....bb bb = Mod 64 bytes RSA key

Via

Der Masterkey wird folgendermaßen eingetragen

M: { xx xx xx yy yy yy yy zz { aa aa aa aa aa aa aa } }

xx xx xx = Provider
yy yy yy yy yy = PPUA unique address
00 yy yy yy 00 =PPUA Shared address
zz = Masterkey Nr.
aa aa aa aa aa aa aa = Masterkey

Der Plainkey wird folgendermaßen eingetragen

V: { xx xx xx zz { aa aa aa aa aa aa aa } }

xx xx xx = Provider
zz =Key Nr.
aa aa aa aa aa aa aa = Plainkey

Der 16Byte aes-Key wird folgendermaßen eingetragen

T: { xx xx xx { cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc } }

xx xx xx = Provider
cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc = 16Byte aes-Key

Nagra

Da sich original Nagrakarten mit dem Reciver verheiraten, unterstützt die gbox auch die Eingabe von IRD und Boxkey. Somit ist es möglich originale Nagra Karten mittels der gbox im Multicam zu betreiben

Der IRD und Boxkey wird vollgundermaßen eingetragen

B: { cc cc cc cc { bb bb bb bb bb bb bb } }

cc cc cc cc = IRD
bb bb bb bb bb bb bb = Boxkey

Der Plainkey für Nagra wird folgendermaßen eingetragen

N: { xx xx zz { aa aa aa aa aa aa aa } }

xx xx = SysID
zz = Key Nr.
aa aa aa aa aa aa aa = Plainkey

Nagra.txt

Hier kommen die Nagra RSA Keys rein. Die Dokumentier ich mal nicht sondern gebe nur mal ein paar Beispiele.

[0001 - Dish Networks]
PK2=10E6782BE27A02E190D9CDDA1221C4
VK=56BF16B8448504CF
N2=331EF8FECE.....291BE486A3C3

[2C01 - SABC]
N1=194A6BE322.....646B462F2AD3
E1=C3C0DC47A.....42F22ECAC68C
VK=176ED451C57F7B96
N2=DFAB71BE.....66DF57B522C5

In der entwicklung dieser Software haben viele Personen teilgenommen. Manche in guter Absicht, manche unwissentlich, mit Ihrer Arbeit und Forschung. Das jemand hier aufgeführt ist, bedeutet nicht das er irgendetwas mit der gbox direkt zu tun hat.

Freundliche Grüße und ein großes danke an:

as, aia, Campag5242, Chianti_le, Dagobert, denny, dvbtux, Gizm0, hpsp, Luki, LuckyLooser, nervous, Nirvana, NoClue, Nullahnung, Secuworld, SoldierX, StillerLeser, TBFT, TaGana, Telefonman, TheBorg, strsh, Sky, tmbinc, trilu, tufler17, Ulster94, zapit_emu.

ohne die, diese Software nicht entstanden wäre.

Sorry falls ich jemanden vergessen habe.