

How to write a **SECA** CAM
by JF
Version 1.03 Aug 2004



Table of contents

1) Starting (Reading the smartcard).....	3
1.1) ATR (Answer To Reset).....	3
1.2) Providers in the smartcard.....	3
1.3) Smart card unique address (Serial Number).....	3
1.4) Providers information.....	4
1.4.1) Each provider information.....	4
1.4.1.1) ID, Name, Shared address and expire date.....	4
1.4.1.2) Package BitMap (PBM).....	4
2) Getting data from provider.....	5
2.1) Reading the PAT.....	5
2.2) Reading the CAT.....	5
2.3) Reading PMT.....	6
3) Conversion between broadcasted data and SECA instructions.....	7
3.1) SECA instruction format.....	7
3.1.1) SECA card status codes.....	8
3.2) From ECM to SECA instruction.....	9
3.3) EMM "Serial Number" to SECA instruction.....	10
3.4) EMM shared to SECA instruction.....	11
4) How to select the right ECM PID.....	11
4.1) Brute force mode.....	11
4.2) Elegant mode.....	12
4.2.1) Checking the ProviderID.....	12
4.2.2) Checking the subscription date.....	12
4.2.3) Checking the Provider BitMap (PBM).....	12
4.2.4) Checking the event code.....	13
4.2.5) Allows a purchased event to be viewed.....	13
4.2.6) Changing the card's PIN.....	13
Appendix A: Glossary.....	14

1) Starting (Reading the smartcard)

1.1) ATR (Answer To Reset)

Before reading anything from the service provider you must read the smart card to know which providers you can decrypt using that card and which packages you are subscribed. All information gathered from the card should be stored in memory to be used later except the ATR.

How to get the ATR from the card is hardware dependent. Once you send the reset to the card it will answer 16 bytes in direct convention starting by 0x3F. The meaning of this bytes are not very important to use the card but will allow you to properly configure the hardware to read the card but usually you do not need them.

ATR example: 3B F7 11 00 01 40 96 xx xx xx 0E 6C B6 D6 90 00

To get explanation of bytes meaning refer to ISO7816 and/or SECA-FAQ.

1.2) Providers in the smartcard

In SECA cards there is at least one provider called SECA and a maximum of 16 and they are accessed using their index in the card starting by 0 (zero). This means that SECA provider has always a 0 index. SECA provider is a special provider not used for decrypt and it is used only to create other providers in the card, remove them and other management functions, usually you will not get any information from the service provider for this card provider it is only used in manufacturing.

To gather information about the card and its providers you can do it using this INS:

C1 16 00 00 06

and the card will answer:

16 ?? ?? xx xx yy yy 90 00

xx xx: The amount of providers in the card in bit format. Each "1" bit is a provider in the card.
Examples:

00 03 = 0000000000000011 = 2 Providers
00 07 = 0000000000000111 = 3 providers
7F FF = 0111111111111111 = 15 Providers

yy yy: PIN status and m080 registers (not used in this document).

1.3) Smart card unique address (Serial Number)

Each SECA smartcard has a serial number unique in all the SECA system to uniquely identify it. Using the INS:

C1 0E 00 00 08

the card will answer:

0E ?? ?? xx xx xx xx xx xx 90 00

xx xx xx xx xx xx: 6 bytes unique address. Display in decimal and the number should appear in the backside of the card.

1.4) Providers information

Each provider in the card have special info like subscription end and subscribed packages. To gather information from them you must enter in a gather information loop for each provider using their index in the card.

1.4.1) Each provider information

1.4.1.1) ID, Name, Shared address and expire date

Using the INS:

C1 12 xx 00 19

xx = Provider Index.

Card will answer something like:

12 pp pp nn nn nn nn nn nn nn nn nn nn nn nn nn yy yy yy zz dd dd rr 90 00

pp pp: Provider identification. 0019, 002E, 0003, 0064, 0072, ... used to reference broadcasted data with provider index in the card.

nn .. nn: Provider name in ASCII format.

yy yy yy: Shared address. This address is used by 256 users.

zz: Custom Byte. This byte identify the user in the shared address group.

yy yy yy zz = PPUA (Shared address + Custom Byte), unique for each user in a provider.

dd dd: Expire date in SECA format (From left to right 7 bits Year + 1990, 4 bits month, 5 bits day).

1.4.1.2) Package BitMap (PBM)

Each provider in the card have an associated Package BitMap (**PBM**) which tell to the decoder which packages the user can see. This information is checked in every decrypt instruction by the card, this means that even when you fool the decoder about the subscribed package the card will refuse to decode a channel that do not belong to a subscribed package. The PBM is an array of 8 bytes and they are checked from right to left. The maximum value of each byte is 0x3F, the meaning of the high bit is unknown to me, but it seems to be that if this bit is set full byte should not be checked, always denied access.

To read the provider PBM you need 2 instructions, the first one request PBM and the second one reads it from the card:

C1 34 xx 00 03 00 00 00

xx = Provider index.

The card will answer:

34 90 00

Now read the PBM from the card:

C1 32 00 00 0A

The card will answer:

32 zz yy yy yy yy yy yy yy yy zz 90 00

zz = Record indicator (04 = a record follows / FF = No more records).

yy yy yy yy yy yy yy yy = 8 bytes Package BitMap.

2) Getting data from provider

To start writing a SECA [CAM](#) you need to know which data do you must gather from the provider transmission. In this document I will start knowing only the SID (Service Identification) of the desired channel and the remaining data will be gathered from the information sent by the provider. All the references here are reverse engineered and could be wrong; the testings has been performed over satellite transmissions in April 2003 on systems called SECA1 and SECA2.

As you know the SID of desired service you need to know the PIDs for audio, video and management data. This data are stored in the PMT and in the CAT tables. CAT table is broadcasted over PID 0x010, but PMT table is broadcasted in a provider designed PID, to locate this PID you must read the PAT table broadcasted over the PID 0x001, look for the desired SID and get the PMT PID.

2.1) Reading the PAT

PAT table structure is documented in ISO/IEC 13818-1 and will not be detailed here. It is being broadcasted over the reserved PID 0x000.

Browse the PAT table looking for the desired service number and write down the PMT PID.

2.2) Reading the CAT

CAT table structure is documented in ISO/IEC 13818-1 and will not be detailed here. It is being broadcasted over the reserved PID 0x010.

Browsing the CAT you will get the Conditional Access Descriptor 0x09, the first 16 bits matches the CAM system. SECA CAM are using 0x01?? to identify. Most **CA** systems uses only one EMM PID for each channel, but SECA systems usually use at least 2 PIDs, one for "card by card" messages and another one for shared messages. "Card by card" messages are usually used to activate or deactivate only one card, while shared

messages usually are used to update the operational keys. This means that you should gather more than one PID. The "card by card" EMM PID or also called the "Serial PID" is stored in the usual field reserved in the descriptor description in standard documentation, but shared PID(s) are provided in a private extension in the same packet. This is a pseudo-code description on how to get this extra PIDs:

Descriptor:=Bits.GetBits(8)	This value must be the descriptor 0x09
DescriptorLen:=Bits.GetBits(8)	Descriptor block length in bytes
CA_System:=Bits.GetBits(16)	Conditional Access System
I:=0	Reset and start a loop
While i<DescriptorLen do	
Reserved:=Bits.GetBits(3)	Reserved data (not used)
EMMPID:=Bits.GetBits(13)	Main EMM PID("Card by card").
Inc(i,2)	
If CA_System and 0xFF00 = 0x100 Then	SECA CA_System
ExtraEMM:=Bits.GetBits(8)	Amount of extra EMM PIDs
Inc(i,1)	
While ExtraEMM>0 Do	Start a new loop
Reserved:=Bits.GetBits(3)	Reserved data (not used)
EMMShared:=Bits.GetBits(13)	EMM Shared (PPUA)
ProvID:=Bits.GetBits(16)	SECA Provider ID for that EMM Shared. If provider is not present in the card you can discard this EMM Shared PID.
Inc(i,4)	
Dec(ExtraEMM)	
Loop	
End If	
Loop	

Now you are collected data enough to open EMM PID(s) for the selected service.

2.3) Reading PMT

Browsing the PMT you will get PIDs for audio, video and others reading it as explained in ISO/IEC 13818-1, but in the ECM part for SECA ECMs you will find a private section holding extra ECM PIDs for current service. This ECMs are selected based in the providers of your card and in the rights that you must have in each provider to use it. This is an example of a PMT about the SECA private section:

```
017 0-Stream Type (8): 0x02 (2) ITU-T Rec. H.262 | ISO/IEC 13818-2 Video
018 0-Reserved (3): 0x7 (7)
018 3-Elementary_PID (13): 0x00A9 (169)
020 0-Reserved (4): 0xF (15)
020 4-ES_Info_Length (12): 0x013 (19)
022 0-->Descriptor TAG (8): 0x09 (9)
023 0-->Descriptor Length (8): 0x11 (17)
024 0-->Conditional Access System ID (16): 0x0100 (256) Canal Plus (SECA)
026 0-->Reserved (3): 0x7 (7)
026 3-->CA_PID "ECM" (13): 0x05E2 (1506)
028 0-->Private Data 00 64 FF 00 00 00 01 00 00 00 02 1A 83
```

The ECM PID 0x05E2 could be considered a "default" PID as it will be used by all non 100% SECA aware CAMs. In SECA2 usually this PID broadcast data intended for preview only (maximun 2 minutes), but in the private

section you will find other PIDs to be used in that channel. You can gather information from private section in this way, starting from the Conditional Access descriptor:

Descriptor:=Bits.GetBits(8)	It must be 0x09
DescriptorLen:=Bits.GetBits(8)	Descriptor block length in bytes
I:=0	Reset counter and start loop
While i<DescriptorLen Do	
CA_System:=Bits.GetBits(16)	Conditional Access System
If CA_System and 0xFF00 = 0x0100 Then	SECA CA
Reserved:=Bits.GetBits(3)	Reserved.
ECM:=Bits.GetBits(13)	The ECM PID
ProviderID:=Bits.GetBits(16)	The Provider ID for that PID, if that provider ID is not in the card you can discard this ECM PID and all data in this loop.
SECADescriptor:=Bits.GetBits(8)	Special SECA descriptor
If SECADescriptor=0xFF Then	
PBM:=Bits.GetBits(64)	64 bits for Package BitMap (PBM)
Date:=Bits.GetBits(16)	Today date in SECA format
End If	
If SECADescriptor=0x7F Then	
EventID:=Bits.GetBits(16)	EventID of current transmission
PBM:=Bits.GetBits(64)	64 bits for Package BitMap (PBM)
End If	
End If	
Loop	

With this information you can open all needed filters for EMM processing. How to convert PID data to the SECA instruction format will be discussed below.

3) Conversion between broadcasted data and SECA instructions

3.1) SECA instruction format

SECA instructions follows the ISO7816 standard and will be described in this document as:

CLA INS P1 P2 LEN PAY

- **CLA**: Class. For SECA only class 0xC1 is valid.
- **INS**: Instruction code. Referr to SECA-FAQ for details about it's possible values.
- **P1**: Provider index in the card and other information.
- **P2**: Key to be used and other information.
- **LEN**: Instruction data payload.
- **PAY**: Optional data payload. INS dependent.

INS carry information about the INS direction (CARD -> CAM or CAM -> Card). If **INS and 0x02 = 0x00** then information will flow from the **CAM** to the **card**, **LEN** must indicate the amount of data and **PAY** will carry the information. Example:

C1 34 00 00 03 00 00 00

If **INS** and **0x02 = 0x02** then information will flow from the **card** to the **CAM**, **LEN** must indicate the requested data length from the card and **PAY** must be equal to nothing. Example:

C1 32 00 00 0A

In both modes the card will always answer first the INS echo, 1 byte length, then optional payload if INS and 0x02 = 0x02, and finally the status bytes, 2 bytes length.

3.1.1) SECA card status codes

As noted above the card always report how it finish the requested operation if the instruction is well formed. The card will not answer if PAY is shor that specified in LEN for CAM -> Card instructions. In this case a card reset is the suggested operation.

Not all status codes are know but here you can find a list of currently described in SECA-FAQ:

SECA card status codes
6700: Invalid input length
6B00: Wrong reference parameter/byte
6D00: Unsupported or invalid instruction
6E00: Instruction class not supported
6F00: No precise diagnostic
6281: Submitted data probably incorrect
6282: EOF before finishing read process
6284: Selected file not valid
6501: Hardware error
6800: Previous instruction not supported by the card
6A00: P1 and/or P2 value(s) invalid
6A80: Data parameters invalid
6A82: File not found
6A83: Register not found
6A84: No free space for file or register
9000: Success
9002: Invalid signature or absent
9004: Unsupported provider
9005: Nano for SECA only
9006: No memory for preview record
9007: Not allowed, provider locked

SECA card status codes
9008: Nano 0x01 not allowed
9009: PPUA CW not in F0 bitmap (The card is not subscribed)
9010: Invalid PIN
9011: Bit test unset. Unsupported instruction
9013: Invalid key. Only MKs
9014: Invalid previous instruction
9015: Nano not allowed or invalid input data
9016: Not found or not allowed
9017: Not allowed
9018: Provider already exists
9019: PPUA has been modified
901A: No valid JETONS purchase
901B: No JETONS or credit
901C: JETONS purchase OK using credit
901D: Primary key not available
901E: Error
901F: Secondary key not available
9021: Key is not 0x0F - Denied
9022: The card is in invalid mode
9024: Checksum fail
9026: No more preview
9027: Preview mode
9301: Date card has been expired
9302: No decode
9304: Too low parental rating
9305: Your geographic zone has been banned (Temporal)
9401: P1 value invalid
9402: P2 value invalid
9600: Invalid input, or null event or all nanos processed. No decode
960A: Unsupported key index

3.2) From ECM to SECA instruction

Data arrived from ECM PID is divided in tables. Table header for ECM data can be **0x80** and **0x81**.

When this value changes from one to the other a new ECM must be sent to the card.

ECM data from ECM PID could look like this:

ECM data example																
0000:	80	00	30	00	19	00	00	0C	71	00	00	19	07	D3	00	04
0010:	12	00	27	1A	95	13	02	13	09	D1	26	A6	69	73	C1	C1
0020:	30	81	7F	4A	78	B6	91	64	62	18	82	BD	40	51	DD	74
0030:	BD	F9	C8													

And the data should be mapped in this way:

Length:= (ECM[1] < 8 or ECM[2]) and 0x0FFF (0030)

INS:= 0x3C (ECM SECA instruction)

P1:= Get index from the card for provider **ECM[3] < 8 or ECM[4] (00 19)**

P2:= ECM[7] (0x0C)

LEN:= Length - 0x05 (0x2B)

PAY:= ECM[8..Length+3] (71 00 00 19 ... BD F9 C8)

There is an extension not currently used, so I can no say that it should be mapped in this way or not. The high nibble of P1 carry extra information extracted from ECM[5] or ECM[6]. **P1:= P1 or ECM[5] or P1:= P1 or ECM[6]**. One of this two options should be the valid one, but I do not know which one. I think this extension will never be used.

3.3) EMM "Serial Number" to SECA instruction

Data arrived from "main" EMM PID is directed to the card serial number. This number is unique for all SECA cards. EMM "Serial Number" uses the reserved EMM table ID 0x82 and could look like this:

EMM "Serial Number" data example																
0000:	82	00	62	00	00	00	FC	3C	DA	00	19	00	80	CF	91	23
0010:	5D	AE	18	65	D6	0C	FF	35	36	5E	F0	97	C9	C7	3D	77
0020:	3F	46	13	CB	7F	68	F8	10	53	7E	73	B7	87	0D	3C	F0
0030:	D7	23	35	9E	11	E7	5B	9E	E9	32	43	A8	B9	69	71	2F
0040:	3F	A7	91	AB	9B	93	58	0F	47	0F	66	CA	C2	9F	7C	9D
0050:	61	3F	9F	C2	67	61	F4	B3	E5	25	69	48	D8	19	7A	BE
0060:	AD	F7	66	C1	86											

And the data should be mapped in this way:

Length:= (EMM[1] < 8 or EMM[2]) and 0x0FFF (0062)

INS:= 0x40 (EMM SECA instruction)

P1:= Get index from the card for provider **EMM[9] < 8 or EMM[10] (00 19)**

P2:= EMM[12] (0x80)

LEN:= Length - 0x09 (0x59)

PAY:= EMM[13 .. Length+3] (CF 91 23 ... 66 C1 86)

This instruction should only be sent if the serial number matches the card serial number. To test the

serial number get it from **EMM[3 ... 8]**, as SECA serial numbers are 6 bytes in length, and check it against the card serial number.

There is an extension not currently used, so I can no say that it should be mapped in this way or not. The high nibble of P1 carry extra information extracted from EMM[13]. **P1:= P1 or EMM[11]**.

3.4) EMM shared to SECA instruction

Data arrived from extra EMM PIDs are directed to the shared address of each provider in the card. This number is unique for 256 possible users in each provider. EMM "Shared" uses the reserved EMM table ID 0x84 and could look like this:

EMM "Shared" data example	
0000:	84 00 55 00 19 00 12 E8 00 81 E5 33 6E CD 8E 7B
0010:	3A D8 8F 74 7F C3 C8 A1 2D F6 C8 49 C4 A2 C6 3B
0020:	AB 9D B9 0E E5 30 0C 13 8E 25 4F 02 14 46 9A AA
0030:	9B AA 77 6E 83 BA FB 11 32 A8 0B CC E6 16 29 6C
0040:	B4 4D 36 0A 80 45 44 7A 1E 52 FD 74 2D BB E8 2B
0050:	28 92 CF 1B 82 67 3E 48

And the data should be mapped in this way:

Length:= (EMM[1] << 8 or EMM[2]) and 0x0FFF (0055)

INS:= 0x40 (EMM SECA instruction)

P1:= Get index from the card for provider EMM[3] << 8 or EMM[4] (00 19)

P2:= EMM[9] (0x81)

LEN:= Length - 0x07 (0x4E)

PAY:= EMM[10 .. Length+3] (E5 33 6E ... 67 3E 48)

This instruction should only be sent if the shared address matches the card shared address for the provider. To test the shared address get it from **EMM[5 ... 7]**, as SECA shared addresses are 3 bytes in length, and check it against the shared address of the provider.

There is an extension that has been used one time by CSD Spain in April 2003, the high nibble of P1 carry extra information extracted from EMM[8]. **P1:= P1 or EMM[8]**.

4) How to select the right ECM PID

On serveral services there are more than one ECM PID and you must be able to select the right one. To accomplish this job you should check the subscription date, the PBM and the EventID.

4.1) Brute force mode

You can start a loop for all ECM PIDs and check the status code that the card answers. This method is a bit slow but requieres less "intelligence" by the CAM. Pseudo code example:

ECM select pseudo code example
<pre>For ECM = 1 to ECM.Count Open ECM PID if Provider exists in card then Build SECA ECM and send it to the card If Status=0x9000 Then exit End If Close ECM PID Next // If code reaches this point no ECM can successfully decode the service, try preview mode. For ECM = 1 to ECM.Count Open ECM PID if Provider exists in card then Build SECA ECM and send it to the card If Status=0x9027 Then exit End If Close ECM PID Next // If code reaches this point no ECM can successfully decode the //service. No ECM needed.</pre>

After a lot of checks against this technique and the next one (Elegant Mode) I must confess that the brute force one is the better one as the (Elegant Mode) seems to work different in SECA1 and SECA2 and even between platforms.

4.2) Elegant mode

For this mode you should check the ECM data against data collected in the boot card procedure (when the card is inserted in the smartcard slot). This mode has not been reversed engineered 100% so it could have errors.

4.2.1) Checking the ProviderID

Select only the ECMs which Provider ID is one of the available in the card.

4.2.2) Checking the subscription date

Select only the ECMs which date is lower than the expire date of each subscription.

4.2.3) Checking the Provider BitMap (PBM)

This is the hardest procedure and the most unclear. Currently I'm using a procedure that seems to work in all services. Check each ECM-PBM byte from right to left against Card-PBM, for the selected provider, using bitwise AND operation only when ECM-PBM is different than FF (in this case reject the byte and take it as a false comparison). If one of the bytes results a value different than zero you can use that ECM.

If ECM-PBM[Counter] and Card-PBM[Counter] != 0 Then UseThisECM

4.2.4) Checking the event code

To check the event code you must send an instruction to the card asking if the EventID is present in the card. If present you can use this ECM.

In fact there is no need to check if the event is present, send the instruction and if the card answer 901A then you can buy the event using JetONS or authorize the use of a event view in the purchased event in the card.

4.2.5) Allows a purchased event to be viewed

When the card have a purchased event and you try to watch it, the card will answer 901A, requesting the user to confirm that one "view" will be used right now. Purchased events have a "views" counter that will be decremented everytime you watch that event. SetTopBoxes usually shows an OSD message telling something like "Press OK to view", if you press OK an instruction will be sent to the card allowing the event view.

This instruction looks like this:

C1 30 00 02 09 00 00 00 00 00 00 00 00 00 FF

Highlighted bytes are the PIN stored in the card (00 00 means no PIN). If the PIN matches the PIN stored in the card the answer will be 9000, if it does not match it will answer 9010, at this point you have two possibilities, change the card's PIN or transmit the same INS again with a different PIN. The PIN values is transmitted in "decimal" format, well BCD in fact. PIN 1234 will be encoded as 12 34, so PINs like FF FF are not allowed (in fact never tested by myself).

Adult contents will ask you for the PIN always when it is different than 00 00.

4.2.6) Changing the card's PIN

You can change the PIN in your card using the instruction:

C1 30 00 00 10 00 00 00 00 00 00 00 00 00 **XX XX** 00 00 00 00 00 00 00 00 **YY YY**

Where **XX XX** is the PIN in the card and **YY YY** is the new PIN. If the value of **XX XX** does not match with the value in the card the answer will be 90 10.

Note: Most SECA2 decoders seems to have the PIN in the firmware so the card functions are not used at all, but this ones are present in the card and can be used anyway.

Appendix A: Glossary

CAM: Conditional Access Module. Hardware PCMCIA that receive data from the SeTopBox and send it to a subscription card to see encrypted channels. The CAM must be attached to a CI.

CI: Common Interface. Usually connect a CAM with a SetTopBox.

SetTopBox: Standard DVB receiver, satellite, cable or terrestrial.

SID: Service Identification. A 16 bits number that identify a service (usually TV, Radio or Data) in a data stream.

PAT: Programm Association table.

CA: Conditional Access.

EMM: Entitled Management Message.

ECM: Entitled Control Message.

EventID: Event code that identifies a purchasable event, like a movie, match, show,...

PBM: Provider BitMap. Describes which packages the user is paying for.

SECA: Societe Europeene de Controle d'Acces. European Society of Access Control. Also known as Mediaguard. [More...](#)

PCMCIA: Personal Computer Memory Card International Association. [More...](#)

CAT: Conditional Access Table.