Attacks on Pay-TV Access Control Systems

Markus G. Kuhn

Computer Laboratory

UNIVERSITY OF CAMBRIDGE

Generations of Pay-TV Access Control Systems

Analog Systems

- remove sync information, try to confuse gain-control in receiver, etc.
- cryptography is not essential part of decoding process
- still dominant type for most cable-TV premium channels

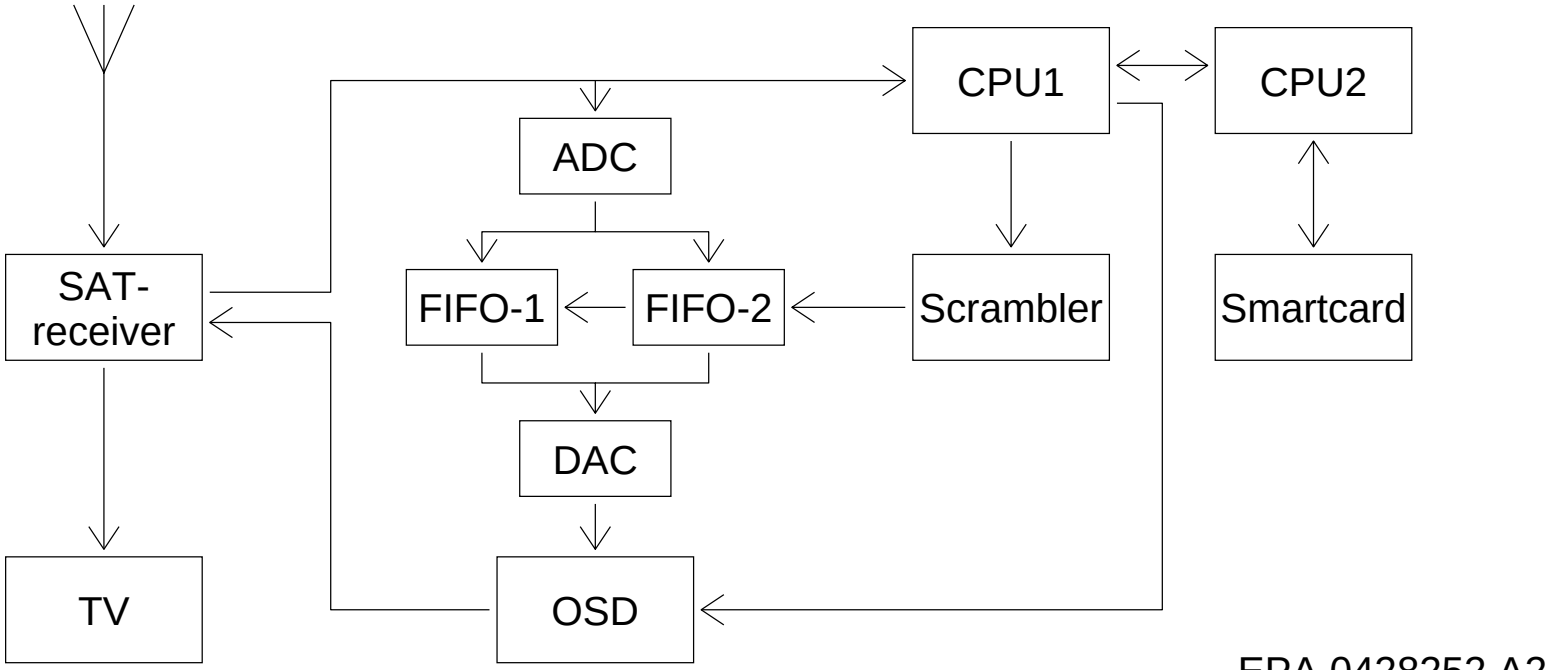
Hybrid Systems

- broadcasted signal conforms to analog TV standard (PAL, D2MAC, NTSC, SECAM)
- analog signal scrambled with digital framebuffer using a cryptographically transmitted control word
- fully cryptographic subscription management using smartcards
- examples: VideoCrypt, EuroCrypt (EN 50094), Syster Nagravision

Digital Systems

- broadcasted signal is digitally modulated, encrypted, and multiplexed MPEG-2 audio and video data stream
- cryptographic subscription management using smartcards as with hybrid systems
- examples: DVB, DSS/VideoGuard

Example of a Hybrid System: VideoCrypt



EPA 0428252 A2

Features:

- scrambling by active-line rotation, requires only memory for one single image line
- vertical-blank-interval data contains 32-byte messages with blacklist/whitelist data
- smartcard calculates 60-bit MAC as control word from 32-byte messages every 2.5 s
- CPU1 salts control word with frame counter to generate 60-bit PRNG seed per frame
- Scrambler uses 60-bit seed to generate cut-point sequence per frame

An Image Processing Attack on VideoCrypt

unscrambled source signal

broadcasted scrambled signal

result of cross-correlation with cutpoints marked

edge detector avoids horizontal penalty zones around cut points

final b/w descrambling result obtained without knowledge of card secret

The VideoCrypt Smartcard Protocol

Flow control

ISO 7816 T=0 protocol: sent by decoder / smartcard

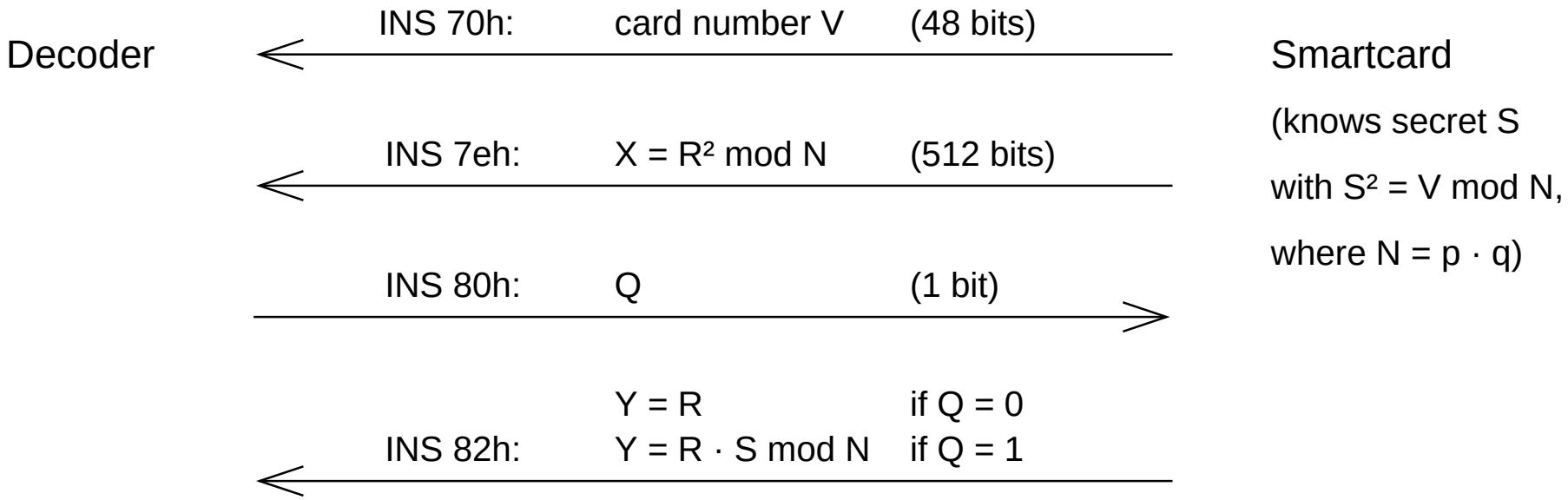
CLA INS P1 P2 P3 INS DATA[1] . . . DATA[P3] SW1 SW2

Instructions

INS	length (P3)	sent by	purpose
70h	6	card	card serial number
72h	16	decoder	message from previous card
74h	32	decoder	message from broadcaster
76h	1	decoder	authorize button pressed
78h	8	card	control word (MAC of 74h)
7ah	25	card	onscreen display message
7ch	16	card	message to next card
7eh	64	card	Fiat-Shamir squared random number
80h	1	decoder	Fiat-Shamir challenge bit
82h	64	card	Fiat-Shamir response

VideoCrypt or How not to use the Fiat-Shamir ZKT

Protocol



Decoder receives Q periodically from broadcaster and forwards it to the smartcard

Decoder is supposed to reject smartcard if the following test fails (first generation did not):

$Y^2 = X \bmod N$ if Q = 0 $Y^2 = X \cdot V \bmod N$ if Q = 1

Attack

Decoder has no memory to verify that X is different each time, so pirate card just observes

V, R, $R^2 \bmod N$, and $R \cdot S \bmod N$ from any card and replays those values each time.

Replay attacks against VideoCrypt

Vulnerabilities

1) all VideoCrypt smartcards working on the same channel reply identically
2) the scrambled VideoCrypt signal can be replayed with a normal home VCR

Real-time card sharing (old proposal, not implemented)

One owner of a genuine card provides the control words in real-time via wire or radio to owners of decoders without a card (60 bits every 2.5 s).

Offline Internet card sharing (common practice!)

One owner of a genuine card records control words and synchronization information for a specific show (say Star Trek on Sunday, 18:00) in a VideoCrypt Logfile (VCL) and publishes this on her Web page.
Decoder owners without card record the scrambled programme, then download VCL file and put decoder between VCR and TV. A PC then emulates card and replays control words from VCL file. VideoCrypt Broadcast Logfiles (VBL) allow a posteriori VCL file generation.

Potential risk

Covert channel might identify card owner in public VCL files, therefore use VCL voter

Secret Hash/MAC Algorithms in VideoCrypt Smartcards

Hash and Signature Check Structure:

```
j = 0;
answ[0..7] := 0;
for i:=0 to 26 do
    round(msg[i]);
b := 0;
for i:=27 to 30 do
    round(b);
    round(b);
    if answ[j] != msg[i] then
        signature wrong
    j := (j + 1) mod 8;      only in P07
    b := msg[i];
in P09 handle nanocommands here
for i:=1 to 64 do
    round(msg[31]);
```

Input: msg[0..31]

Output: answ[0..7]

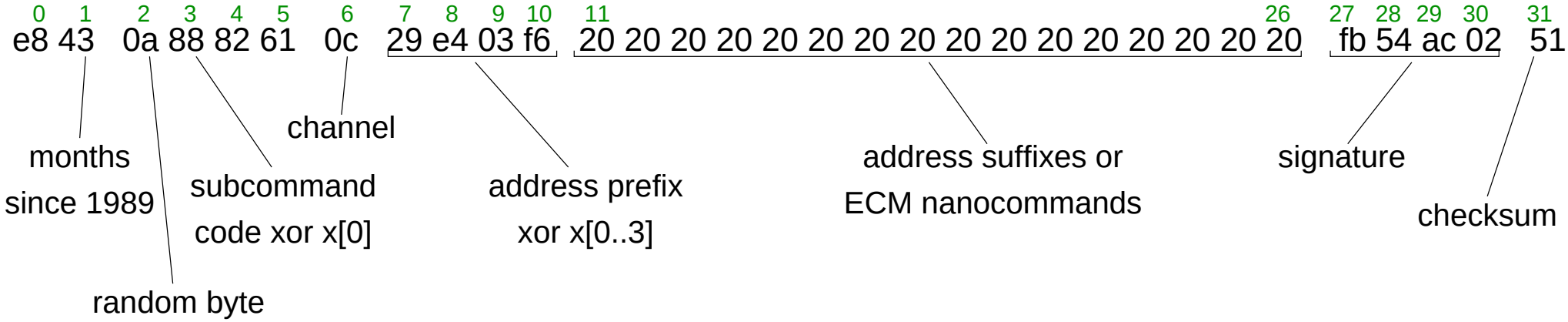
all variables are 8-bit unsigned

Round Function in BSkyB P07:

```
parameter p
answ[j] := answ[j] xor p;
c := sbox[answ[j] / 16] +
    sbox[answ[j] mod 16 + 16];
c := rotate_right(rotate_left(not c, 1) + p, 3);
j := (j + 1) mod 8;
answ[j] := answ[j] xor c;
```

P09 card used completely different round function

BSkyB P09 Structure of 32-byte Message in Instruction 74h



XOR Scrambling:

```
a := msg[1] xor msg[2];
swap_nibbles(a);
b := msg[2];
for i:=0 to 3 do
    b := rotate_left(b, 1);
    b := b + a;
    x[i] := b;
```

Subcommands:

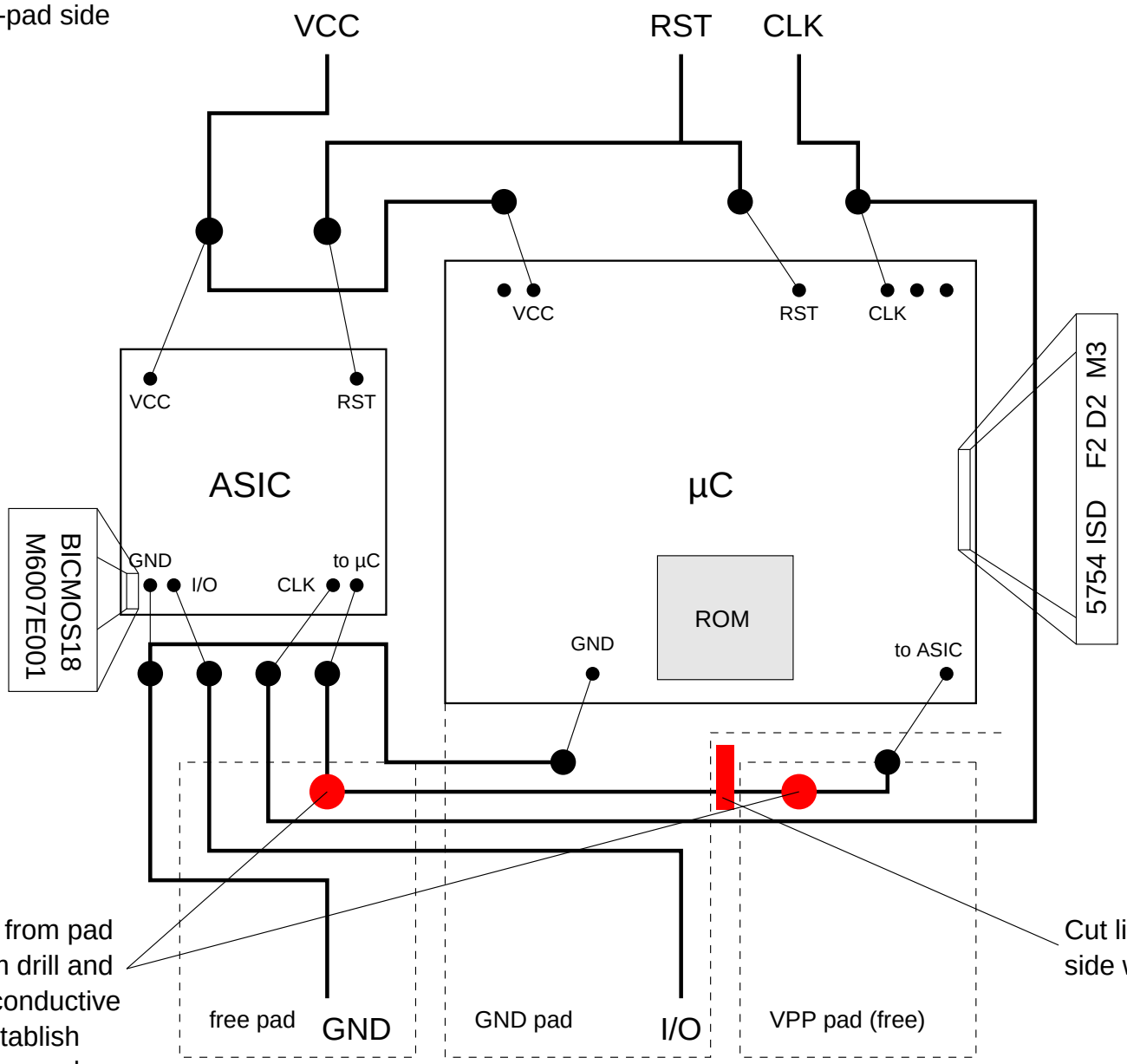
00	deactivate card
01	deactivate Sky Movies
. . .	
20	activate card
21	activate Sky Movies
. . .	
40	PPV management
80	ECM nanocommands
. . .	

Nanocommands:

cause calculated jumps into highly obscure machine code, many add additional rounds, some read or write RAM or EEPROM locations, the nanocommand interpreter is designed to be extremely non-portable and difficult to understand

Conductive Silver Ink Attack on the BSkyB P10 Card

view from non-pad side

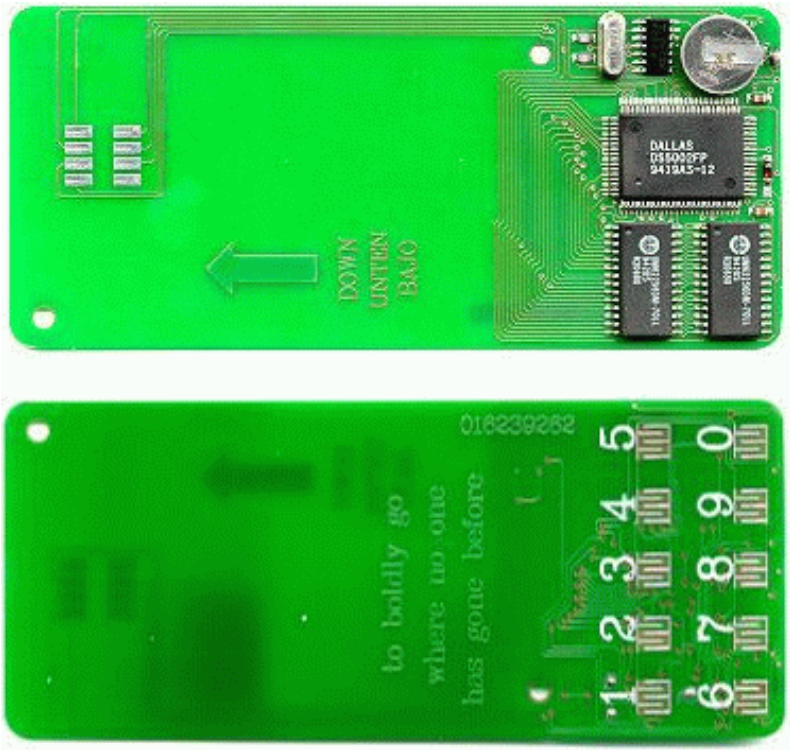


Drill two holes from pad side with 1 mm drill and fill holes with conductive silver ink to establish contact with free pads

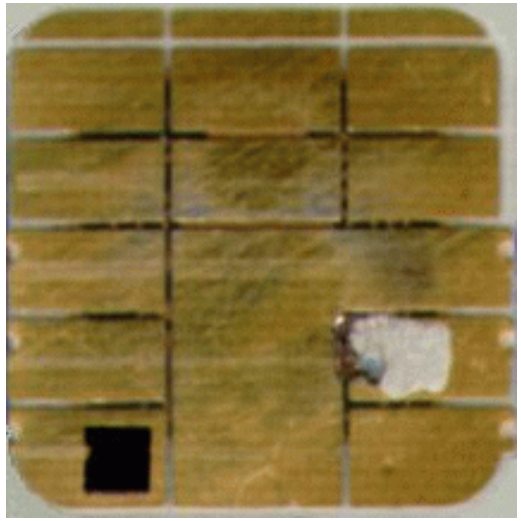
Cut line from pad side with sharp knife

M. Kuhn

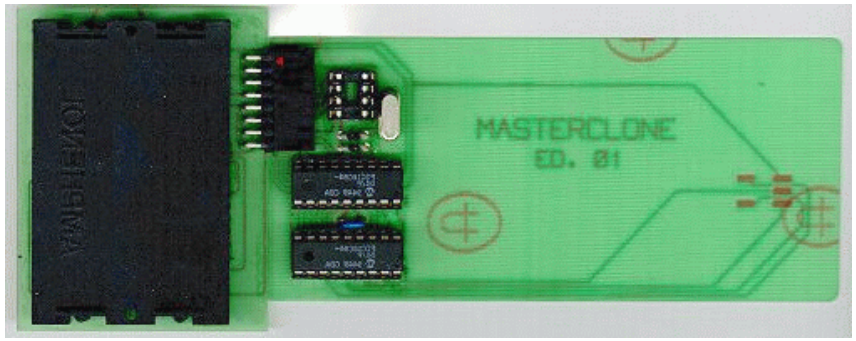
Some Pay-TV Pirate Devices



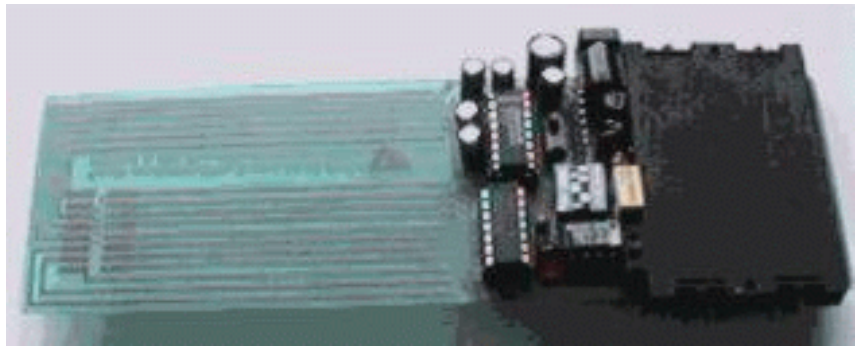
"Battery-powered smartcard", Megasat Bochum



Conductive silver ink attack on BSkyB P10 card (top), with card CPU replaced by external DS5002FP (right)

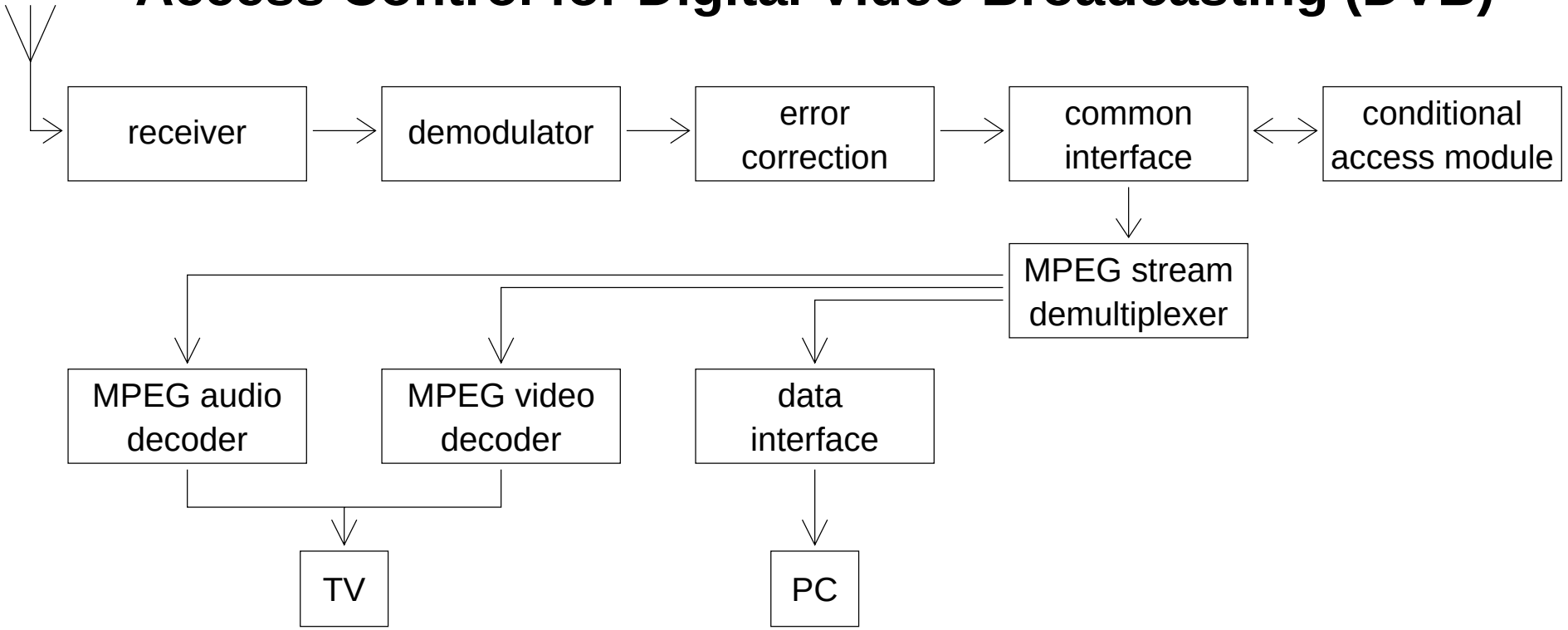


BSkyB P9 deactivation blocker



ISO 7816 to RS-232 adapter (Season7)

Access Control for Digital Video Broadcasting (DVB)



Access control issues:

- Standardization of complete access control system was politically not possible
- Standardization of Common Interface (PCMCIA slot) to allow plug-in access control
- Standardization of Common Scrambling Algorithm will at least allow SimulCrypt, where different access control systems can decrypt the same control words in order to descramble the same programme

Robust Key Management Scheme for Pay-TV Smart Cards

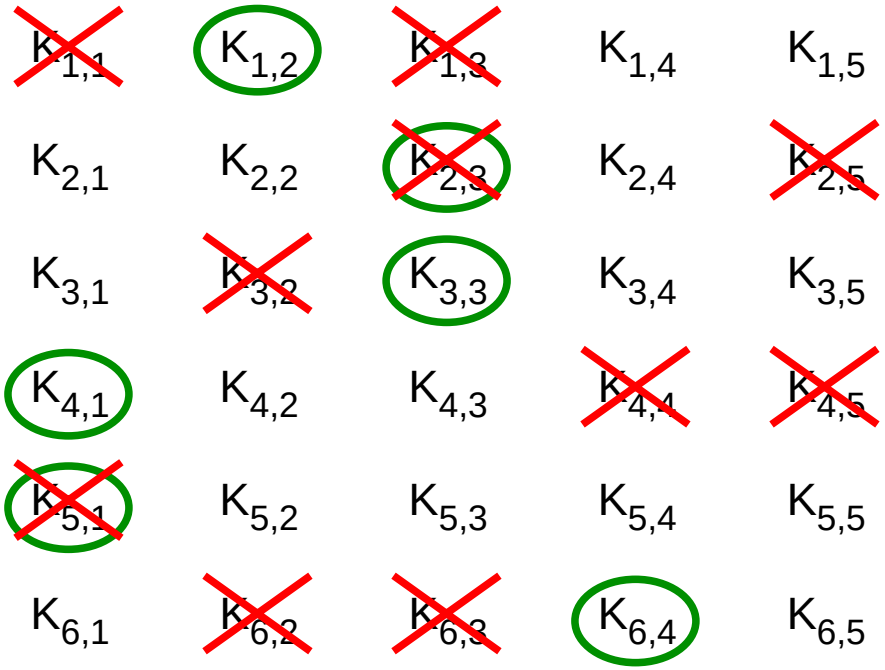
Idea

— Every card contains a subset of L=10 keys out of a pool of K·L=300 keys $K_{i,j}$ which are used for session key uploads

— If pirates open C=20 cards, only $(1-(1-1/K)^{C})^{L}$ = 0.08% of the genuine cards have to be replaced to recover confidentiality of session key updates

Example

L=6, K=5, C=2



Compromised Key

Key in an uncompromised card

— Single rows or all uncompromised keys are used for session key uploads

— Each card knows one key per row

— Cards that know only compromised keys have to be replaced

Lessons Learned from Pay-TV Piracy

- Every security microcontroller and ASIC will be reverse engineered within weeks
  if pirates see a chance to make a million dollars profit from doing it
- Routine recovery from attacks by ECMs, key updates, exchange of security modules, etc.
  must already be planned for in the design phase of a large scale cryptographic application
- Today's EEPROM processor smart card technology is unsuitable for holding global secrets
- Continuous pirate market observation and analysis of pirate devices becomes
  routine activity for any consumer multimedia access control system operator
- Obfuscated programming, customized processors, and other portability surprises
  in security module software are successful for only a few days and should be replaced
  by more flexible key management (Kerckhoffs' principle)
- Analog and hybrid pay-TV systems do not provide signal confidentiality and will
  eventually be broken by real-time image processing attacks