deDECTed.org

Andreas Schuler Erik Tews Ralf-Philipp Weinmann

December 29, 2008

Who we are

People from various locations participate in the project:

Darmstadt	Cryptanalysis, packet capturing, FPGA implementations, reverse engineering
Luxembourg	Cryptanalysis, reverse engineering, packet capturing, writing drivers
Trier	reverse engineering, packet capturing, writing drivers
Munich	reverse engineering, packet capturing, writing drivers
Weimar	Cryptanalysis
Berlin	infrastructure, chip reverse engineering, counceling
Wiesbaden	kismet integration

in the past, there was...

CT1(+) Analog escommunication, differenct frequencies for both directions

CT2 Analog communication, same frequency for both directions, time multiplexing

No encryption at all, no security (?)

DECT usage

DECT is used for:

- Cordless phones
- Wireless ISDN access
- Babyphones
- Emergency calls
- Remotely controllable door openers
- Cordless EC termimals
- Traffic lights

situation in germany:

- ≈ 30.000.000 base stations currently in use

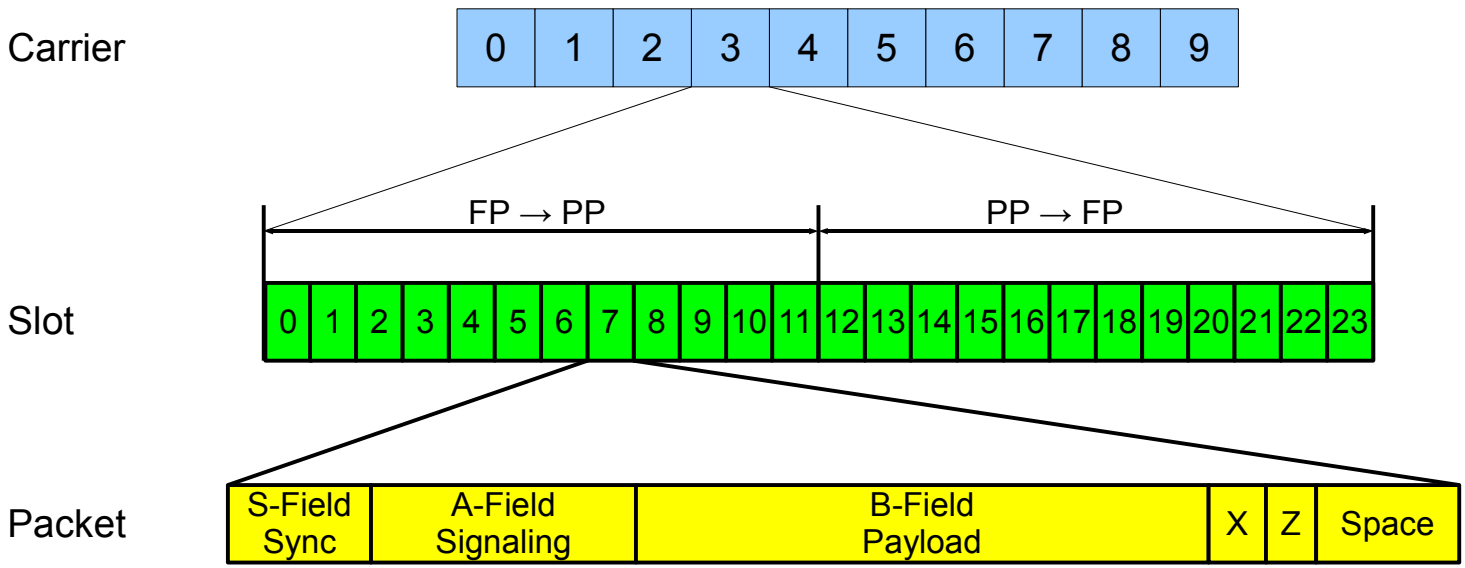
Terms

FP	Fixed Part (base station)
PP	Portable Part (cordless phone/handset)
RFPI	Radio Fixed Part Identity
IPUI	International Portable Users Identity
DSC	DECT Standard Cipher
DSAA	DECT Standard Authentication Algorithm
UAK	User Authentication Key (shared key between handset and base station)

DECT

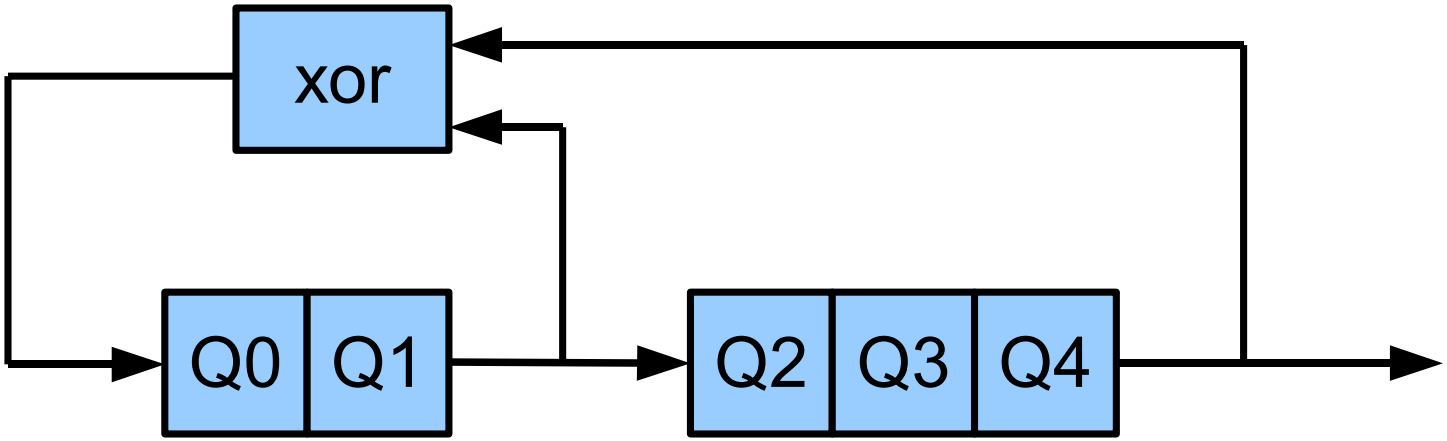
- ETSI EN 300175
- Digital communication
- GFSK modulation
- EU: 10 carriers (1880-1900 MHz) with 250mW
- US: 5 carriers (1920-1930 MHz) with 100mW
- channel spacing 1.728 kHz
- 24 time slots per channel, 12 upstream, 12 downstream

DECT time/frequency multiplexing



Scrambling

- avoids DC-Offset
- B-Field data will be XORed with output from LFSR
- LFSR will be initialized with framenumber
- LFSR is public



Encryption

- A-Field control channel (setup communication,dialing,...) will be XORed with bitstream from DSC
- B-Field data will be XORed with bitstream from DSC

FP/PP behaviour

FP (station)

- Broadcasting network informations (RFPI,...)
- Scanning on all carriers and possible slots for PP activity

PP (phone)

- Don't send in idle mode
- Scanning and making a list of carrier average RSSI
- Syncronizing with base station
- Select best carrier/slot-combination for communication and opening connection
- Initiate Encryption

Sniffing difficulties

- Stations not synchronized
- No packet source/destination field like in ethernet-packets
- We don't know where PP opens connection
- For descrambling the framenumber must be known

USRP DECT Sniffer

- Can capture all packets on a channel
- CPU requirements are high (≈ 2 GHz CPU required)
- Time multiplexing is difficult to handle
- Sending frames is not supported

Total costs for this tool: 1000 EUR

ComOnAir DECT Sniffer

- Can scan for stations or active calls
- Can sync on stations and dump active calls
- CPU requirements low
- Sending frames supported soon

Total costs for this tool: 23 EUR

ComOnAir reversing

- Card with Windows driver as basestation for SIP telephony
- No Linux driver

ComOnAir reversing

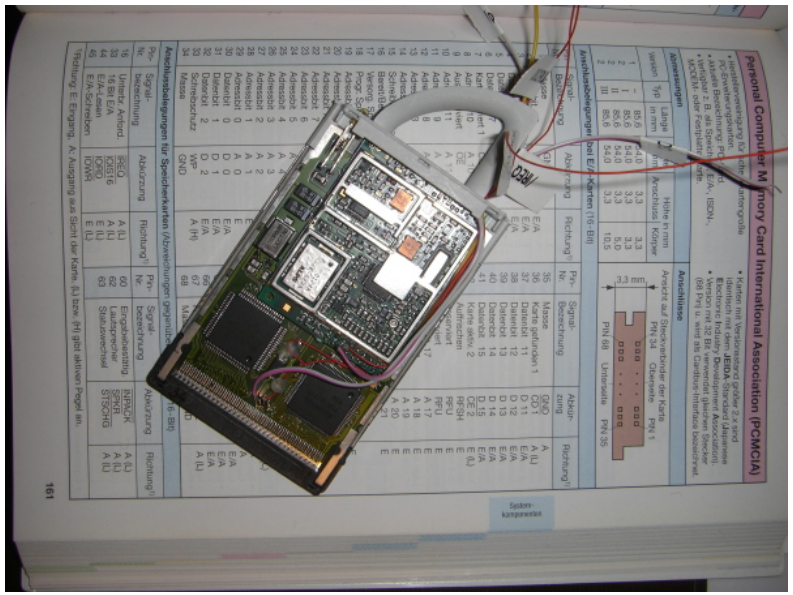
- Reading out PCMCIA tuples
- Writing simple Linux character device driver
- Get access to IO memory

ComOnAir reversing

- Removing case
- Searching datasheets
- Reversing hardware
- Reversing Windows driver, find firmware image
- Try to activate hardware
  - Write firmware with unknown command-set
  - Upload firmware to chip
  - Wait for interrupts

ComOnAir reversing

- ...and try...and try...and try...
- Adding measurement tools to card

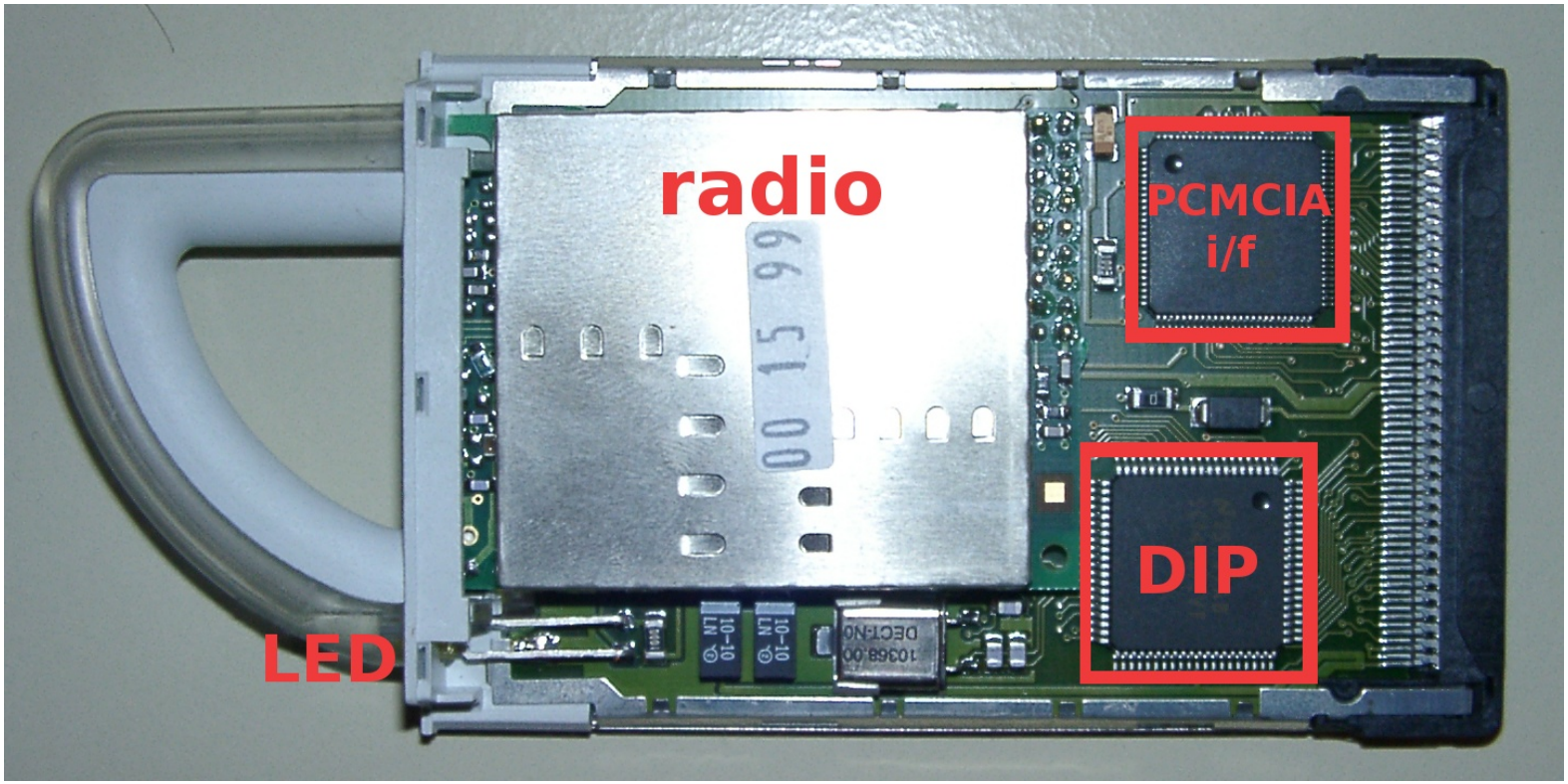


ComOnAir reversing

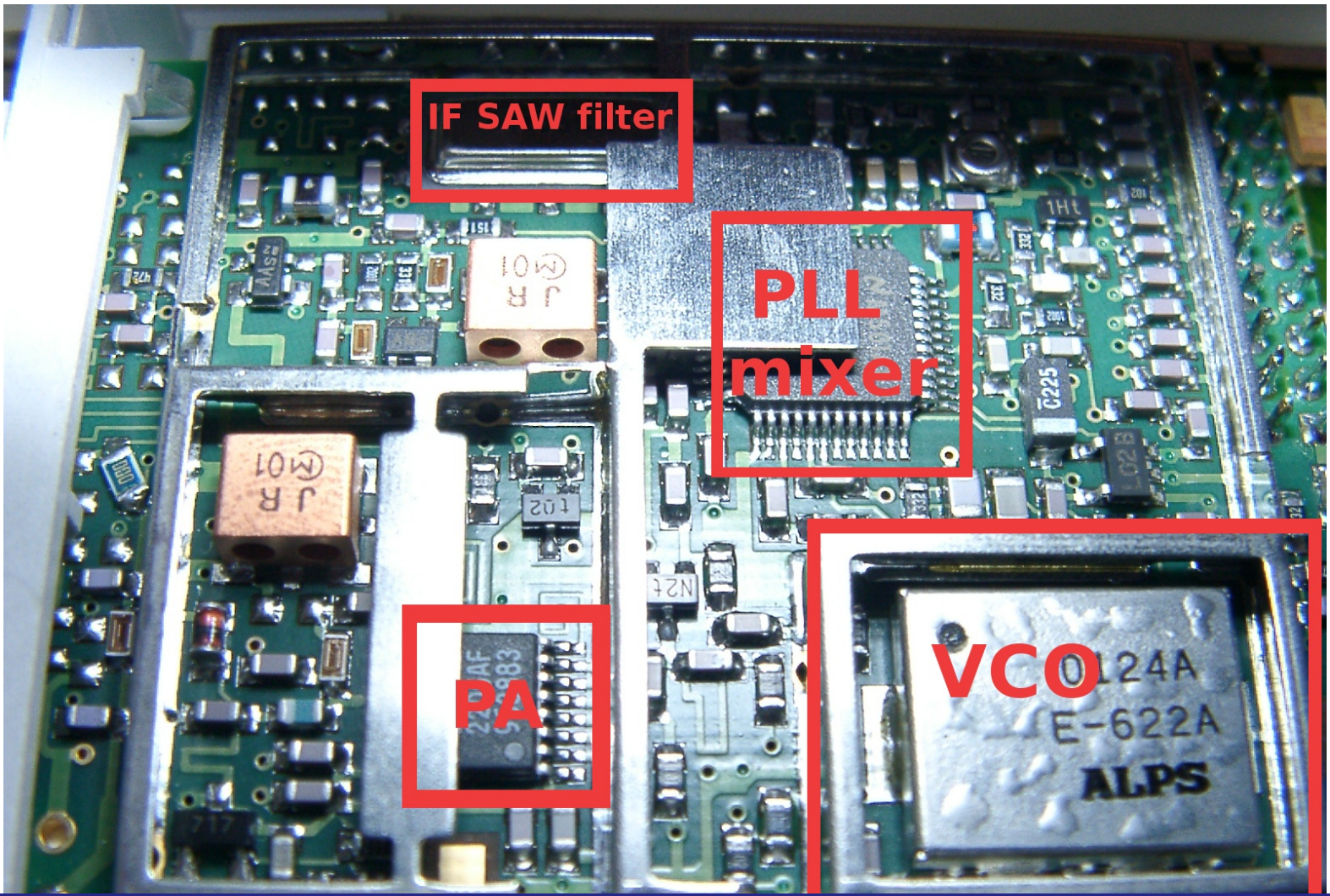
```
commit b2185f943fd642bd46ca4e13f87d3fce374fbe69
Author: xxx xxx <xxx@xxx.xxx>
Date:   Wed Dec 3 23:59:21 2008 +0000

    WE HAVE INTERRUPTS cat /proc/interrupts ! :))
```

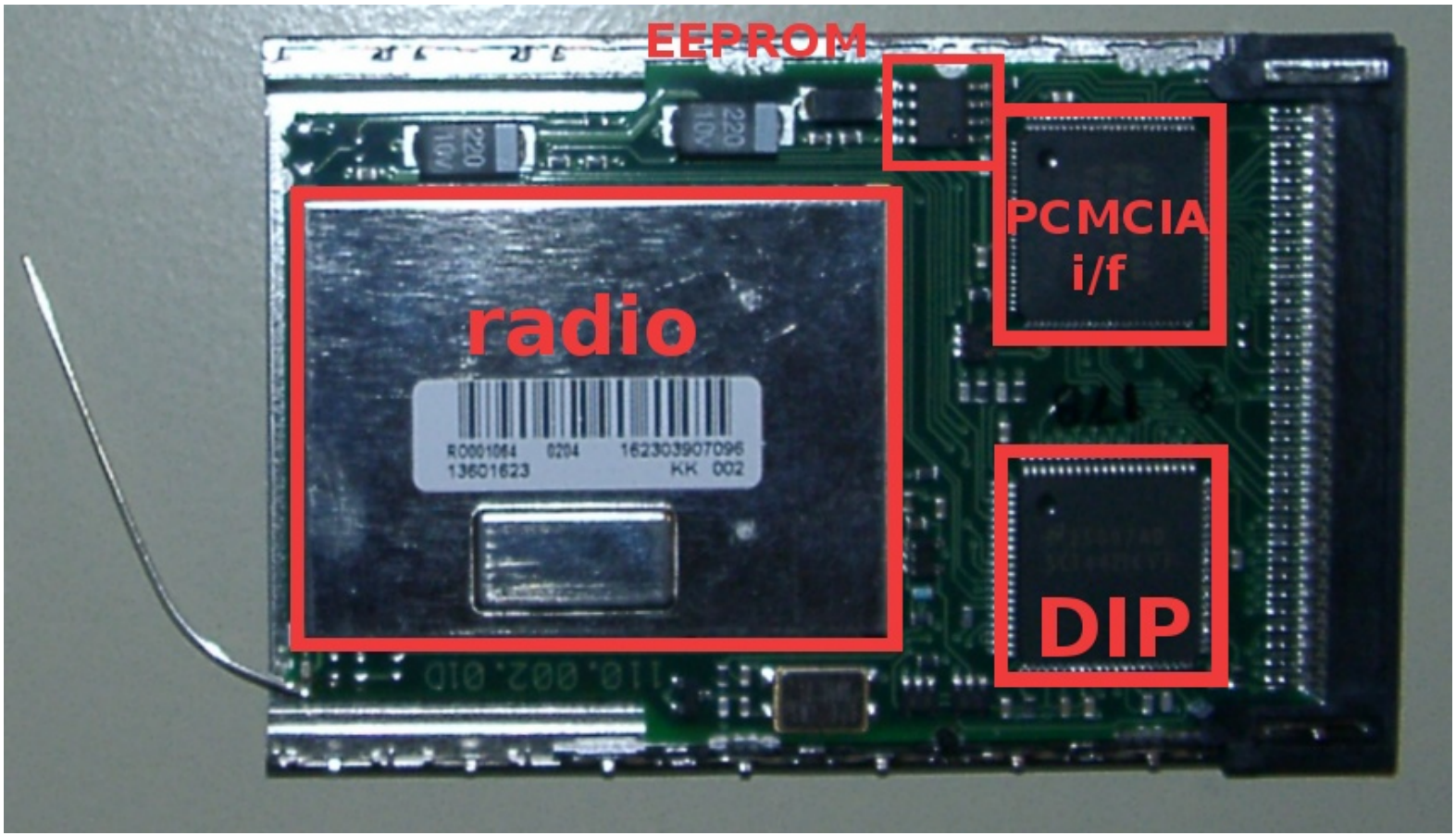
ComOnAir Type III total



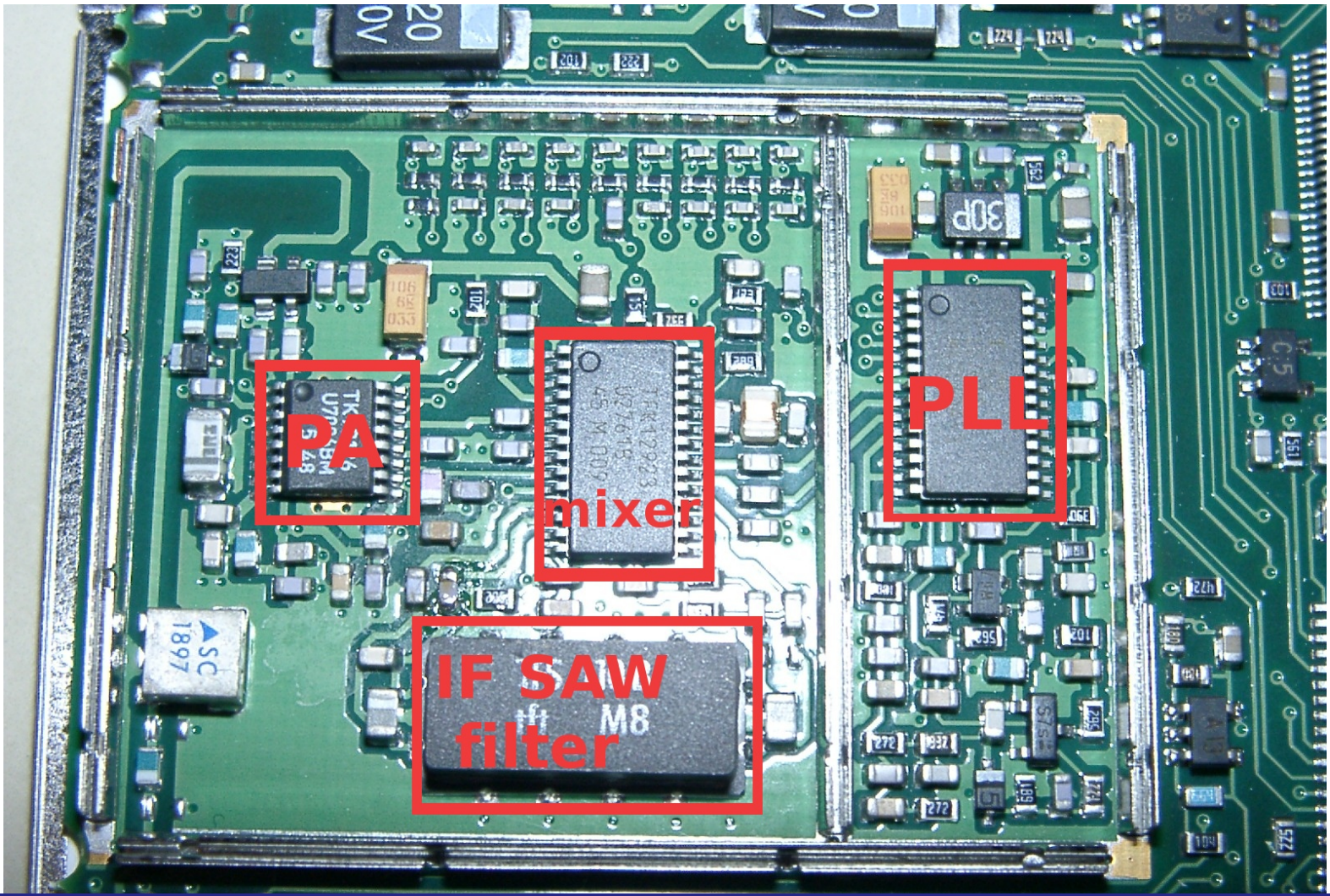
ComOnAir Type III radio



ComOnAir Type II total



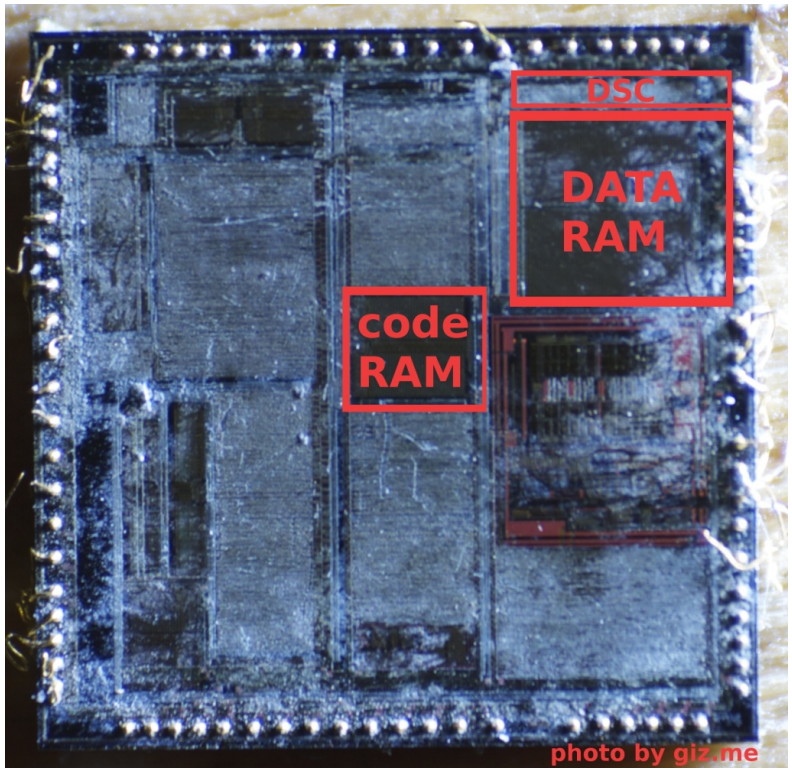
ComOnAir Type II radio



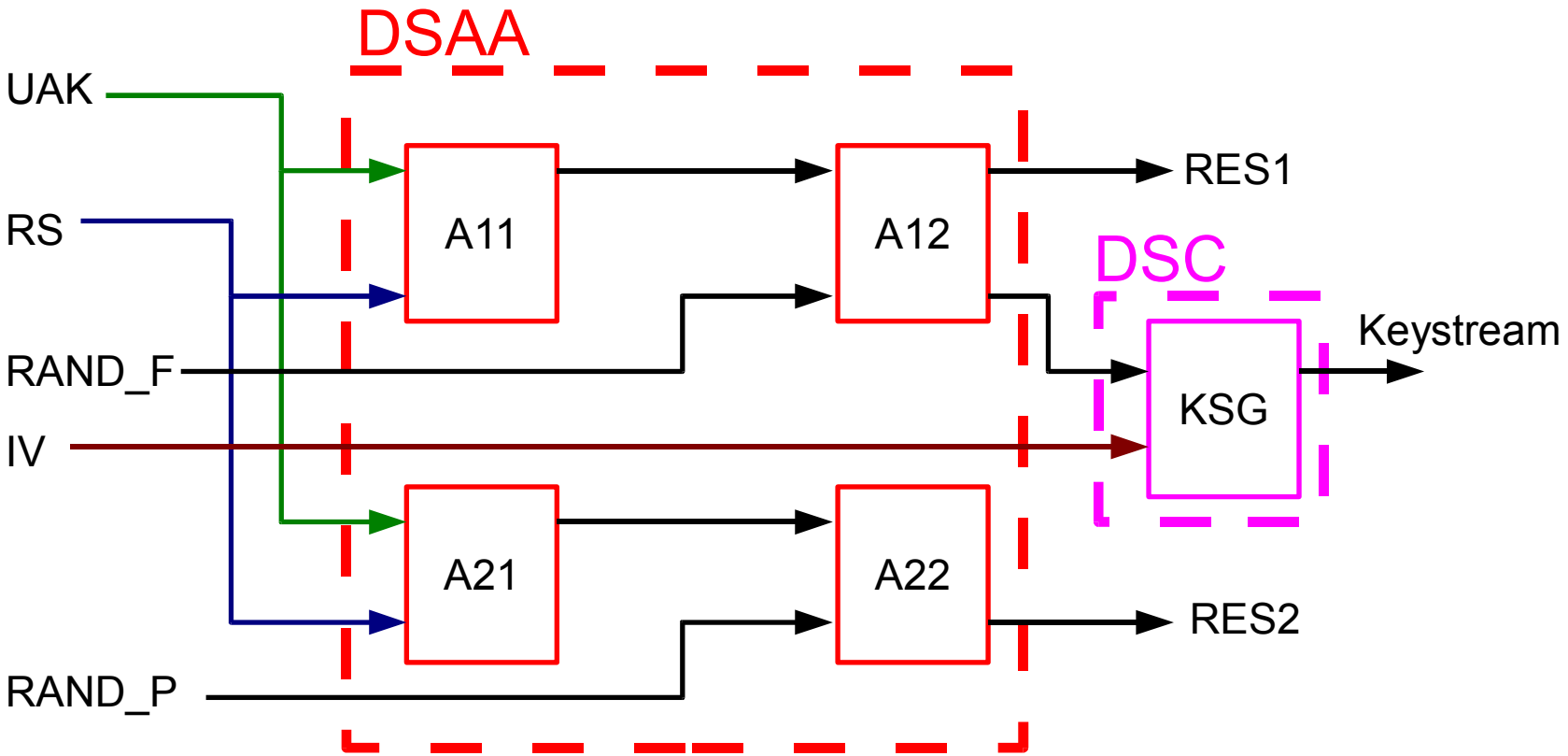
National/Sitel Dedicated Instruction Processor

- Programmed Dispatcher
- All instructions 16 bit
- Coprocessor (Harvard)
- Code RAM 512 byte
- Data RAM 2 kb
- No accu
- No arithmetic or logic commands
- No comparing commands
- Patching program code in runtime (from outside)
- DECT Interception Processor

SC14421



DECT Security overview



However...

- Sometimes, there is no authentication and encryption at all
- Sometimes, only base stations require authentication of a portable part
- Sometimes, no encryption is used

This allows trivial attacks with the right hardware:

Passive sniffing of voice data

When no ciphering is active, it is possible to capture and record all audio data:

- Used a standard PCMCIA DECT controller to implement a DECT sniffer
- A driver for linux for the card has been written
- A userspace utility scans for an active call and tracks the first one found
- Packets are recorded to a pcap file
- The file can later be played with an audio player
- Codec fine tunings needed: sound quality somewhat lacking at the moment

Total costs for the attack: 23 EUR.

Impersonating a DECT base station

When encryption is active, this attack doesn't work. We also implemented an advanced attack:

- Phones often require no authentication of the base station
- Impersonating a base station is possible
- Even when a phone supports encryption, most phones will not abort connection if base station does not
- Calls can be rerouted
- Recording of rerouted calls is possible
- Implementation requires attacker to enter RFPI of base station to impersonate and IPUI of phone to accept

Total costs for this attack: 23 EUR.

DSAA

Used for:

- Authentication of PP
- Authentication of FP
- Key generation for DSC
- Generation of UAK for DECT/GAP devices

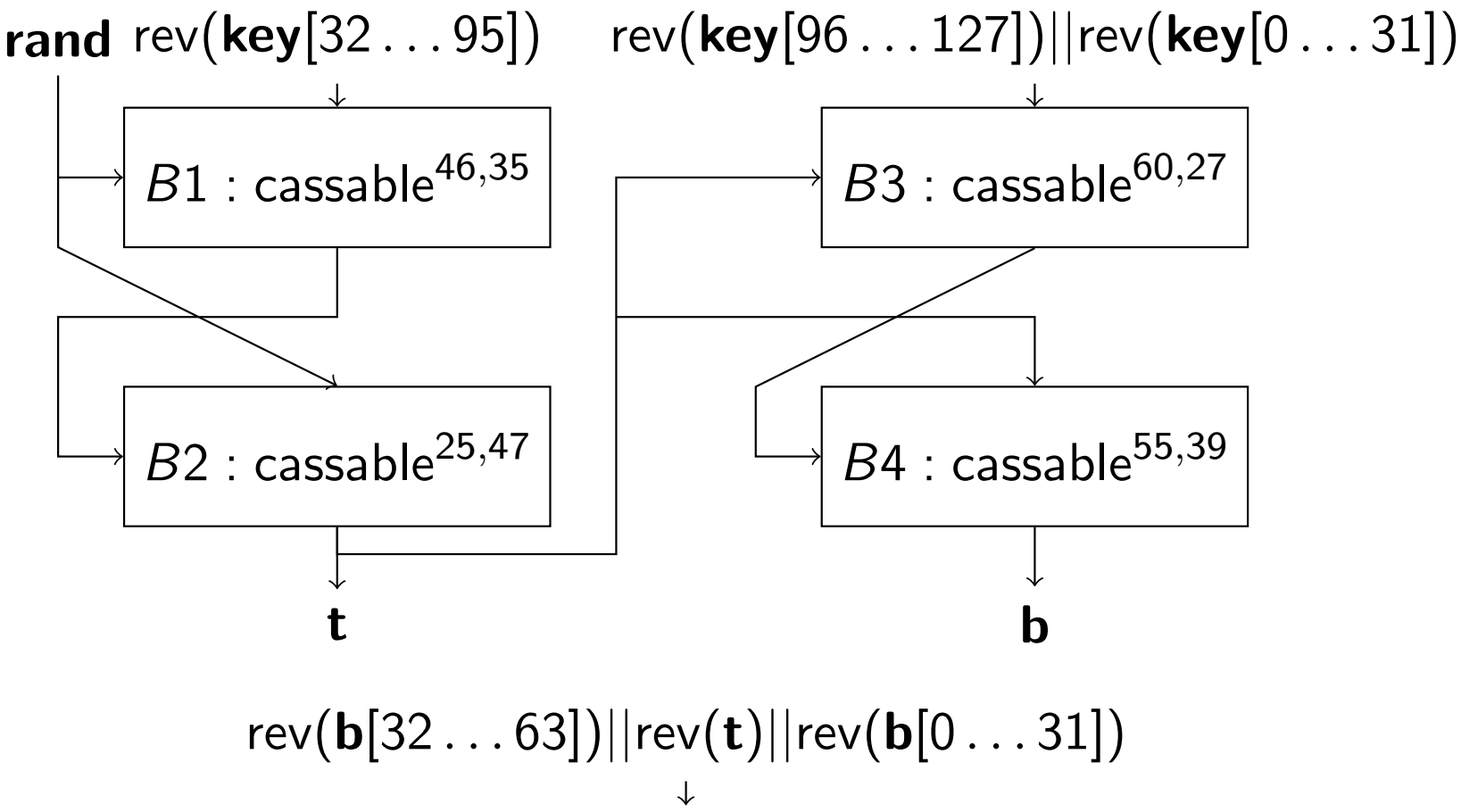
Algorithm is secret

Reversing DSAA

We decided to reverse engineer DSAA:

- A12, A21, and A22 are just simple wrappers around A11
- A11 takes a 128 bit key and a 64 bit random to generate a 128 bit output
- A11 uses four different block ciphers we call *cassable* to generate the output

A11 structure



The cassable block cipher

- cassable is a SPN type construction
  - input is 64 bit
  - key is 64 bit
  - output is 64 bit
  - internal state also has 64 bit
- for key scheduling, a bit permutation is used
- each variant of cassable only differs in this bit permutation
- to add the round key, $\oplus$ is used
- a single cassable invocation does 6 rounds in total
- each round consists of
  - a key addition ($\oplus$)
  - S-box application
  - one of three different mixing functions
- no final key addition

cassable cryptanalysis

- No final key addition at the end, reduces strength to five effective rounds
- At first look, full diffusion after three rounds
- However, full diffusion only after four rounds
- S-Box allows linear cryptanalysis for 2-3 rounds versions
- Practical algebraic attacks possible up to 3 rounds version of cassable
- A differential attack possible on the full cipher with about 16 choosen input-output pairs and computational effort compareable to 2^{37} invocations of cassable
- However, this has no direct impact on DSAA so far

DSAA summary

- Paper *Attacks on the DECT authentication mechanisms* accepted to CT-RSA 2009
- Paper contains description and analysis of DSAA
- C and Java implementations will be available at dedected.org
- A high performance VHDL implementations for FPGA cards is ready, but not yet open source

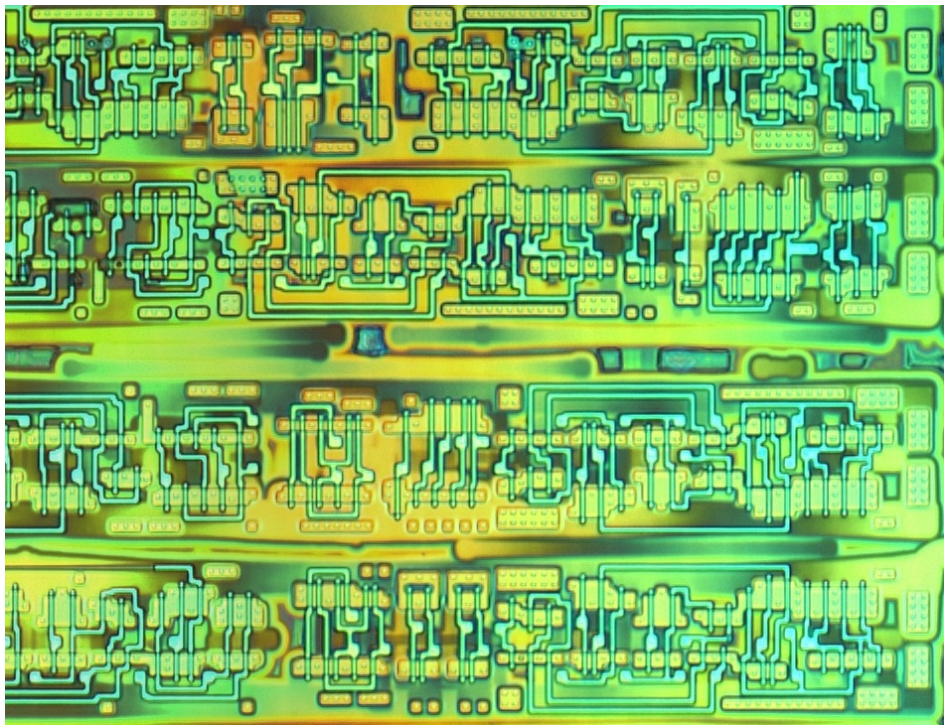
The DECT Standard Cipher

- Did not find any software implementation
- From the ETSI non-disclosure agreement for the DSC: "6. Not to register, or attempt to register, any IPR (patents or the like rights) relating to the DSC and containing all or part of the INFORMATION."
- U.S. Patent 5,608,802, registered by Alcatel, originally registered in Spain in 1993: "A data ciphering device that has special application in implementing Digital European Cordless Telephone (DECT) standard data ciphering algorithm [...]"
- Oops!

DSC: information learned from the patent

- Irregularly clocked combiner with 1 bit of memory
- 3 irregularly clocked LFSRs (2/3) of length 17,19,21
- 1 regularly clocked LFSR (3) of length 23
- key setup: load key, then 40 blank steps (irregularly clocked)
- check whether register is zero after 11 steps, load 1 into every zero register

DSC: hardware based reversing

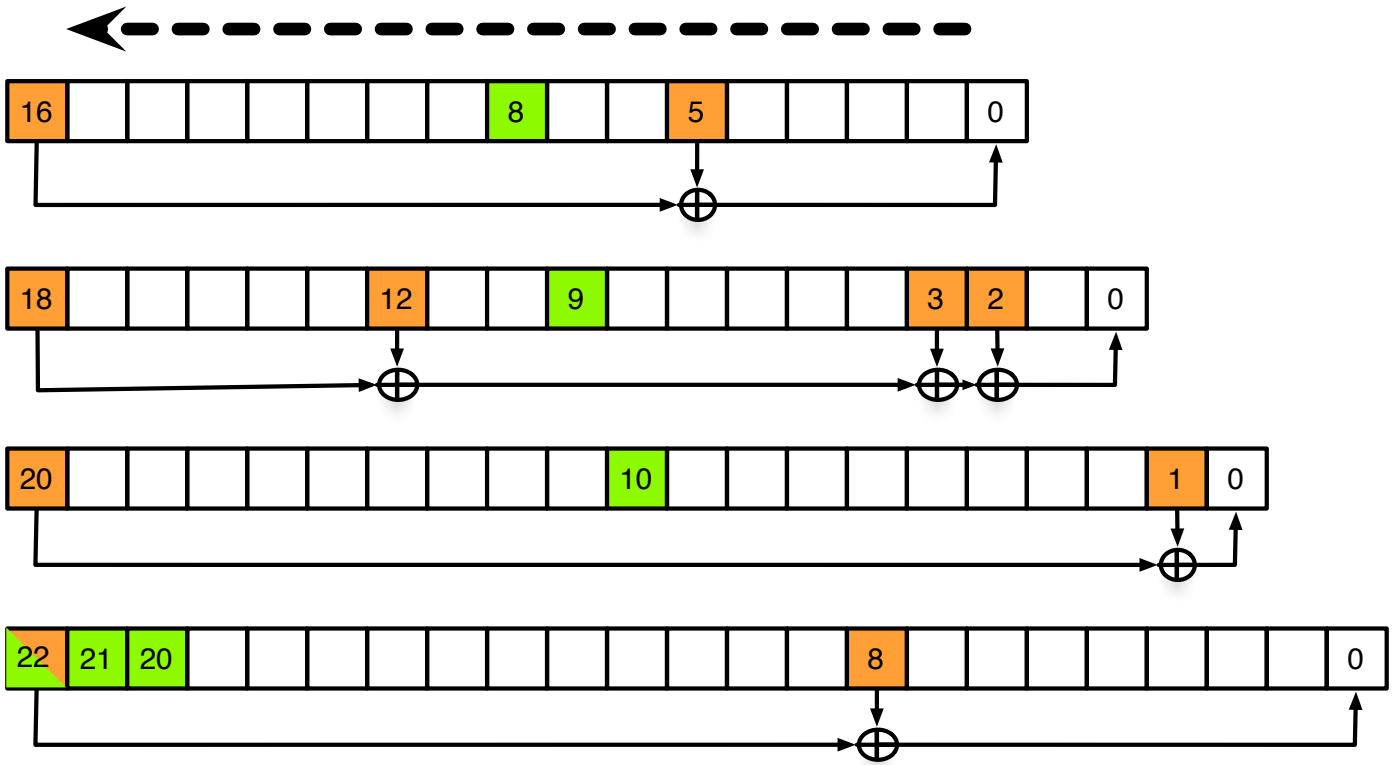


- done by starbug, Karsten Nohl, Flylogic and Mazzo

DSC: software based recovery

- NSC/SiTel SC144xx CPUs have commands to save internal state in DIP memory (11 bytes)
- DIP memory can be read from host
- Can load/save state after and before pre-ciphering (`D_LDS`, `D_WRS`)
- Single-step through key loading to determine feedback taps
- Isolate subset of bits determining clocking differentially in pre-ciphering
- Interpolate clocking function (it's linear actually, could've seen that with bare eyes)
- Output combiner is still missing at the moment

DSC: a diagram



clocks_R1 = 2 + ((R2[9] + R3[10] + R4[22]) mod 2)

clocks_R2 = 2 + ((R1[8] + R3[10] + R4[21]) mod 2)

clocks_R3 = 2 + ((R1[8] + R2[9] + R4[22]) mod 2)

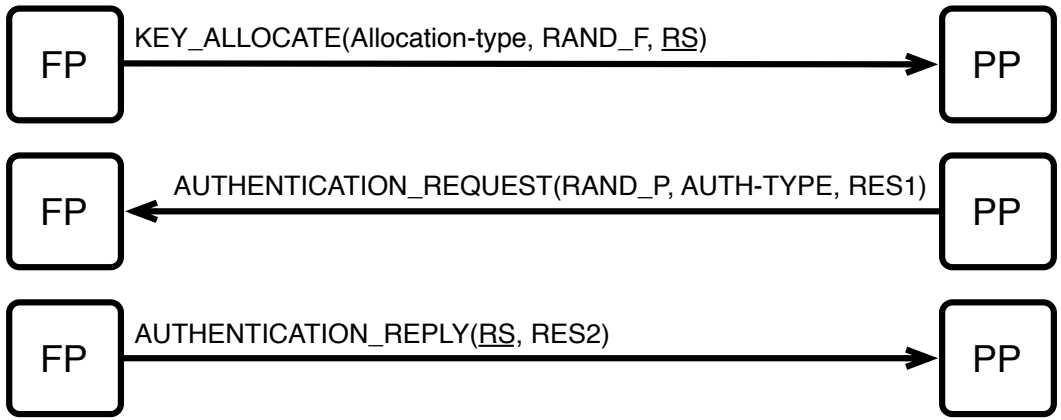
Preliminary analysis of DSC

- Had hoped for R4 doing the clock control (like A5/2)
- Larger internal state: attacks against A5/1 not directly transferrable
- Short pre-ciphering phase

UAK allocation (GAP)

- UAK: 128 bits, master secret shared between FP and PP
- "Pairing mode"
- Authentication code (PIN) shared between FP and PP
- Only depends on 64 bits of `RAND_F` + n bits of PIN
- Entropy for RANDs: where from?

UAK allocation (GAP)



- $RES1 = A12(A11(K, RS), RAND\_F)$
- $UAK = KS' = A21(K, RS)$
- $RES2 = A22(KS', RAND\_P)$

Example a of low-entropy PRNG

```
uint16_t counter;
uint8_t xorvalue;

void next_rand(uint8_t *rand)
{
  int i;

  for(i = 0; i < 8; i++) {
    rand[i] = (counter>>i) ^ xorvalue;
  }

  xorvalue += 13;
}
```

Practical UAK recovery

- Actual entropy of PRNG on last slide: 22 bits
- Grab two challenge-response pairs $((\texttt{RS}, \texttt{RAND\_F}), \texttt{RES1})$ sent by FP off the air
- Each pair acts as 32-bit filter
- Iterate over all 4-digit PINs: $\approx 3 \cdot 2^{35.29}$ DSAA operations
- Assume 0000 PIN: $2^{23.58}$ DSAA operations ($\approx$ 50 secs on an Intel C2D 2.4GHz)
- Impact: impersonate handsets, decrypt encrypted calls etc.

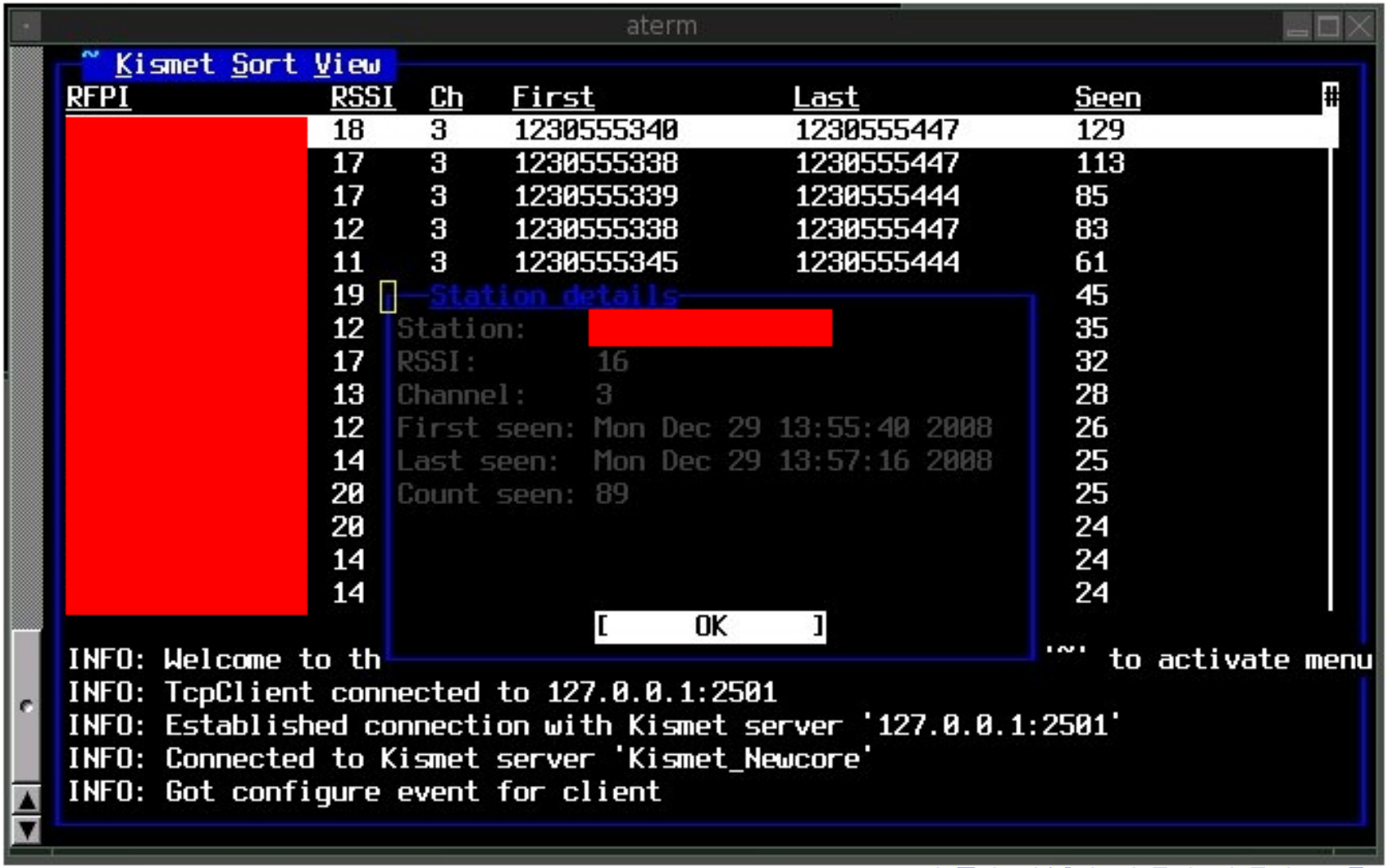
UAKs: Knowing how to attack

- Have more PRNG examples: thus far every one was bad
- Classify/distinguish RANDs, determine underlying DECT stack/PRNG
- Use 16-bit EMC (Equipment Manufacturer's Code) contained in RFPI
- Only 2^{12} devices per EMC, database not available

UAKs of prepaired phones

- Unchartered territory for us so far
- Can read (serial) EEPROM to obtain UAK
- Most phones have test points (for personalisation) in battery case
- Expect externally generated (non)-randomness

Kismet


aterm
Kismet Sort View
RFPI RSSI Ch First Last Seen #
18 3 1230555340 1230555447 129
17 3 1230555338 1230555447 113
17 3 1230555339 1230555444 85
12 3 1230555338 1230555447 83
11 3 1230555345 1230555444 61
19 45
12 35
17 32
13 28
12 26
14 25
20 25
20 24
14 24
14 24
Station details
Station:
RSSI: 16
Channel: 3
First seen: Mon Dec 29 13:55:40 2008
Last seen: Mon Dec 29 13:57:16 2008
Count seen: 89
[OK]
INFO: Welcome to th '~' to activate menu
INFO: TcpClient connected to 127.0.0.1:2501
INFO: Established connection with Kismet server '127.0.0.1:2501'
INFO: Connected to Kismet server 'Kismet_Newcore'
INFO: Got configure event for client

Other stuff (soon on dedected.org)

- Paper about DSAA cryptanalysis
- FPGA implementation project for DSAA
- Kismet
- DSC implementation
- GNU Radio Plugin
- Some infos about the AVM Fritzbox 7270
- More infos about specific phonse

Getting the card

The com-on-air pcmcia type 2 card can be found at:

- The foedbud shop here locally
- on ebay (enough cards are available)

We would like to thank..

- The Chaos Computer Club for great support of the project and providing hardware
- TU-Darmstadt, Uni Luxenburg, Bauhaus Universitt Weimar
- Mazzoo for great help with the linux driver
- Starbug, Karsten, and Flylogics for Chip reverse engineering
- Alcatel for filling the DECT ciphering device patent
- many other people

Contact: team@dedected.org