

Attacks on DECT (deDECTed.org)

Erik Tews

March 13, 2009

- 1 A hacker's view of DECT
 - What is DECT?

- 2 DECT Security (★ 1992, † 2008)
 - Concept
 - DSAA
 - DSC
 - PRNG

- 3 [stuff on dedected.org](http://stuff.on.dedected.org)
 - Tools

Who we are

People from various locations participate in the project:

- 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, counseling
- Wiesbaden** kismet integration
- Darmstadt** Cryptanalysis, packet capturing, FPGA implementations, reverse engineering

in the past, there was...

CT1(+) Analog escommunication, differencnt 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 terminals
- Traffic lights

situation in germany:

- $\approx 30.000.000$ base stations currently in use



Terms

FP Fixed Part (base station)

PP Portable Part (cordless phone/handset)

RFPI Radio Fixed Part Identity

IPIU International Portable Users Identity

DSC DECT Standard Cipher

DSAA DECT Standard Authentication Algorithm

UAK User Authentication Key (shared key between handset and base station)

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
- Synchronizing with base station
- Select best carrier/slot-combination for communication and opening connection

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 framenummer 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



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

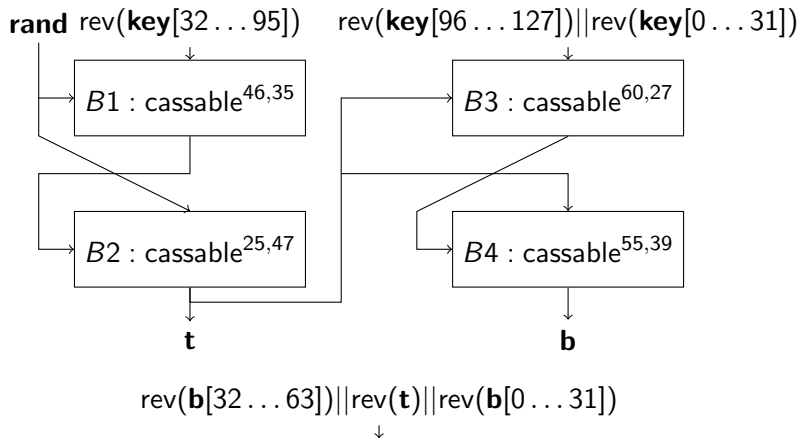
Algorithm was secret

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 chosen input-output pairs and computational effort comparable 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
- Also available on <http://eprint.iacr.org/>
- Paper contains description and analysis of DSAA
- C and Java implementations are available at dedected.org
- A high performance VHDL implementations for FPGA cards is ready, but not yet open source

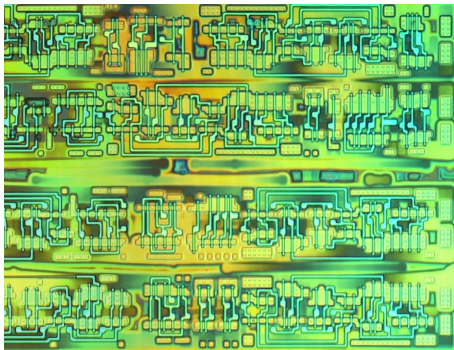
The DECT Standard Cipher

- 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

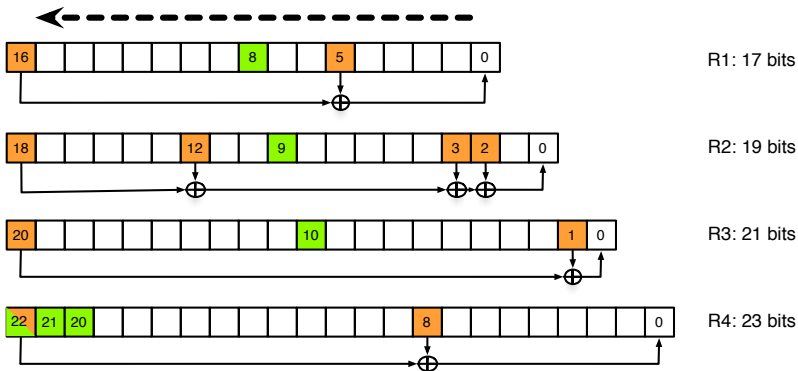


- done by starbug, Karsten Nohl, Flylogic and Mazzo

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



$$\text{clocks_R1} = 2 + ((\text{R2}[9] + \text{R3}[10] + \text{R4}[22]) \bmod 2)$$

$$\text{clocks_R2} = 2 + ((\text{R1}[8] + \text{R3}[10] + \text{R4}[21]) \bmod 2)$$

$$\text{clocks_R3} = 2 + ((\text{R1}[8] + \text{R2}[9] + \text{R4}[22]) \bmod 2)$$

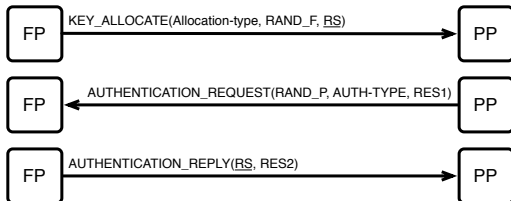
Preliminary analysis of DSC

- 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 ((RS, RAND_F), 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 (≈ 50 secs on an Intel C2D 2.4GHz)
- Impact: impersonate handsets, decrypt encrypted calls etc.

Kismet

```

aterm
~ Kismet Sort View
RFPI      RSSI  Ch  First           Last           Seen
-----
[REDACTED] 18   3  1230555340     1230555447     129
[REDACTED] 17   3  1230555338     1230555447     113
[REDACTED] 17   3  1230555339     1230555444     85
[REDACTED] 12   3  1230555338     1230555447     83
[REDACTED] 11   3  1230555345     1230555444     61
19 [ ] - Station details [OK] 45
12 Station: [REDACTED] 35
17 RSSI: 16 32
13 Channel: 3 28
12 First seen: Mon Dec 29 13:55:40 2008 26
14 Last seen: Mon Dec 29 13:57:16 2008 25
20 Count seen: 89 25
20 24
14 24
14 24

INFO: Welcome to th
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
- Some infos about the AVM Fritzbox 7270
- More infos about specific phonse

We would like to thank..

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

Contact: team@dedected.org