Breaking Hitag2

Hacking at Random 2009

Henryk Plötz, Karsten Nohl
henryk+har09@ploetzli.ch nohl@virginia.edu

August 15th 2009

Hitag types

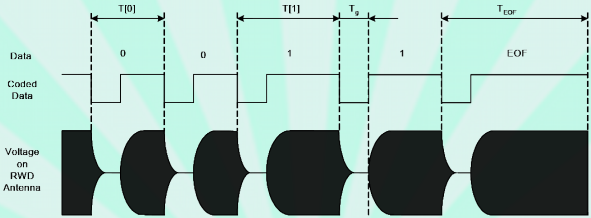
- First step: Google (or bing :-) for information
- Results:
  - Data sheet HT2 DC20 S20, HITAG$^{\text{TM}}$2 Transponders
  - Data sheet HITAG S HTS IC H32/HTS IC H56/HTS IC H48 Transponder IC
- Hitag 1, S and 2
- Hitag S seems to be an update of Hitag 1: more complex protocol compared to Hitag 2, with anticollision etc.

Hitag S: memory

- 32 bit, 256 bit or 2048 bit memory size
- organized in pages of 4 byte each, first (page 0) is 32 bit UID
- first few pages store 48-bit key for reader authentication and 24 bit password for tag authentication

Hitag S: radio

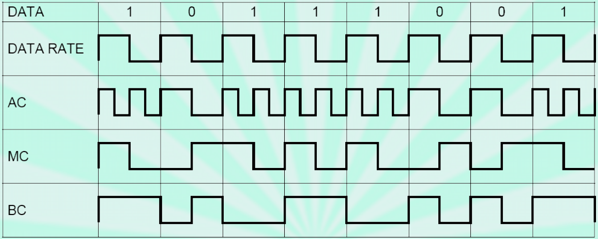
- operating frequency 100. . .150 kHz
- data transmission reader to tag: 100% ASK binary pulse length coding
  - 0 bit: field off for 4. . .10 $\frac{1}{f_c}$, field on for 18. . .22 $\frac{1}{f_c}$
  - 1 bit: field off for 4. . .10 $\frac{1}{f_c}$, field on for 26. . .30 $\frac{1}{f_c}$



- data transmission tag to reader: load modulation with manchester, biphase coding or a special “anticollision” coding (frequency modulation)
- data rates:

Hitag S: radio

- operating frequency 100...150 kHz
- data transmission reader to tag: 100% ASK binary pulse length coding
  - 0 bit: field off for 4...10 $\frac{1}{f_c}$, field on for 18...22 $\frac{1}{f_c}$
  - 1 bit: field off for 4...10 $\frac{1}{f_c}$, field on for 26...30 $\frac{1}{f_c}$
- data transmission tag to reader: load modulation with manchester, biphase coding or a special "anticollision" coding (frequency modulation)



Hitag S: radio

- operating frequency 100...150 kHz
- data transmission reader to tag: 100% ASK binary pulse length coding
  - 0 bit: field off for 4...10 $\frac{1}{f_c}$, field on for 18...22 $\frac{1}{f_c}$
  - 1 bit: field off for 4...10 $\frac{1}{f_c}$, field on for 26...30 $\frac{1}{f_c}$
- data transmission tag to reader: load modulation with manchester, biphase coding or a special "anticollision" coding (frequency modulation)
- data rates:
  - reader to tag: 5.2 kbit/s on average
  - tag to reader: 2, 4 or 8 kbit/s
- Additional "tag talks first" modes for emulation of old, stupid tags

Hitag S: radio

- operating frequency 100...150 kHz
- data transmission reader to tag: 100% ASK binary pulse length coding
  - 0 bit: field off for 4...10 $\frac{1}{f_c}$, field on for 18...22 $\frac{1}{f_c}$
  - 1 bit: field off for 4...10 $\frac{1}{f_c}$, field on for 26...30 $\frac{1}{f_c}$
- data transmission tag to reader: load modulation with manchester, biphase coding or a special "anticollision" coding (frequency modulation)
- data rates:
  - reader to tag: 5.2 kbit/s on average
  - tag to reader: 2, 4 or 8 kbit/s
- Additional "tag talks first" modes for emulation of old, stupid tags

Hitag S: commands (excerpt)

- UID REQUEST Adv: 1100xb
- Anticollision sequence: <bit position of collision><k bits of UID><CRC8>
- Select: 00000b<UID><CRC8> (tag responds with configuration)
- Select but stay quiet: 00000b<UID>0b<CRC8>
- Authentication (when configuration indicates authentication mode): <32 bit RND><32 bit secret data> (tag responds with encrypted password)

etc. pp.

Hitag 2: Memory

- 256 bit of memory in 8 pages of 4 byte each
- page 0 stores UID
- page 3 stores 1 byte configuration, 3 byte tag password
- password mode: page 1 stores reader password
- crypto mode: pages 1 and 2 store 48 bit key
- pages 4 thru 7 are for user data

Hitag 2: Configuration

- Biphase or Manchester modulation
- TTF public modes A (MIRO, EM H400x), B (ISO 11784) and C (PIT compatible)
- Crypto mode or password mode
  - Password mode:
    - "mutual authentication" through passwords: reader sends reader password, tag responds with tag password
  - Crypto mode:
    - reader sends 32 bit nonce, 32 bit authentication token
    - tag responds with encrypted configuration and tag password
    - all further communication is encrypted
- lock bits

Hitag 2: Commands

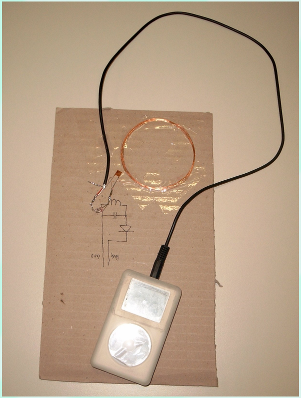
- Activation procedure (password mode):
  - R→T: `11000b`
  - T→R: `<UID (e.g. page 0)>`
  - R→T: `<reader password (e.g. page 1)>`
  - T→R: `<configuration, tag password (e.g. page 3)>`
- Activation procedure (crypto mode):
  - R→T: `11000b`
  - T→R: `<UID (e.g. page 0)>`
  - R→T: `<32 bit random><32 bit authenticator>`
  - T→R: `ENC(<configuration, tag password (e.g. page 3)>)`
  - all further communication encrypted with the cipher stream

Hitag 2: Commands (cont.)

- Read page: `11b<3 bit page number>00b<INV(3 bit page number)>`
- Write page: `10b<3 bit page number>01b<INV(3 bit page number)>`
  Tag repeats command (possibly encrypted!)
  Reader sends page content (32 bits)

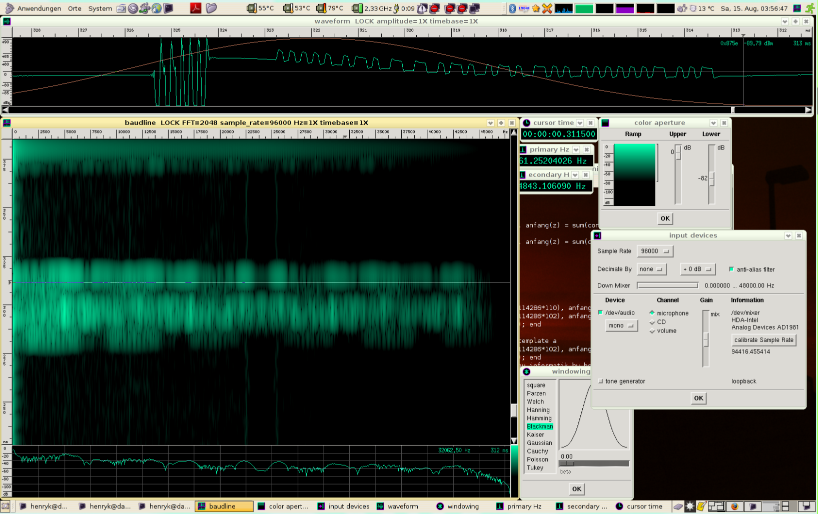
Sniffing

- 125 kHz carrier, max 8 kbit/s data rate with manchester coding → highest frequency is 16 kHz
- Simple diode detector for demodulation
- Soundcard as A/D
- A couple lines of C for decoding



- Reused hardware from `http://www.informatik.hu-berlin.de/~ploetz/analyzing-an-unknown-access-control-system.pdf`

RFID sniffing with baudline



(live demo)

Recording a trace and decoding it

```
henryk@dawn ~/prog/hitag $ arecord -c 1 -f S16_LE -r 48k foo.wav
Recording WAVE 'foo.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono
^CAborted by signal Interrupt...
henryk@dawn ~/prog/hitag $ ./Debug/hitag t foo.wav
 55470:    RWD, 0, 5  11000
 55594:     MC, 4, 0  00110101001100110111000000010001    35337011
 56198:    RWD, 4, 0  01001101010010010100101101010010    4D494B52
 56583:     MC, 4, 0  00000110101010100100100001010100    06AA4854
```

(live demo)

Example of a write transaction

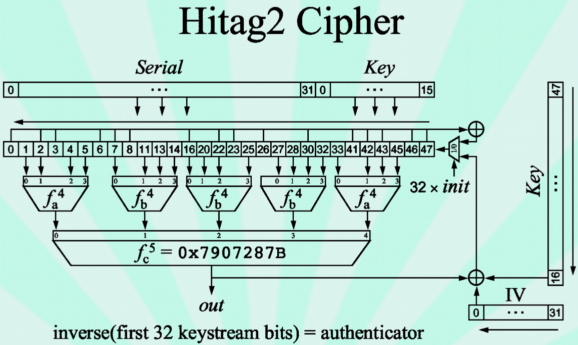
```
henryk@dawn ~/prog/hitag $ ./Debug/hitag t tag1_getsnrreset+write3+write1_nocrypto.wav
273281:    RWD, 0, 5  11000
273406:     MC, 4, 0  00110101001100110111000000010001    35337011
274008:    RWD, 4, 0  01001101010010010100101101010010    4D494B52
274394:     MC, 4, 0  00000110101010100100100001010100    06AA4854
275875:    RWD, 1, 2  1001101100    9B
276051:     MC, 1, 2  1001101100    9B
276432:    RWD, 4, 0  00000110101010100100100001010100    06AA4854
277770:    RWD, 1, 2  1000101110    8B
277946:     MC, 1, 2  1000101110    8B
278327:    RWD, 4, 0  01001101010010010100101101010010    4D494B52
```

Example of an encrypted read transaction

```
henryk@dawn ~/prog/hitag $ ./Debug/hitag t tag2_getsnrrest+fullread_crypto.wav
233119:    RWD, 0, 5  11000
233258:     MC, 4, 0  10100100001111010111000000010001    A43D7011
234132:    RWD, 8, 0  1110001011110010100111010111011110101001010001111011110010011010
                                                          E2F29D77A947BC9A
234846:     MC, 4, 0  00111100010000110100111111010011    3C434FD3
382263:    RWD, 1, 2  0101011000    56
382436:     MC, 4, 0  01110101001100010100111110100010    75314FA2
383990:    RWD, 1, 2  0101111100    5F
384168:     MC, 4, 0  01100110100111100000010000011101    669E041D
385743:    RWD, 1, 2  1100100101    C9
385918:     MC, 4, 0  01110000101010110010100001111100    70AB287C
387719:    RWD, 1, 2  0011001110    33
387894:     MC, 4, 0  11101010111010000111010011111110    EAE874FE
389464:    RWD, 1, 2  0011001011    32
389640:     MC, 4, 0  11110101101101011011111000011001    F5B5BE19
391198:    RWD, 1, 2  1110110100    ED
391377:     MC, 4, 0  11101100011111100110100000100111    EC7E6827
392889:    RWD, 1, 2  0001111000    1E
393062:     MC, 4, 0  00001011110101110000010010110111    0BD704B7
394637:    RWD, 1, 2  1001110110    9D
394815:     MC, 4, 0  11111011010100110010000110010101    FB532195
```

Crypto

- Look what we found on the net:
  `http://cryptolib.com/ciphers/hitag2/`



f_a^4 = `0x2C79` = abc+ac+ad+bc+a+b+d+1

f_b^4 = `0x6671` = abd+acd+bcd+ab+ac+bc+a+b+d+1

(by I.C. Wiener 2006-2007)