Analyzing a Modern Cryptographic RFID System

27th Chaos Communication Congress: "We come in peace"

Milosch Meriac, Henryk Plötz
meriac@openpcd.de, ploetz@informatik.hu-berlin.de

December 29th 2010

Overview

HID offers two proprietary systems

HID Prox (ca. 1991)

- 125 kHz, proprietary modulation and encoding
- No security, read-only
- Cloners readily available (Jonathan Westhues, Chris Paget); demo'd at 26C3

HID iClass (est. 2002); Subject of this talk

- 13.56 MHz, partially ISO 15693 or 14443-B
- Writeable, electronic purse function, multiple applications
- claims (3)DES security

There are many things Wiegand

The word "Wiegand" stands for one of many things:

- John R. Wiegand is a scientist who, in 1975, discovered

There are many things Wiegand

The word "Wiegand" stands for one of many things:

- John R. Wiegand is a scientist who, in 1975, discovered
- the Wiegand effect which is a nonlinear magnetic phenomenon, used in specially produced

There are many things Wiegand

The word "Wiegand" stands for one of many things:

- John R. Wiegand is a scientist who, in 1975, discovered
- the Wiegand effect which is a nonlinear magnetic phenomenon, used in specially produced
- Wiegand wire, the application of which in access control gave rise to the

There are many things Wiegand

The word "Wiegand" stands for one of many things:

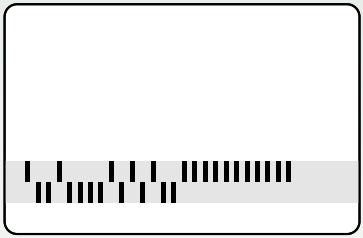
- John R. Wiegand is a scientist who, in 1975, discovered
- the Wiegand effect which is a nonlinear magnetic phenomenon, used in specially produced
- Wiegand wire, the application of which in access control gave rise to the
- Wiegand format, which –used on keycards– indirectly defines the

There are many things Wiegand

The word "Wiegand" stands for one of many things:

- John R. Wiegand is a scientist who, in 1975, discovered
- the Wiegand effect which is a nonlinear magnetic phenomenon, used in specially produced
- Wiegand wire, the application of which in access control gave rise to the
- Wiegand format, which –used on keycards– indirectly defines the
- Wiegand interface and Wiegand protocol that are used between door reader and security panel.

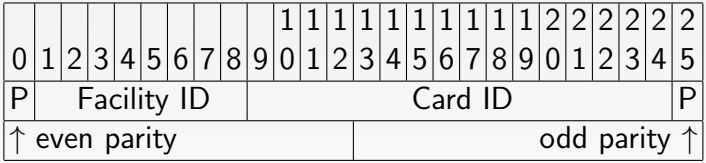
The Wiegand interface is still widely used



- The Wiegand interface has 3 wires:
  - GND
  - DATA0
  - DATA1
- To send a ‘0’-bit, a pulse is sent on DATA0
- To send a ‘1’-bit, a pulse is sent on DATA1
- Very widely used, especially in the U.S.
- Even to this day every HID reader has a Wiegand output

The Wiegand format is a standardized ID layout

- Wiegand wire access control cards could store few bits
- The de-facto standard Wiegand format has 26 bits:



Other formats provide a larger ID space

- HID takes *very* high pride in its support of several formats
- A format is the mapping between a bit string and its fields: facility ID (if any), card ID, parity bits or other checksums
- The fields need not be consecutive
- Different format lengths exist, next to the old 26-bit standard: 35 bit, 37 bit, even 44 bit

HID treats formats as a security feature

Don't succumb to the argument *made by alternate card suppliers* **that proprietary card formats are** *more expensive and are an attempt by manufacturers* **to keep you from buying cards from open sources**. *The use of proprietary formats offered by an OEM or one that is exclusive to a particular site is a desirable best practice.*

Cards with proprietary formats are much more difficult to fraudulently obtain [...]

– HID, "Best Practices in Access Control"

HID offers cards pre-programmed with a variety of formats

- "Corporate 1000" is the HID term for custom 35-bit formats
  - The specific field mapping is unique for each format
  - HID assigns and 'manages' the format
  - Card orders for a format must be authorized by the 'format owner'
- There seems to be a large and unhealthy obsession with formats

HID offers media in multiple physical und logical formats

- Physical: ISO ID-1 card, (adhesive) tag, keyfob
- Logical: 2k or 16k bits (256 or 2k bytes), 2 or 16 areas

Cards are organized in multiple logical units

- The smallest addressable unit is a block of 8 bytes
- Multiple blocks make up an application area
- There are 2 application areas per page
  - A 2k card has 1 page
  - A 16k card can have 1 or 8 pages
    - When 8 pages: each page has 256 bytes
- There are provisions in place for 32k credentials which have two books that each behave as 16k

All page layouts are similar to the 2k/2 case

Block	Content
0	Card Serial Number
1	Flags (App. Limit x, lock bits, etc.)
2	Secure Stored Value Area
3	Key 1
4	Key 2
5	Application Issuer Area
6	Application 1 (secured by Key 1)
⋮	
x	
$x+1$	Application 2 (secured by Key 2)
⋮	
31	

Flags offer some freedom in credential configuration

- Variable application limit allows to customize the memory assignment for the two applications
- Lock bits allow read-only status for individual blocks 6 through 12, or all blocks
- 16 bits of One-Time Programmable (OTP) memory, can only be set from 1 to 0

Layout of block 1

Byte	Content
0	Application Limit
1	OTP
2	OTP
3	Write Lock
4	Chip Config
5	Memory Config
6	E.A.S. (unused yet?)
7	Fuses

The HID access control application is special

- First application on each credential is the HID access control application
  - Only page of 2k/2 or 16k/2 credentials
  - First page of 16k/16 credentials
  - First book of 32k credentials
- Application limit fixed to 0x12
- Secure Stored Value Area not available for purse applications
  - Pages 1–7 of 16k/16 credentials can be used for purse applications: Key 1 is the Debit key, Key 2 is the Credit key

The HID access control application is 13 blocks in size

Block	Content	
6	HID Application Directory	HID Extended Application Directory
7	HID Access Control ID	
8	HID Access Control ID	
9	HID Access Control ID	PIN
10	Password	
11	RFU	
⋮		
18		

iCLASS Security Levels

- **Standard Security**: two keys are shared across all HID readers world-wide. Swiping any standard security card in front of a standard security reader results in "beep-n-blink" of the reader. Cards are provided by HID and have a unique combination of a card ID (not UID) and a facility ID.
- **High Security**: system specific keys for each installation. As the authentication keys differ, Standard Security cards and cards from other system won't result in 'beep-n-blink" of the reader.
- **iCLASS Elite**: like *High Security*, but keys maintained by HID – customer gets preprogrammed cards.

Configuration Cards to switch readers to High Security

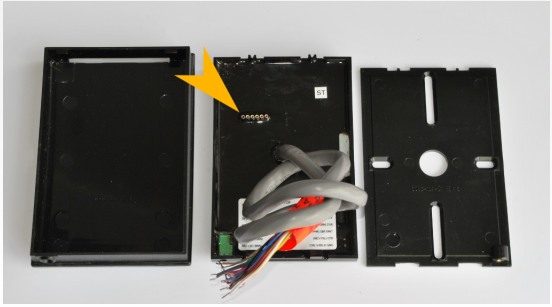
- card programmers like CP400 can create reader configuration cards
- configuration cards turn readers into high security mode by updating keys
- can optionally enable key rolling to switch all cards presented to the reader from Standard Security to the new key in High Security mode.

Breaking RW400 reader security

- same keys used on all Standard Security reader, incentive is high to extract keys
- break a single reader once and enter anywhere
- RW400 readers are widely available on Ebay, good choice as RW means "Read & Write" support
- RW400 – model number 6121AKN0000 attacked

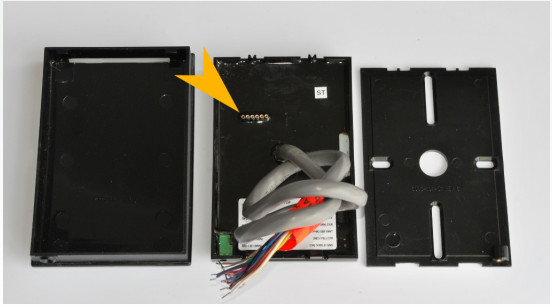
In-System-Programming Connector

- same keys used on all Standard Security reader, incentive is high to extract keys
- breaking open a reader reveals **PIC18F452** CPU
- 6 pin connector on the back is a PIC In-System-Programming connector
- connector obfuscated by swapping pin 1 & 3



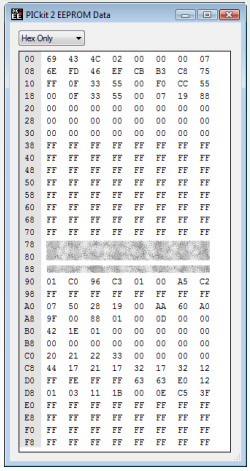
Breaking PIC18F452 Copy Protection

- created custom ICSP to delete single memory pages
- erasing boot block, flashing dumper firmware there
- erase everything except of the boot loader, putting dumper firmware on the end
- dumper firmware outputs FLASH & EEPROM content over UART
- joining binary dumps in a single hex file – flashing readers with In-System-Debug enabled



Key Spotting - Finding the (3)DES Encryption Keys

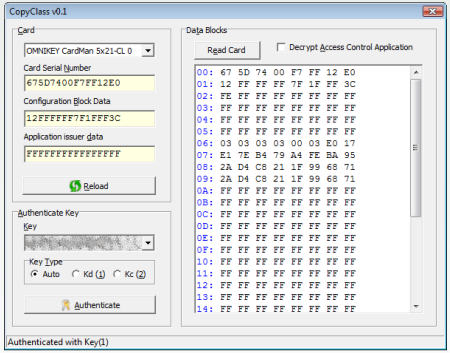
- easy to spot in the 256 byte EEPROM dump
- four 8 byte blocks look random in the dump
- using In-System-Programmer, narrowing down keys by changing bytes
- changes in the DES authentication key will stop "beep-n-blink"
- changes in the 3DES encryption key will result in garbled Wiegand packets
- stored keys need to be reverse-permuted to make them usable in a standard OMNIKEY reader



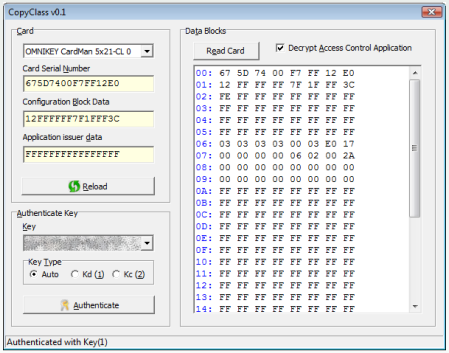
Reading & Writing the protected HID Access Control Application

- using the previously acquired keys with a OMNIKEY 5321/6321 RFID reader
- reading & writing to the HID Access Control Application is possible
- reading and decrypting configuration cards is possible as well
- copying cards is possible as the reader ignores the hardware CSN

Reading & Writing iCLASS - Weaponized



Encrypted iCLASS card



Decrypted iCLASS card

That ain't no 15693

```
5016386830 ( 415290) RWD(  8):     C  0A
5117130050 ( 415300) RWD(  8):     C  0A
5118211255 (  94395) TAG(  0):   u C
5118519430 ( 415300) RWD(  8):     C  0C
5119278915 (3209430) TAG( 80):        A2 E2 15 E0 FE 5F 02 5C 14 D7
5122865580 (2831550) RWD( 72):     C  81 A2 E2 15 E0 FE 5F 02 5C
5126043015 (3209430) TAG( 80):        12 15 AF 00 F7 FF 12 E0 6F 7D
5130416390 ( 415300) RWD(  8):     C  00
5131157435 (  94395) TAG(  0):   u C
5135117960 (2831550) RWD( 72):     C  81 12 15 AF 00 F7 FF 12 E0
5138294355 (3209430) TAG( 80):        12 15 AF 00 F7 FF 12 E0 6F 7D
5142636920 (1321390) RWD( 32):     C  84 00 73 33
5151130550 ( 717330) RWD( 16):     C  88 02
5152173775 (2605302) TAG( 64):     C  FF FF FF FF AD FF FF FF
5171918610 (2831560) RWD( 72):     C  05 1B 80 72 59 A0 CE 7B 3E
5175076415 (1397046) TAG( 32):     C  C1 57 1F 2B
5196043170 (4341720) RWD(112):     C  87 02 FF FF FF FF AC FF FF FF A4 E7 42 63
5203093625 (3209430) TAG( 80):        AC FF FF FF FF FF FF FF B3 41
5207496660 (1321400) RWD( 32):        0C 01 FA 22
5209163405 (3209430) TAG( 80):        12 FF FF FF 7F 1F FF 3C 8C 87
```


The air protocol has ISO 15693 bit encoding but custom commands

- Selection
  - Reader command `0A`, card responds with a single SOF
  - Reader command `0C`, card responds with a fixed identifier
  - Reader command `81` followed by the identifier from the previous step, card responds with CSN
  - Reader command `81` followed by CSN, card responds with CSN

```
5117130050 ( 415300) RWD(  8):     C  0A
5118211255 (  94395) TAG(  0):   u C
5118519430 ( 415300) RWD(  8):     C  0C
5119278915 (3209430) TAG( 80):        A2 E2 15 E0 FE 5F 02 5C 14 D7
5122865580 (2831550) RWD( 72):     C  81 A2 E2 15 E0 FE 5F 02 5C
5126043015 (3209430) TAG( 80):        12 15 AF 00 F7 FF 12 E0 6F 7D
```


The air protocol has ISO 15693 bit encoding but custom commands

- Selection
- Authentication
  - Reader command 88 02, card responds with stored value block
  - Reader command 05 followed by authentication, card responds with authentication

```
5151130550 ( 717330) RWD( 16):     C  88 02
5152173775 (2605302) TAG( 64):     C  FF FF FF FF AD FF FF FF
5171918610 (2831560) RWD( 72):     C  05 1B 80 72 59 A0 CE 7B 3E
5175076415 (1397046) TAG( 32):     C  C1 57 1F 2B
```


The air protocol has ISO 15693 bit encoding but custom commands

- Selection
- Authentication
- Writing
  - Reader command 87 followed by block number, new contents, authenticator, card responds with new block contents

```
5196043170 (4341720) RWD(112):     C  87 02 FF FF FF FF AC FF FF FF A4 E7 42 63
5203093625 (3209430) TAG( 80):        AC FF FF FF FF FF FF FF B3 41
```


The air protocol has ISO 15693 bit encoding but custom commands

- Selection
- Authentication
- Writing
- Reading
  - Reader command 0c followed by block number and CRC, card responds with block contents and CRC

```
5207496660 (1321400) RWD( 32):        0C 01 FA 22
5209163405 (3209430) TAG( 80):        12 FF FF FF 7F 1F FF 3C 8C 87
```


Authenticators seem to be 4 bytes

- Mutual authentication:
  - No random number from card, but stored value block is part of authentication
  - 4 byte random number from reader
  - 4 byte authenticator from reader
  - 4 byte authenticator from card
- Write authentication:
  - 4 byte authenticator
  - Strange behaviour for special blocks:
    - Writing key means transmitting XOR of current and desired value
    - Writing to stored value block swaps low and high word
- No message authentication!
  - CRC are similar to ISO 15693 but with custom post XOR

All the king's horses and all the king's men

- Authentication key derivation based on CSN, no binding between CSN and anything else
- Verbatim copy of blocks is possible
  - Content encryption does not help against impersonation
- No MAC: Man in the middle attacks lead to privilege escalation
  - Use an authorized card to survive the mutual authentication, then do whatever you want
- Standard Security is broken, on the order of Legic Prime or HID Prox

Open Questions

- Exact algorithms for key derivation
- Algorithm for authentication
- Full card and reader emulation
- Replay of write commands
- Using unexpected commands
  - Would 88 00 work, or similar?

The End

Questions?

More information and paper on PIC firmware extraction at
http://www.openpcd.org/HID_iClass_demystified