Practical RFID Attacks

Chaos Communication Camp 2007

Milosch Meriac   Henryk Plötz
meri@openpcd.org   henryk@ploetzli.ch

Chaos Communication Camp 2007

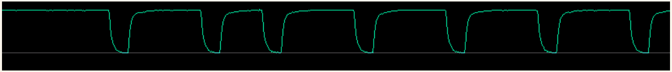
2007-08-10

ISO 14443

- international standard for Proximity Integrated Circuit Cards (PICC)
- works on 13.56MHz
- four parts:
  1. physical characteristics
  2. radio frequency power and signal interface
  3. initialization and anticollision
  4. transmission protocol
- two types (parts 2 and 3):
  - A most common, used in Mifare
  - B less common, transmits more power to the card, used in some ePassports

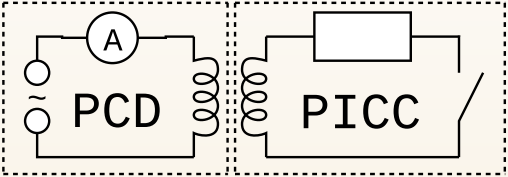
ISO 14443A Modulation: PCD to PICC

- type A uses 100% Amplitude Shift Keying (ASK) for the data from PCD to PICC
  - the carrier is switched off for very short amounts of time
  - easily receivable over a long range (as in 5m, maybe 10m, maybe more, depending on your receiver)
- easy to see in amplitude demodulated signal:



ISO 14443A Modulation: PICC to PCD

- type A uses load modulation on a 847kHz subcarrier for the data from PCD to PICC
  - the card repeatedly switches a load (a resistor) on and off



  - very weak signal: about 60dB to 80dB below the carrier signal
  - hard to receive over distances of more than a dozen cm, very hard to receive over more than 2m

Anticollision

- ISO 14443 defines an anticollision method to handle more than one card in the field
- Each card has a UID (either fixed or randomly generated) of 4, 7 or 10 bytes
- Upon reader request all cards simultaneously transmit their UID in the clear
- Reader detects collisions and resolves them through binary search

Mifare Ultralight

- ISO 14443A (like all Mifare cards)
- inexpensive Mifare type
- 16*4=64 bytes of storage: 10 bytes read-only/factory-programmed (including 7 bytes UID), 6 bytes PROM (including 2 bytes for lock-bits), 48 bytes usable memory
- no encryption, no security features (besides the unchangeable UID)

Mifare Ultralight Memory Layout

Offset				
0x00	UID	UID	UID	CC
0x04	UID	UID	UID	UID
0x08	CC	XX	Lock	Lock
0x0c	OTP	OTP	OTP	OTP
0x10	User area			
0x14				
0x18				
0x1c				
0x20				
0x24				
0x28				
0x2c				
0x30				
0x34				
0x38				
0x3c				

Mifare Classic

- standard Mifare type, very common
- 1k or 4k of storage, organized into sectors organized into blocks of 16 bytes each
  - 1k: 16 sectors of 4 blocks
  - 4k: 32 sectors of 4 blocks, plus 8 sectors of 16 blocks
- Each sector has two keys (A and B) that can be given different access rights (keys and rights are stored in the last block of each sector)
- Proprietary stream cipher called “Crypto1”, key size is 48 bits

Mifare Classic (contd.)

- On-air communication is encrypted with a session key, derived during challenge-response authentication
- 4 byte UID
- Special “value” block types to store monetary values in a block with “INCREASE” and “DECREASE” commands

Mifare Classic Memory Layout

Offset			
0x00	Manufacturer block		
0x10	User area		
0x20			
0x30	Key A	Access bits	Key B
0x40	User area		
0x50			
0x60			
0x70	Key A	Access bits	Key B
0x80	User area		
0x90			
0xa0			
0xb0	Key A	Access bits	Key B

⋮

Mifare DESfire

- Compatible to ISO 14443-4
- Uses DES or Triple-DES for security
- 7 byte UID
- Not yet very widely used

T=CL

- Transmission protocol, specified in ISO 14443-4
- Defines a way to transmit APDUs (Application Protocol Data Unit), similar to contact-based ISO 7816 smart-cards
- APDU commands standardized in ISO 7816-4 (e.g. SELECT FILE, READ BINARY, READ RECORD)
- Can be handled in software like a normal, contact-based smart-card
- No security specified in ISO 14443, instead just use the existing ISO 7816 infrastructure, including Secure Messaging

Electronic Passports (contd.)

- On-air transmission is either unencrypted, or secured through Secure Messaging following BAC (Basic Access Control)
  - Challenge-response authentication for key derived from optical MRZ
  - Session encrypted with session key, derived during authentication
- Other optional security measures include encryption of the data on the passport, or Extended Access Control (EAC) for access to advanced biometric data

Sniffing results: Mifare Classic

Time[us]	Size	Src	Content
0	7 bits	R	26
157	2 bytes	C	04 00
34158	2 bytes	R	93 20
270	5 bytes	C √	B4 79 F7 D7 ED
46431	9 bytes	R √	93 70 B4 79 F7 D7 ED C7 27
865	3 bytes	C √	08 B6 DD
23127	4 bytes	R √	60 00 F5 7B
492	4 bytes	C	F3 FB AE ED
10515	8 bytes	R	**7C** 74 **07** EB **0F** 7B D5 **1B**
775	4 bytes	C	**3D** **0E** **A0** E2
59213	4 bytes	R	**65** **8D** 65 1F
449	18 bytes	C	52 F6 46 35 **89** **BA** E2 E9 B2
			2D **F8** CD **AE** **C8** **6C** B2 **DE** 04

Source is Reader (R) or Card (C), **boldface** indicates bytes with wrong parity bit, √ indicates correct checksum, all content bytes are in hex

Detailed explanation

26 →	REQA
→ 04 00	ATQA
93 20 →	ANTICOL, Cascade level=1
→ B4 79 F7 D7 ED	UID plus check byte
93 70 B4 79 F7 D7 ED →	SELECT with UID
→ 08 B6 DD	SAK plus CRC

Detailed explanation (contd.)

60 00 F5 7B →	AUTH1A block 0 +CRC
→ F3 FB AE ED	? rand1?
7C 74 **07** EB **0F** 7B D5 **1B** →	? H(rand1),rand2?
→ **3D 0E A0** E2	? H(rand2)?
65 8D 65 1F	READ block 0, +CRC, enc
→52 F6 46 35 **89**...	content block 0, +CRC, enc

How to use an oscilloscope to examine a random HF RFID communication (13.56MHz or 100kHz range

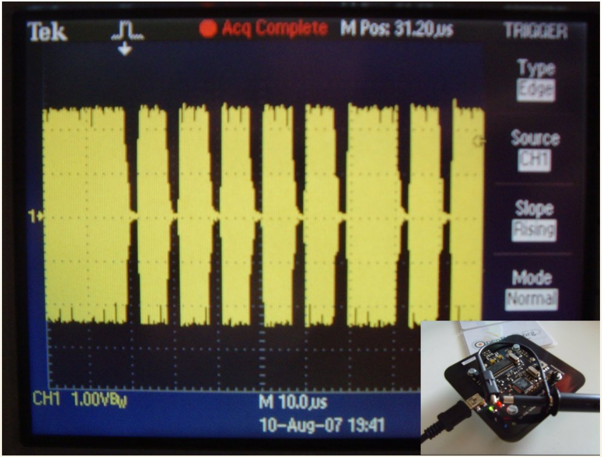


Figure: sniffed MIFARE 1K sector reading (ISO 14443A)

How to use an oscilloscope to examine a random HF RFID communication (13.56MHz or 100kHz range

- Connect the ground cable to the connetor tip like seen on the page before
- Put the resulting Loop Antenna between RFID card and RFID Reader
- Press "Autoset" or equivalent on your oscilloscope to fit waveform (Oscilloscope selects AC mode etc.)
- Move the trigger level slowly between 30 to 110 percent of the average waveform envelope till you get a stable picture like on the page before
- For your first tests make sure that you have constant data transmissions between reader and tag to get a feeling for trigger level selection

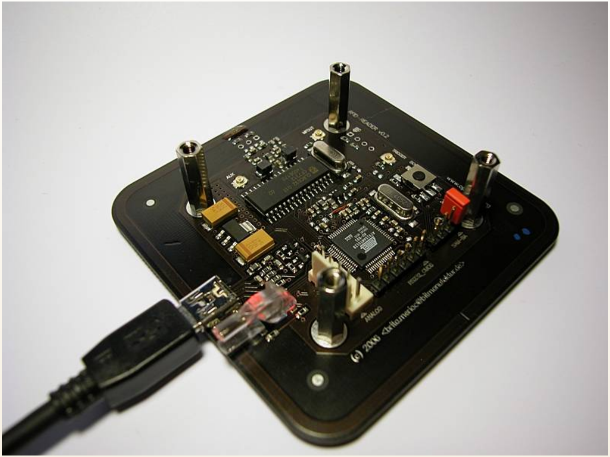
What to do with the data you see

- Verify the carrier frequency
- try to map the modulation patterns to known modulation
- figure out what bitrates are used
- check how long the transations last
- short transactions of only few bytes are a clear indication of UID based authentication schemes - easy to break
- check if packets are constantly changing or if you get fixed patterns which will enable replay attacks

Building your own Loop Antenna

- for building a much better Loop Antenna for few dollars worth of material see the presentation papers in our RFID sniffer section of 22C3 talk
- for serious attacks you may want to use an high performance OpAMP to buffer and amplify the resulting signal near the antenna
- OpenPICC provides a high quality HF frontend as a reference for long range sniffers
- GNUradio fits ideally your demands for long range sniffing attacks - pre-amplification and signal buffering is vital in this case

OpenPCD Hardware Overview



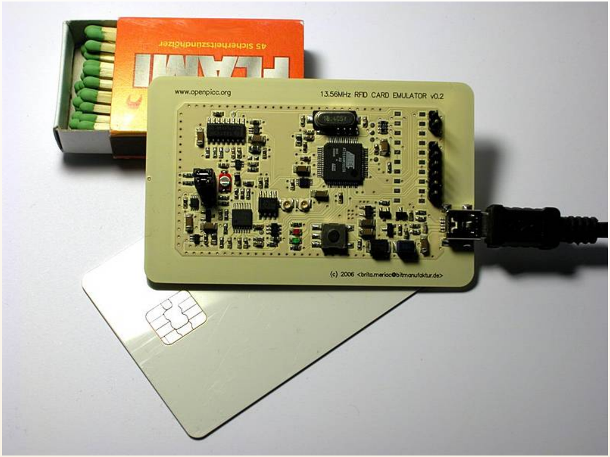
OpenPCD Hardware Overview

- 32 bit ARM-based Open Source RFID Reader/Writer (AT91SAM7S128)
- supported in LibRFID
- stand-alone operation possible
- CL RC632 based chipset - well supported in LibRFID
- native MIFARE support
- JTAG debug interface
- I2C & RS232-CMOS interface

OpenPCD Special Features

- DMA accelerated sampling of MFOUT signals for Tag-Reader communication
- DMA accelerated transmission of freely selectable bitpatterns for Reader-Tag communication
- DMA clock is derived directly from carrier signal - synchronous sampling possible
- Output of modulation/demodulation steps on analog ports for inspecting signal quality of Emulators
- Carrier-derived hardware timer can be used to create test patterns for sniffers and emulators
- Modulation depth and bitrates freely selectable
- LibRFID ported to OpenPCD - stand-alone RFID brute force cracker is simple to compile

OpenPICC Hardware Overview



OpenPICC Hardware Overview

- 32 bit ARM-based Open Source RFID Sniffer/Emulator (AT91SAM7S256)
- stand-alone operation possible
- JTAG debug interface
- I2C & RS232-CMOS interface

OpenPICC Special Features

- DMA accelerated sampling of demodulated reader-tag-communication (binary)
- analog to binary conversion treshold level freely selectable by using a D/A-converter-controlled comparator
- DMA accelerated transmission of freely selectable bitpatterns for Tag-Reader communication
- DMA clock is derived directly from carrier signal - synchronous sampling possible
- carrier signal is regenerated by using a PLL to provide clock during modulation pauses
- application software available for logging and decoding Reader-Tag-Communication (ISO14443A) with OpenPICC

Combine your tools wisely

- OpenPCD can be connected to OpenPICC over TTL-based serial interface
- a stand alone battery powered device can be created with OpenPCD/OpenPICC clones RFID card on-the-fly without a computer needed
- OpenPICC/OpenPCD can be easily used to gather encrypted MIFARE communication
- within next days we will publish some transaction with known keys to support Crypto-Analysis of the encryption algorithms used for MIFARE

Denial of service

- OpenPICC hardware supports emulating an unlimited number of tags in the reader field
- can be used to verify anticollision algorithms used
- 13.56MHz RFID protocols can be modified to verify protection against fuzzing attacks

Our TODO-List

- get finally anticollision running in OpenPICC - very important prerequisite for emulation RFID cards
- provide tons of samples of MIFARE standard 1K communications with known keys to enable cryptoalaysis
- port OpenPCD and OpenPICC operating system to FreeRTOS in the hope that this will attract more users in active participation in our project

Thanks for listening.