OWASP

The Open Web Application Security Project

About me

- Name: Christian Becker
- Security Consultant at Context IS
- Interested in all kind of hacking

Agenda

- – Background story
- – RFID
- – Exploring an unknown tag
- – NFC in Web Applications

Background Story

– RFID tags + PIN used in access control

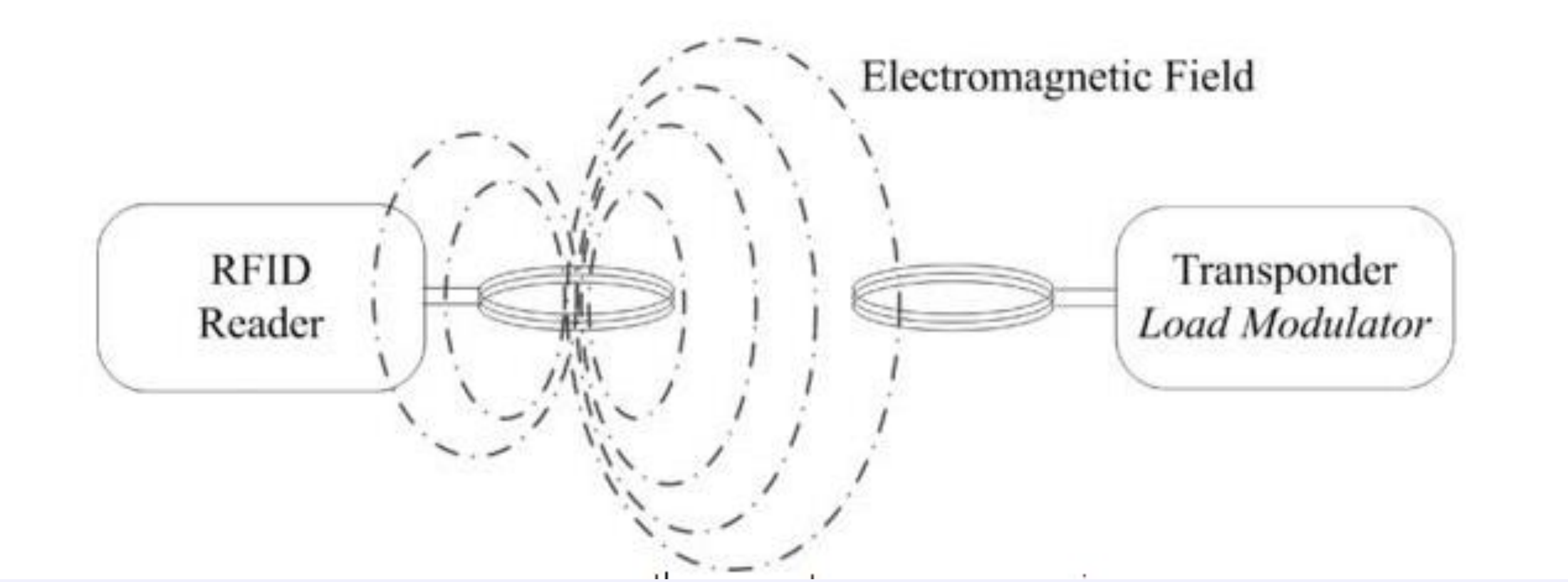
Radio-Frequency Identification

- Wireless use of electromagnetic fields to transfer data
- Used for tracking/identifying objects
- Active & passive transponder

Common usage of RFID

Band	Range	Remarks
125 KHz or 134 KHz (LF)	10cm	Animal identification, factory data collection/ livestock tracking
13.56 MHz (HF)	10cm – 1m	Ticketing (Public transport), contactless payment, data transfer applications, etc
443 MHz (UHF)	1m – 100m	Warehouse / logistics
…		

How does it work?



Carrier & Modulation

- – Energy is sent via electromagnetic waves
- – Influenced by Performance, Frequency & Phase to encode messages

$\Rightarrow$ Modulation

- – The unchanged electromagnetic wave is called “Carrier”

That’s all we need for now

Identifying the tag

1. Recon & Setup
2. Low, high or ultra-high frequency?
3. Obtaining the data trace
4. Examining the data trace
5. (Cloning / replaying the data trace)

1. Recon & Setup

- About the keyfob
  - Colour: blue
  - Tagged with a 10 digit number (HEX?)
  - Probably a passive RFID tag

1. Recon & Setup
   - OS: Kali Linux
   - Proxmark3 (RFID/NFC reader/writer/simulator) with custom firmware [https://github.com/iceman1001/proxmark3]
   - Smartphone with NFC support
   - Keyfob

2. Low, high or ultra-high frequency?

- Smartphone/Proxmark3 with high frequency antenna

⇒ Didn’t receive any data

- Proxmark3 with low frequency antenna

⇒ Works

⇒ The keyfob is a low frequency RFID Tag

3. Obtaining the data trace

Demo time

4. Examining the data trace

   – No changes when repeating the previous step

   ⇒ There is some kind of recurrence

   ⇒ Probably no crypt/replay measurements involved

   ⇒ Very likely that the signal can be repeated

What’s next?

- Demodulating the signal (analog capture to bitstream
- Decoding the signal

Amplitude-shift keying (ASK)

"Amplitude-shift keying (ASK) is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. In an ASK system, the binary symbol 1 is represented by transmitting a fixed-amplitude carrier wave and fixed frequency for a bit duration of T seconds. If the signal value is 1 then the carrier signal will be transmitted; otherwise, a signal value of 0 will be transmitted."

http://en.wikipedia.org/wiki/Amplitude-shift_keying

⇒ Luckily, this can be done by the proxmark3 software

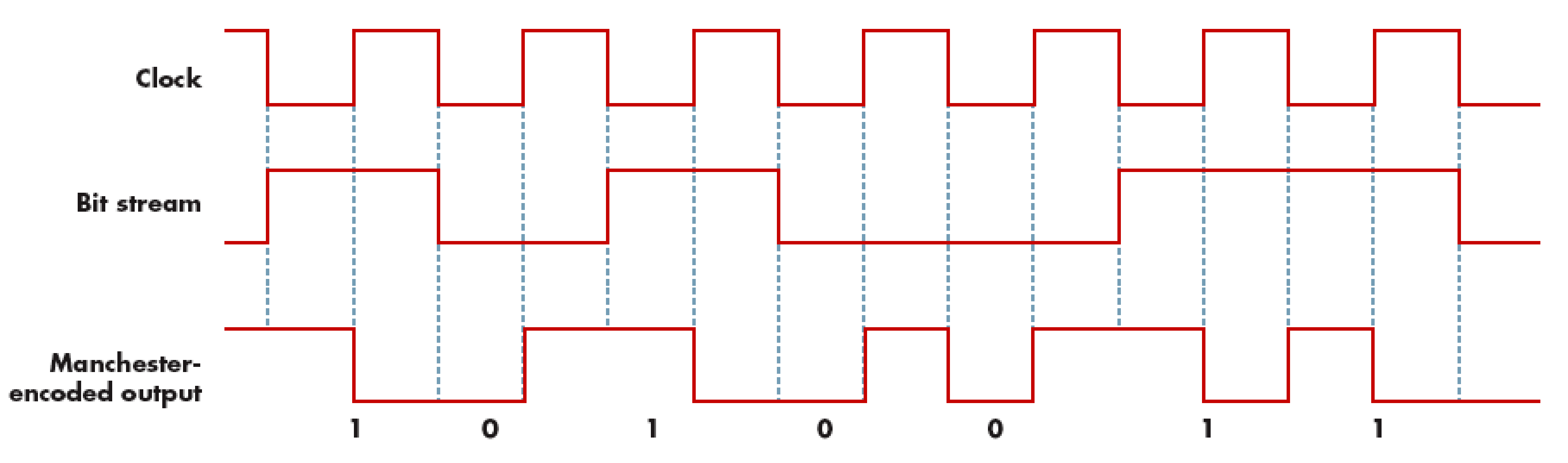
Decoding the signal

Demo

- Common encodings in RFID system
    - NRZ
    - Manchester
    - Unipolar RZ
    - DBP
    - Miller
    - Modified Miller
    - Differential
    - Puls-Pause

Manchester Encoding

- Bit Stream = Clock ^ Manchester Value
- 0 = low-to-high transition
- 1 = high-to-low transition



Decoding the signal

Demo

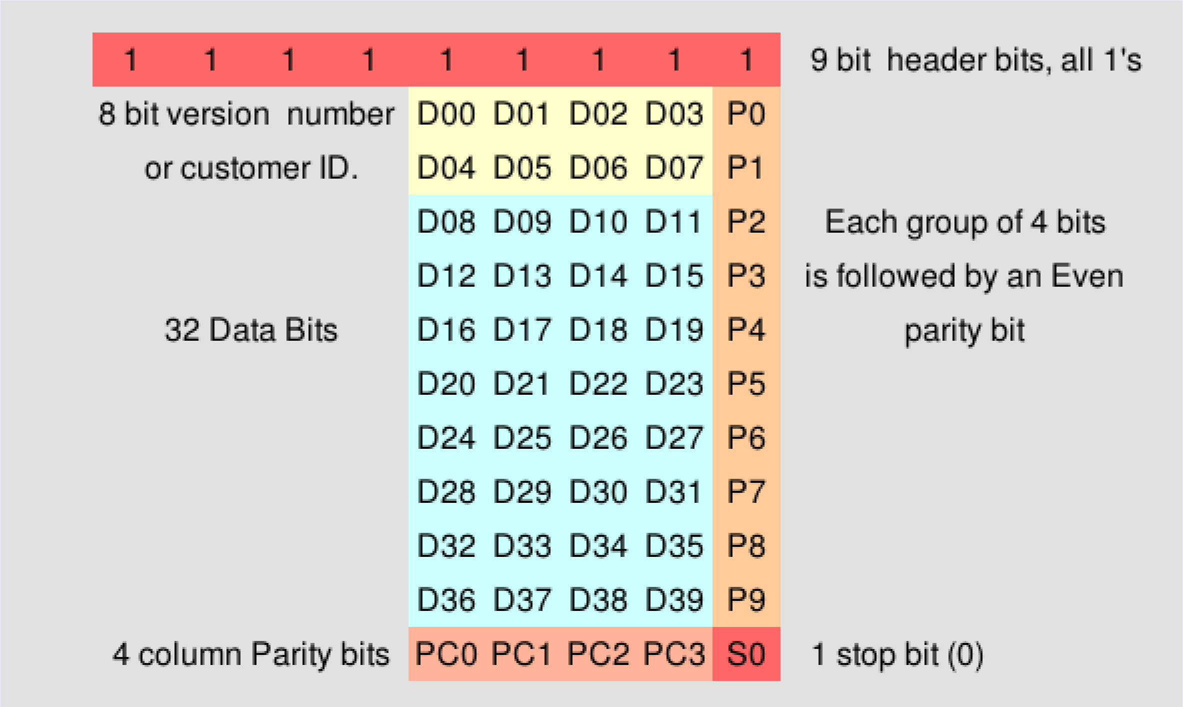
Analysis

- 64 Bit
- Remember: 10 digit number (HEX?) printed on top
- 9 * 1 = Header?
- 10 * 4 Bit = 40 Bit
- + Checksums/Parity bits??

Binary	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Demo

EM4100 protocol



1 1 1 1	1	1	1	1	1	9 bit header bits, all 1's
8 bit version number	D00	D01	D02	D03	P0	
or customer ID.	D04	D05	D06	D07	P1	
	D08	D09	D10	D11	P2	Each group of 4 bits
	D12	D13	D14	D15	P3	is followed by an Even
32 Data Bits	D16	D17	D18	D19	P4	parity bit
	D20	D21	D22	D23	P5	
	D24	D25	D26	D27	P6	
	D28	D29	D30	D31	P7	
	D32	D33	D34	D35	P8	
	D36	D37	D38	D39	P9	
4 column Parity bits	PC0	PC1	PC2	PC3	S0	1 stop bit (0)

- Decoded signal
  - – Seems to be the right decoding

⇒ Unfortunately, the tag id is not the one printed on top of the tag.

Cloning / replaying the data trace

- Using a T55x7 tag [Hid ProxCard II, EM4100 and Indala tags]
- Writing the “ID” to the T55x7 tag

⇒Tag can now be used to unlock the door

That’s it

NFC in Web Applications

- Currently not possible without external plugins
  - W3c draft: http://w3c.github.io/nfc/
  - Will be integrated in FirefoxOS v2.2 [https://developer.mozilla.org/en-US/docs/Web/API/NFC_API]
  - Supported in Chrome Apps

Security issues

- Common Web issues
- NFC related issues
  - Replay
  - Weak Crypto
  - Sniffing

Example

- NFC RacingApp