RFID & MIFARE Ultralight

Erik Poll
Digital Security
Radboud University Nijmegen

RFID tags

- RFID = Radio-Frequency IDentification
- RFID devices are called tags or transponders
- More powerful RFID tags can be called (contactless) smartcards

- Inductive coupling is used for
  - energy transfer to card
  - transmission of clock signal
  - data transfer

  - simple tags only support data transfer from the tag to reader

Various kinds of RFID tags

- Animal identification RFID tags (ISO 11784 & 11785)
  - only transmit permanently programmed id code
- Advanced transponders (ISO 14223)
  - have more data and support writing & write-protection
  - compatible with ISO 11785
- Contactless smartcards
  - close coupling: a few mm (ISO 10536)
  - proximity: less than 10 cm (ISO 14443)
  - vicinity: more than 10 cm (ISO 15693)

Many of these contactless smartcards are *not* very smart: memory cards instead of microprocessor cards

Various kinds of RFID tags

- Container identification (ISO 10374)
  - active – battery-operated - transponder
- Anti-theft systems (VDI 4470)
  - only one bit of information
- Item management (ISO 18000 + others)
  - ie rfid bar codes
  - GTAG (Global Tag), joined effort of EAN (European Article Numbering Association) and UCC (Universal Code Council)

pros & cons of contact vs contactless?

- pros contactless
  - easy of use
  - no wear & tear of contacts on card and terminal
    - less maintenance
    - less susceptible to vandalism
- cons contactless
  - easier to eavesdrop on communication?
    - terminal communication easier to eavesdrop than tag communication
  - communication possible without owner's consent
    - for replay or relay man-in-the-middle attacks
  - cheap tags have limited capabilities to provide security (eg amount of data, access control model, crypto)
    - illustration later using ov card

Anti-collision

Additional complexity of contactless cards:

- several cards may be activated by reader
- anti-collision protocol needed for terminal to select one card to talk to

Anti-collision protocol may leak information!

- eg test version of Dutch passport used a *fixed* number in the anti-collision protocol. Real one uses random number

Memory transponders

- read-only
  - communication one way only
- writable, no write-protection
  - 1 byte to 64 Kbyte, in fixed blocks, eg 16 bit, 4 byte,..
  - no protection on writing
- writable, some write-protection
  - password/key or more complicated authentication procedure
  - state machine operating system
  - possible offering segmented memory
    - each memory segment with its own key
  - important standard: MIFARE
  - possibly dual interface

Microprocessor transponders

- like normal smartcard, but (also) wireless
- but with a lot less power
  - ISO 14443 5 mW
  - GSM 11.11 50 mW
  - ISO 7816 300 mW

- MIFARE plus
  - contactless interface behaves as MIFARE memory card
  - contact interface is regular smartcard

RFID hacking

- eavesdrop on communication
- talk to real tag or terminal
- then
  - hack a real RFID tag
  - use a blank one
    - but MIFARE serial number cannot be written
  - simulate the tag
    - new mobile phones with NFC (Near Field Communication) can be used to simulate RFID tag or RFID reader

MIFARE

- widely used proprietary standard by NXP (formerly Philips)
- closely related to – and basis for - ISO 14443
- several versions, incl.
  - MIFARE Ultralight, provides only memory with some write restrictions (locking)
  - MIFARE standard 4k, also provides authentication and communication encryption by proprietary CRYPTO01 algorithm

Info available from eg
www.nxp.com/products/identification/mifare
mifare.net

Common MIFARE weaknesses

- **75%** of MIFARE RFID applications use default (transport) keys or keys used in examples in documentation

  [Source: Lukas Grunwald, DEFCON14, 2007]

  - *A0A1A2A3A4A5* is an initial transport key that many tags ship with. Googling for *A0A1A2A3A4A5* produces links to documentation with other example keys to try!

MIFARE Ultralight

- No keys to protect memory access
- Instead some read-only and write once memory for security
- Memory organised in 16 pages of 4 bytes
  - first part is read-only
    - includes 7 byte serial number
  - second part is One Time Programmable (OTP)
    - you can write 1's, not 0's
    - includes data for locking
  - third part is readable & writable

MIFARE Ultralight memory layout

Page	byte 0	byte 1	byte 2	byte 3
0	sn0	sn1	sn2	checksum
1	sn3	sn4	sn5	sn6
2	checksum	???	lock 0	lock1
3	OTP 0	OTP 1	OTP 2	OTP 3
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

read only

serial no

OTP

read/ write

application data

MIFARE ultralight memory access control

2 bytes for locking:

- 12 bits to lock data pages 4 .. 15 : L_i
- 1 bit to lock OTP area (page 3) : L_{opt}
- 3 bits to block locking of OTP, pp 4-9 and 10-15:

  BL_{OTP}, $BL_{4\text{-}9}$, $BL_{10\text{-}15}$

All these bites are OTP

L_7	L_6	L_5	L_4	L_{OTP}	$BL_{10\text{-}15}$	$BL_{4\text{-}9}$	BL_{OTP}
L_{15}	L_{14}	L_{13}	L_{12}	L_{11}	L_{10}	L_9	L_8

OV card

- MIFARE Ultralight for disposable tickets
- lock bytes initially 0x00F0, locking pages 12-15
  - data in pages 12-15 can still be read
- lock bytes set to 0xF8FF to invalidate card
- two bytes of the OTP used as counter
  - in unary style, eg 1111 1111 1111 1110 means one ride left
- pages 4-7 and 8-11 used to record last two transactions
  - meaning of certain bits clear
    - 000=purchase, 001=check in, 010=check out, 110=transfer
- pages 12-15 used for unknown card-specific data

[Source "Security Evaluation of the disposable OV chipkaart", by UvA students Pieter Siekerman and Maurits van der Schee , July 2007]

flaw 1

- lock bytes initially 0x00F0, set to 0xF8FF to invalidate tag
- we can change an invalid tag so that some terminals fail to recognize it as invalid; can you guess the flaw?
- remaining 3 lock bits can still be set to one, so that lock bytes become 0xFFFF
- flaw in some terminals: tags with lock bytes 0xF8FF are recognized as invalid, but tags with 0xFFFF are not
  - *Can you guess the terminal code that causes this?*

- enables easy experiments with "invalid" cards

flaw 2

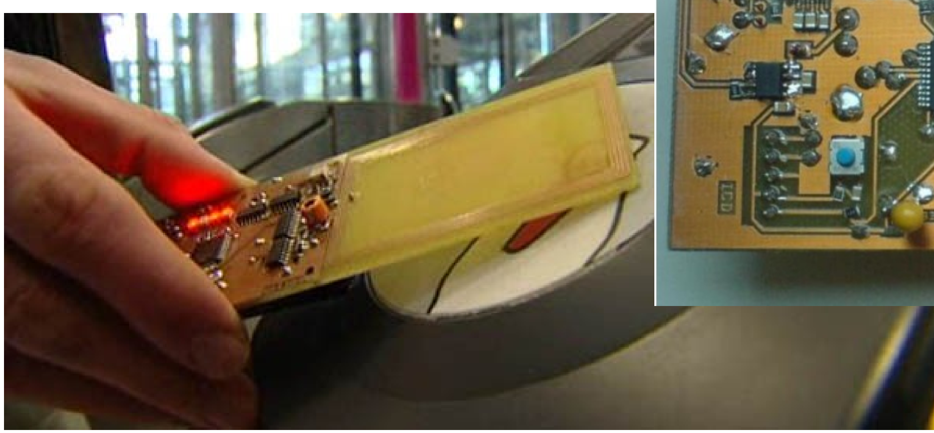
- on check-in, counter is incremented and transaction info written to pages 4-7
- on check-out, transaction info written to pages 9-11
- can you guess how a ticket could be used for multiple check-outs?
- by rewriting the transaction info (which is not write protected), we can use the same card to check-out again
  - *How could you prevent this flaw?*

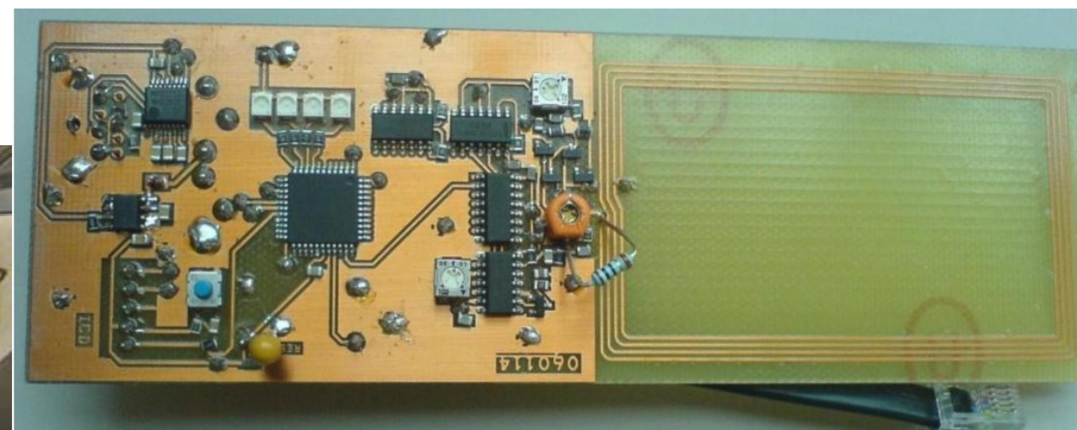
flaw 3

- More serious, and reportedly fixed
- Attack found
  - purchace single/multiple ride ticket
  - back-up data in page 4-11 (incl. purchase transaction)
  - use card, checking in and checking out
  - rewrite content of page 4-11, overwriting check-in and check-out transactions with purchase transaction
  - card can now be used again, but OTP counter is not increased: infinite number of free rides
- Cause?? Counter not checked & increased if purchase transaction is found in memory?

flaw 4

- What's a more fundamental weakness of these MIFARE Ultralight tags?
- What if we can simulate a tag?
- *No* way to protect against spoofing of tags.





Ghost device by Roel Verdult

MIFARE Classic

- Relies on proprietary CRYPTO1 algorithm
- Approx. 1 billion Mifare Classics in use
  - London Oyster Card, Dutch ov-chipkaart, Dutch government building access, Radboud university access
- CRYPTO1 partially reverse engineered by Plotz, Nohl and Starbug at CCC in Dec'2007
  - optical reverse engineering
- CRYPTO1 completely reverse engineered & *broken* by MIFARE team of Radboud University, March 2008
  - logical reverse engineering

[Garcia et al., Dismantling MIFARE Classic, ESORICS'2008]

Started as MSc projects of Roel Verdult and Gerhard de Koning Gans

More info on http://www.ru.nl/ds/research/rfid/

google for MIFARE & youtube

More RFID problems

- Vulnerabilities in first-generation RFID-enabled credit cards. by Heydt-Benjamin et. al. Tom O'Hare, Financial Cryptography'2007.
- http://prisms.cs.umass.edu/~kevinfu/papers/RFID-CC-manuscript.pdf