MIFARE—
Little Security, despite Obscurity
Motivation

- Most security systems use cryptography
  - Too many use proprietary ciphers
  - Many are weak, but secret
- We find cipher implementations from silicon
  - Cheap approach, no crypto knowledge required
  - We want to enable you to do the same

“No more weak ciphers. No more paranoia.”
Sean O’Neil
Motivating example: RFID
RFID tags

- Radio Frequency IDentification
- Tiny computer chips
- Passively Powered
RFID Applications

- RFIDs are becoming ubiquitous
- Integrated in many security applications
  - Payment, Access Control
  - Passports, Car Ignition
  - Implants, ...

RFIDs will be universal identifier. Might replace passwords, PINs, and fingerprints.
RFID-Crypto Mismatch

- Mifare
- TU Graz ['05]
- Passports

- No Crypto
- ???
- AES
- RSA
Mifare Security

- NXP claimed:
  - “approved authentication”
  - “advanced security levels”
- Stream cipher
- 48 bit key

Car thefts
(source: hldi.org)
Our Project (Starbug, Henryk Plötz, me)

We reverse-engineered the Mifare crypto and evaluated its security
Reverse-Engineering
Chemically extract chips:
- Acetone
- Fuming nitric acid

Shortcut: buy blank chips!
Mifare Classic RFID tag
Polishing

- Embed chip in plastic
  - Downside: chip is tilted

- Automated polishing with machine
  –or–
  Manually with sand paper

- “On your kitchen table”
  - Starbug
**Imaging Chip**

- Simple optical microscope
  - 500x magnification
  - Camera 1 Mpixel
  - Costs < $1000, found in most labs

- Stitching images
  - Panorama software (hugin)
  - Each image ~100x100 μm

- Align different layers
Chip Layers

- Cover layer
- 3 Interconnection layer
- Logic layer
- Transistor layer
Logic Gates Library

- Chip has several thousand gates on logic layer
- But only ~70 different types
  - Detection can be automated through template matching
Logic Gates
Logic Gates – Inverter
Logic Gates – 2NOR
Logic Gates – 2NOR

Logic Gates Collection:
http://gates.nohlnet.de
Logic Gates Interconnect

- Connections across all layers

- Traced 1500 (!) connections manually
  - Tedious, time consuming
  - Error-prone (but errors easily spottable)
  - Tracing completely automated by now
Tracing Connections
Automated Tracing

Metal wire

Intra-layer via
Mifare Crypto-1

Challenge Response

RNG

Key stream

f(·)

48-bit LFSR

ID

Mutual authentication protocol

48-bit stream cipher
Vulnerabilities
Hardware: OpenPCD (+PICC)

a) Sniffing data
b) Full control over timing!
Random Number Generator

- 16(!)-bit random numbers
  - LFSR –based
  - Value determined by time of read

Our Attack:
- Control timing (OpenPCD)
  = control random number (works for tag and reader!)
  = break Mifare security :)
For Starters: Brute-Force

- Cipher complexity low
  - Was probably a primary design goal
  - Very efficient FPGA implementation

FPGA cluster finds key in 50 minutes!
30 sec. when trading space for time!!

Source: Pico Comp.
Filter function is a network of smaller functions that are statistically biased.

Adversary controls inputs, can probe for internal state bits.

Attack takes < 1 minute on laptop.
Algebraic Attacks

- Unpublished attacks that exploit simple feedback structure and statistical weaknesses
- Works for strong random numbers
  - Hence, even against Crypto-1 on Mifare Plus!

- Attack takes 30 seconds on laptop
- Stay tuned for publication ...
Mifare Classic Weaknesses

- Time Memory Trade Offs
- Key Probing
- Algebraic Attacks
- Weak Authentication Protocol
  - Weak Filter Function
  - 48-Bit Stream Cipher
  - Weak Random Number Generator
- Brute Force (due to small key)
- Replay Attacks (due to predictable random numbers)
## Attack Properties

<table>
<thead>
<tr>
<th></th>
<th>Runtime on FPGA Cluster ($100,000)</th>
<th>Runtime on PC</th>
<th>Works despite strong random numbers (Mifare Plus)</th>
<th>Hard to Detect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay Attacks</td>
<td>(0)</td>
<td>(0)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Brute Force</td>
<td>50 min</td>
<td>—</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time Memory Trade Offs</td>
<td>30 sec</td>
<td>—</td>
<td>No</td>
<td>Maybe</td>
</tr>
<tr>
<td>Trade Offs (TMTO)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key Probing</td>
<td>—</td>
<td>1 min</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Algebraic Attacks</td>
<td>—</td>
<td>30 sec</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Protection insufficient for almost all uses:
- Access control, car theft protection, credit cards, ...
- Perhaps good enough for privacy
Reverse-Engineering is possible
  • you should try! (I’ll help)
Obscurity add security only in the short-run
  • (but lack of peer-review hurts later)
RFID constraints make good crypto very hard
  • How much security is needed?
  • How much more expensive is privacy?
Questions?

Karsten Nohl
nohl@virginia.edu

Talk to me about your reverse-engineering ideas!