CSc 466/566

Computer Security

26 : Authentication I

Version: 2019/12/04 10:06:19

Department of Computer Science
University of Arizona

collberg@gmail.com

Christian Collberg

Outline

1. Introduction
2. Barcodes
3. Magnetic Stripe Cards
4. Smartcards
   - Smartcard Security Issues
   - Invasive attacks
   - Non-Invasive attacks
   - Countermeasures
5. Summary

Means of Authentication

We identify someone by a combination of

1. something they have — smart card, key fob
2. something they know — password, mother's maiden name, first pet's name
3. something they are — fingerprint, retina scan

- Here we'll look at: something physical you posses, or something you are (biometrics).

Outline

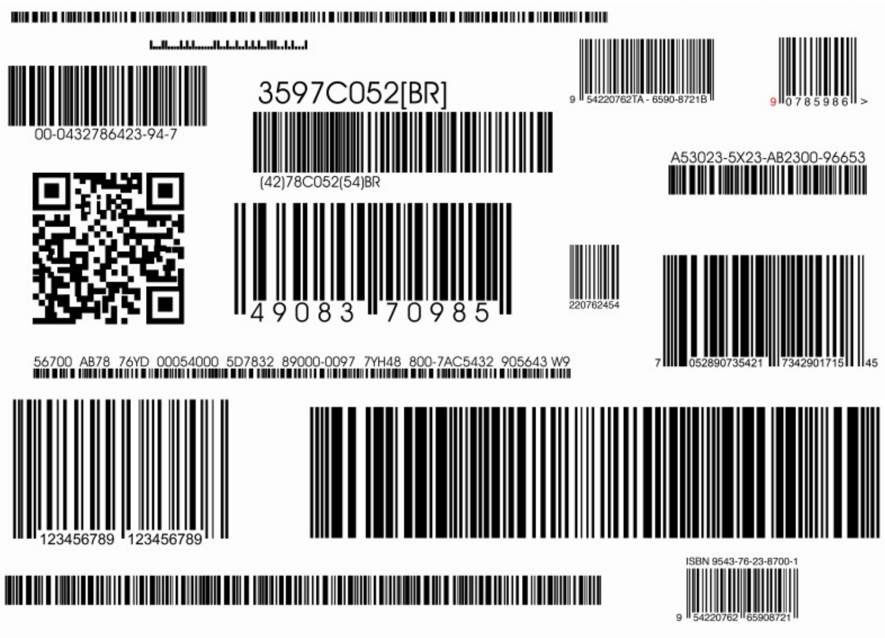
1. Introduction
2. Barcodes
3. Magnetic Stripe Cards
4. Smartcards
   - Smartcard Security Issues
   - Invasive attacks
   - Non-Invasive attacks
   - Countermeasures
5. Summary

Barcodes

- Uses for grocery checkout, postage, etc.
- Easy to duplicate.
- On boarding passes:
  - Barcode holds internal unique identifier;
  - Hard to forge, since only airline knows ID → passenger mapping.

49083 70985

Barcodes



Exercise: Goodrich & Tamassia C-2.12

- Airlines are given a a no-fly list of names.
- Consider the following security measures for airline travel:
  1. Before entering the departure area of the airport, passengers go through a security check where they must present ID and boarding pass.
  2. Before boarding a fight, passengers must present a boarding pass, which is scanned to verify the reservation.
- Show how someone who is on the no-fly list can manage to fly provided boarding passes can be printed online.

✎

✎

Exercise: Goodrich & Tamassia C-2.12

- Which additional security measures should be implemented in order to eliminate this vulnerability?

Exercise: Goodrich & Tamassia C-2.12. . .

✎

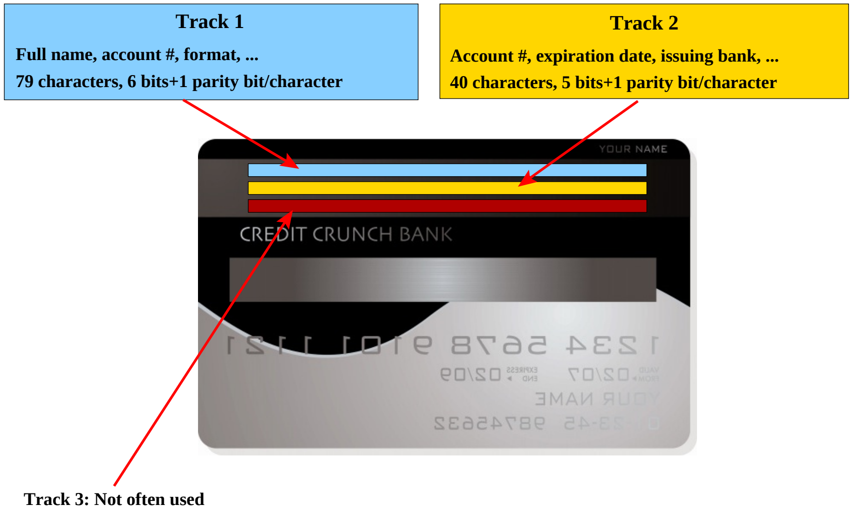
Outline

1. Introduction
2. Barcodes
3. Magnetic Stripe Cards
4. Smartcards
   - Smartcard Security Issues
   - Invasive attacks
   - Non-Invasive attacks
   - Countermeasures
5. Summary

Magnetic Stripe Cards

- Developed in the late 60s.
- Debit/credit cards, drivers' licenses, ID cards
- Three tracks, error correcting code (parity bit) to deal with worn magnetic stripes.

Magnetic Stripe Cards



Track 1

Full name, account #, format, ...

79 characters, 6 bits+1 parity bit/character

Track 2

Account #, expiration date, issuing bank, ...

40 characters, 5 bits+1 parity bit/character

Track 3: Not often used

Magnetic Stripe Cards: Vulnerabilities

- Easy to read.
- Easy to reproduce.
- Some vendors use the card as a stored value card, storing money, points, transportation credits, etc. — cloning attack.

Magnetic Stripe Cards: Countermeasures

1. Embed hologram in the card.
2. Customer signature.
3. PIN code.
4. Secret data formats (security-through-obscurity).
5. Cryptographic signature algorithms to validate data intergrity.

Exercise: Goodrich & Tamassia C-2.11

- A bank wants to store the account number of its customers (an 8-digit number) in encrypted form on magnetic stripe ATM cards.
- We assume the account number is supposed to be secret.
- We assume the attacker can read the magnetic stripe.
- How secure are the methods on the following slides?

Exercise: Goodrich & Tamassia C-2.11...

1. Store a cryptographic hash of the account number:
   ✎

Exercise: Goodrich & Tamassia C-2.11...

2. Store the ciphertext of the account number encrypted with the bank's public key:
   ✎

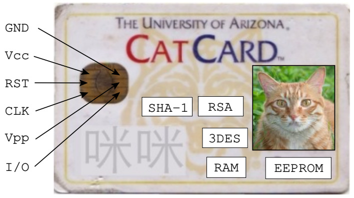
Exercise: Goodrich & Tamassia C-2.11...

3. Store the ciphertext of the account number encrypted with the bank's secret key using a symmetric cryptosystem:
   ✎

Outline

1. Introduction
2. Barcodes
3. Magnetic Stripe Cards
4. Smartcards
   - Smartcard Security Issues
   - Invasive attacks
   - Non-Invasive attacks
   - Countermeasures
5. Summary

Smartcards

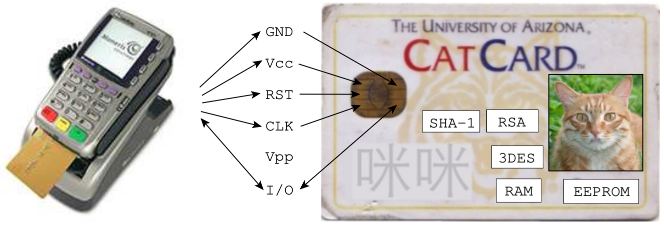


- Mass transit, prepaid phone cards, identification cards, *SIM cards*, pay-TV set-top boxes, credit cards, building access cards, electronic wallet, passports
- Chip-and-pin: credit cards with smart card technology.

Smartcards for security

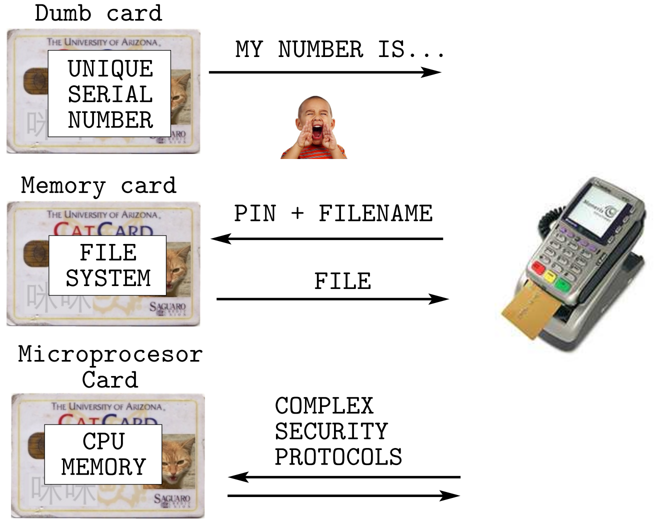
- Security problems with using cryptography:
  1. where do we store cryptographic keys?
  2. who do we trust to use cryptographic keys?
  3. generating, distributing, revoking keys
- Smartcards solve 1 and 2.
- Smartcard and help with 3: distributing keys to users.
- Disk encryption: smart card stores the key.

Smartcard Interface



- Power and clock from Card Acceptance Device (CAD).
- CAD talks to the card over a 1-bit serial link.
- `Vpp`: High power to EEPROM. No longer used.

Types of Smartcards



Dumb card

UNIQUE SERIAL NUMBER

MY NUMBER IS...

Memory card

FILE SYSTEM

PIN + FILENAME

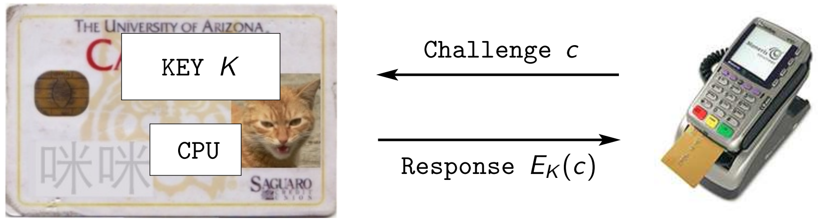
FILE

Microprocesor Card

CPU MEMORY

COMPLEX SECURITY PROTOCOLS

Challenge Response



KEY K

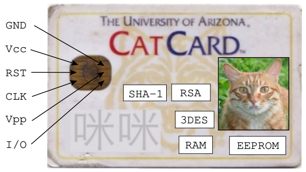
CPU

Challenge c

Response $E_K(c)$

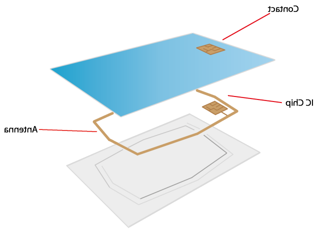
- Cards are often used for authentication, using a challenge-response protocol.
- The key never leaves the card.

Tamper-resistance and Cryptography



- Protected memory in which a secret can be stored.
- Trade-off between tamper-resistance and cost.
- Cryptographic capabilities: generate and store public-key key-pairs, RSA encryption, SHA-1 hashing.

Contactless Smartcards



- Newer card types are contactless.
- They get power from the CAD through an antenna.

Example: Gemalto TOP DM GX4

- JavaCard virtual machine interpreter, 68KB of persistent RAM, 78KB EEPROM, 3DES/AES/RSA encryption, SHA-1 cryptographic hash, and asymmetric key pair generation.
- JavaCard specifies a subset of the Java language and standard libraries designed specifically for smart card programming, along with a virtual machine instruction set optimized for size.

Example: Gemalto TOP DM GX4...

> *the platform implements most advance security countermeasures enforcing protection of all sensitive data and function in the card. ... includes multiple hardware and software countermeasure against various attacks:*
>
> - *Side channel attacks*
> - *Invasive attacks*
> - *Advanced fault attacks*
> - *Other types of attack.*

Exercises

1. What is a dumb smart card?
   ✎
2. What is a memory smart card?
   ✎
3. What is a microprocessor smart card?
   ✎

Smartcard Security Issues

- Flaws in programming the card.
- Non-invasive (side-channel) attacks.
- Invasive attacks.

Classic Implementation Flaws

Mistake 1: Homegrown Cryptography

- Proprietary algorithms are routinely broken.
- Use standard algorithms: RSA, 3DES, AES, SHA2, . . . !

Classic Implementation Flaws. . .

Mistake 2: Flawed Key Management

- 75% of systems using MIFARE RFID tags
  - use the default key (A0A1A2A3A4A5)
  - use keys found in the documentation

  (Lukas Grunwald)

Classic Implementation Flaws. . .

Mistake 3: Flawed Protocols

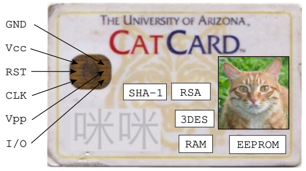
- Nonce used by MasterCard's EMV-CAP internet banking standard is always 0x0000000.
- Keep security protocols simple!

Classic Implementation Flaws. . .

Definition (Security Protocol)

A security protocol is a three line program people still manage to get wrong. (Roger Needham)

Invasive vs. non-invasive attacks



- Invasive attack:
  1. expose the bare chip,
  2. probe the surface to extract information
  3. poke the surface to modify the chip
- Non-invasive attack:
  - monitor execution characteristics (power, radiation, execution time) etc.
  - watch normal operations or induce faults

Invasive attacks

- An invasive attack destroys the card.
- You can use the secret code and data that you collect to clone a new card.
- Invasive attacks are useful when you know very little about the card.
- They may require sophisticated and expensive equipment.

Chipworks (http://www.chipworks.com)

Chipworks can extract analog or digital circuits from semiconductor devices and deliver detailed easy-to-understand schematics that document a single functional block or all the circuits. . . . We decapsulate the chip and analyze the die to locate the circuit blocks of interest. Then, using our Image Capture and Imaging System (ICIS) we generate mosiacs for each level of interconnect. Finally, advanced software and expertise is used to extract the circuits for analysis.

Step 1 — Depackaging

1. Remove the chip from the card itself by heating and bending it.
2. Remove the epoxy resin around the chip by dipping it in 60°C fuming nitric acid.
3. Clean the chip by washing it with acetone in an ultrasonic bath.
4. Mount the exposed chip in a test package and connect its pads to the pins of the package.

Step 2 — Deprocessing

5. Use an optical microscope to take large high-resolution pictures of the chip surface.
6. Identify major architectural features (ROM, ALU, EEPROM, etc.) and/or lower-level features such as busses and gates.
7. Remove the top metal track layer by dipping the chip in hydrofluoric acid in an ultrasonic bath.
8. Repeat from 5, for each layer.

Step 3 — Reverse Engineering

- Reverse engineer the chip
- Analyze the information collected
- Understand the functional units of the chip

Step 4 — Microprobing

9. To allow the probe contact with the chip, use a laser cutter mounted on the microscope to remove (patches of) the *passivation layer* that covers the top-layer aluminum interconnect lines.
10. Record the activity on a few of the bus lines (as many as you have probes) as you go through a transaction with the card.
11. Repeat from 10 until you've collected the bus activity trace from all of the bus lines.

Invasive attacks get harder...

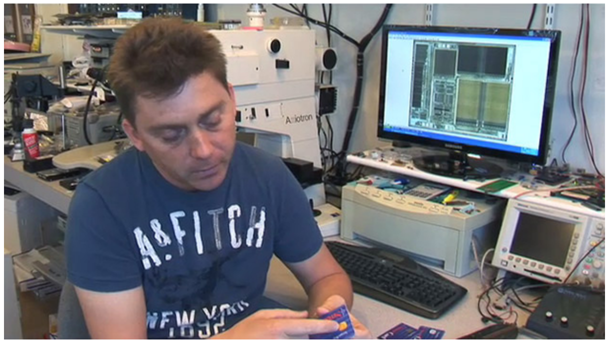
- Invasive attacks get harder over time as chip features get smaller.
- Rent a lab!
- Use your university lab!

Christopher Tarnovsky

Dish Network is accusing News Corp . . . of hiring hacker Christopher Tarnovsky to break into Dish's network, steal the security codes, and use them to make pirated cards to flood the black market. Tarnovsky admitted in court he was paid James Bond villain style, with $20,000 cash payments mailed from Canada hidden inside "electronic devices."

http://gizmodo.com/383753/news-corp-hires-hacker-to-break-into-dish-satellite-network-steal-security-codes-for-p

Christopher Tarnovsky



https://www.youtube.com/watch?v=tnY7UVyaFiQ

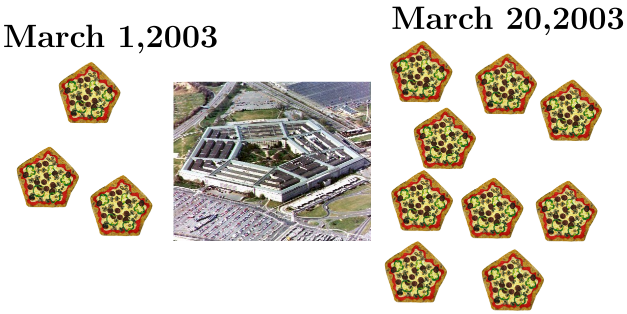
http://www.wired.com/politics/security/news/2008/05/tarnovsky?currentPage=all

Non-invasive attacks

Advantages over invasive attacks:

- No dangerous chemicals!
- Don't destroy the card!
- No expensive equipment!
- Once you have an effective attack against one particular card you can easily reuse it on another of the same model.

Side channel attacks



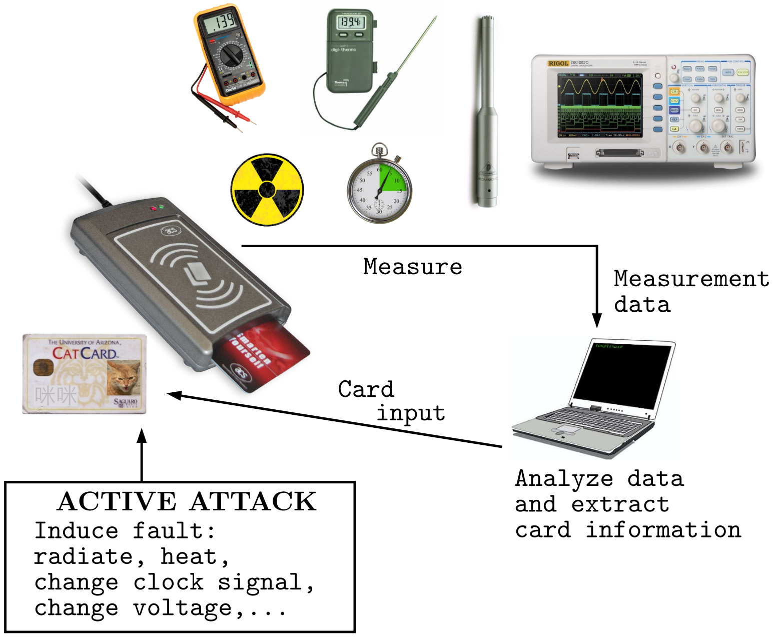
- When is the attack?
- Non-invasive attacks are side channel attacks.

Side channel attacks

Definition (Side channel attacks)

A side-channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms (compare cryptanalysis). For example, timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information, which can be exploited to break the system. (wikipedia)

Side channel attacks



Measure

Measurement data

Card input

Analyze data and extract card information

ACTIVE ATTACK

Induce fault:
radiate, heat,
change clock signal,
change voltage,...

Side channel attacks

- The clicks of a cipher machine in the Egyptian embassy was picked up with a microphone by the MI5: broke encryption!

Non-invasive attacks

- Passive attack:
  - Watch what comes out of the chip
  - . . . , electromagnetic radiation, power consumption, execution time, sounds, temperature
- Active attack:
  - Feed carefully constructed data/power/clock/. . . to the chip,
  - *then* measure the chip's behavior.

Exercises

1. What is a side channel attack?

2. What is an invasive attack?

3. What is a non-invasive attack?

Exercises

1. What is a passive non-invasive attack?

2. What is an active non-invasive attack?

Fault induction

- Methods:
  - generate a sharp voltage spike,
  - increase the clock frequency,
  - subject the chip to an electric field.
- Goal: Cause an error in the computation!
- Not every wrong instruction will cause an exploitable fault — use trial and error!
- Also known as glitching.

Glitch attack

```
void write(char* ptr, int length) {
   while (length > 0) {
      printf(*ptr); ptr++;
      length--;
   }
}
```

- Assume the `write()` function above is on the card and writes a region of memory to the I/O port.
- Can we induce a fault that will allow us to see all the memory on the card?

Glitch attack...

```
void write(char* ptr, int length) {
   while (length > 0) {
      printf(*ptr); ptr++;
      length--;
   }
}
```

- Goal: Force a fault in the boxed code, replacing it with any instruction that doesn't affect the `length` variable.
- Effect: The loop will cycle through all of memory, dumping it on the port!

Timing attacks

- Method:
  1. generate a large number of messages
     "please encrypt this file with your secret key";
  2. send them to the smart card;
  3. measure the time the operations take;
  4. deduce the key from the measurements.

Timing attacks...

- This modular exponentiation routine is used in many cryptographic operations, such as RSA encryption.
- `key` is the `w` bits long private key.

```
s[0] = 1;
for(k=0; k<w; k++) {
   if (key[k] == 1)
       R[k] = (s[k]*y) mod n;
   else
       R[k] = s[k];
   s[k+1] = R[k]*R[k] mod n
}
return R[w-1];
```

Timing attacks...

- The attacker wants to recover the `key`.
- Note that the two boxed codes take different time to run (multiplication is slow)!
- Note that the different boxed codes will run depending on the value of one bit of key!

```
for(k=0; k<w; k++) {
   if (key[k] == 1)
       R[k] = (s[k]*y) mod n;
   else
       R[k] = s[k];
}
return R[w-1];
```

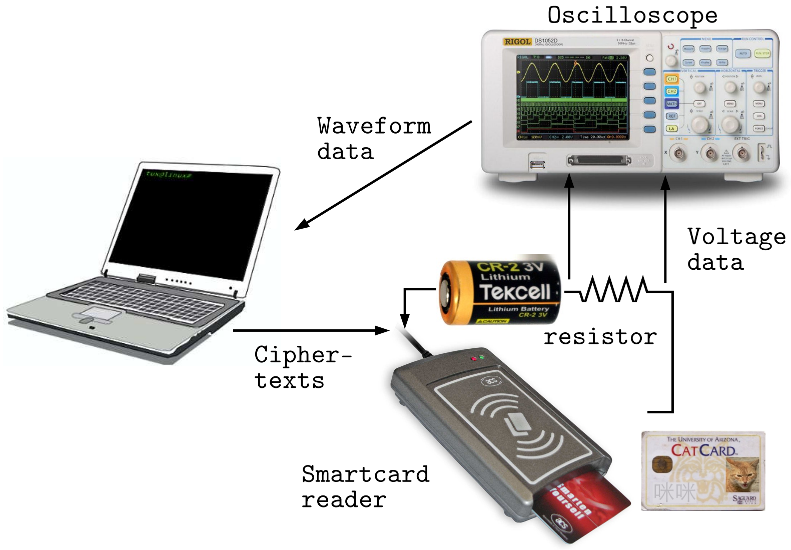
Timing attacks. . .

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.
  3. Ask the smart card to encrypt all the messages, and record their time.
  4. If the messages in M_1 take longer to encrypt than those in M_2, deduce that $x_1 = 1$, otherwise $x_1 = 0$.
  5. Knowing x_1 continue to deduce x_2 in the same manner.

Power analysis attacks

- Draw conclusions about the internal behavior of the chip from measurements of the power that it consumes.
- Different instructions consume different amounts of power
- Busses also draw power as bus lines change between 0 and 1: you can estimate the number of bits that changed on the bus by measuring the amount of power consumed.

Power analysis attacks. . .



Oscilloscope

Waveform data

Voltage data

resistor

Cipher-texts

Smartcard reader

Power analysis attacks. . .

- Easy attack to implement:
  1. Put a resistor on the chip's power supply line;
  2. Put a high-resolution high-sampling-frequency volt meter over the resistor;
  3. Use a computer to store and analyze the current traces.
- Counter noise in the measurements by averaging over a large number of transactions.

Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).

Countermeasures. . .

4. Environmental sensors: Detect if an attacker lowers the clock signal in order to be able to more easily monitor the computations.
5. Anticipate being attacked:
   1. Smartcards should be one part of a complete security architecture.
   2. Architect your systems to detect anomalies and to minimize losses.
   3. Design your system with upgradable security.

Exercises

1. What is a fault induction attack?
   ✎
2. What is a glitch attack?
   ✎
3. What is a challenge response protocol?
   ✎

Exercises

1. What is a timing attack?
   ✎
2. What is a power analysis attack?
   ✎

Exercises

1. What are four countermeasures to side-channel attacks on smart cards?
   ✎

Outline

1. Introduction
2. Barcodes
3. Magnetic Stripe Cards
4. Smartcards
   - Smartcard Security Issues
   - Invasive attacks
   - Non-Invasive attacks
   - Countermeasures
5. Summary

Readings and References

- Chapter 2 in *Introduction to Computer Security*, by Goodrich and Tamassia.

Acknowledgments

Material and exercises have also been collected from these sources:

1. Christian Collberg, Jasvir Nagra, *Surreptitious Software, Obfuscation, Watermarking, and Tamperproofing for Software Protection*, http://www.amazon.com/Surreptitious-Software-Obfuscation-Watermarking-Tamperproofing/dp/0321549252.
2. Tom Olzak, *Protect your network against fiber hacks*, http://www.techrepublic.com/blog/security/protect-your-network-against-fiber-hacks/222
3. Bruce Schneier, http://www.schneier.com/blog/archives/2007/09/eavesdropping_o_1.html
4. Erik Poll, *Embedded Software Security, ISSISP 2015*, http://www.cs.ru.nl/~erikpoll/talks/ISSIPS_erik_poll.pdf