Outline

Preamble

Security examples

What is computer security?

Structure of the course

Contact

- Dusko Pavlovic
- email: dusko@comlab.ox.ac.uk
- mailing list: oxford-security@googlegroups.com
- phone: 273 883
- office: 305
- office hours: after class or email for appointment

Announcements

- sign up sheets for **classes**
- timetable tweaks:
  - weeks 1–2: 3 lectures
  - week 4: no lectures

Announcements

- sign up sheets for **classes**
- timetable tweaks:
  - weeks 1–3: 3 lectures
  - week 4: no lectures
  - November 19: no lecture
Security 1: Introduction
Dusko Pavlovic

Preamble
Examples
Security?
Structure

Course

- What do we expect from the course?
- Why security?
- What is security?

Outline

Preamble

Security examples
- Securing resources: authorization
- Securing information: secrecy
- Securing information: authenticity
- Securing social interactions and networks

What is computer security?

Structure of the course

Securing resources: authorization

Digital Rights Management (DRM)

- art used to be bound to an artist
  - music was available only from a musician
  - a story from a storyteller
  - a painting could only be seen in one place
Securing resources: authorization

Digital Rights Management (DRM)

- mass reproduction bound art to copiable media
  - copying technologies led to copyright-based markets
  - artists could sell lots of books and records
  - Copyright Management: branding, celebrities

- digital networks freed art (science, religion…) from physical tokens (books, CDs…)
  - copying of digital content is essentially costless
  - Copyright Management becomes unviable
  - **Digital Rights Management**: seeks to
    * prevent (sandboxing, Vista…)
    * detect (watermarking …)
    * deter (lawyers …)
  - unauthorized copying of digital content

Securing information: secrecy

**Task: Fair deal of virtual cards**

Design a P2P application for mobile devices to deal virtual cards.

**Problem**

The players mistrust each other’s device. The dealing device must not see the cards that it is dealing.

**Hint**

Each device can encrypt messages, i.e. make them unreadable for others.
Securing social computation

Special case: Virtual coin flipping

Flip a virtual coin (without using a physical coin).

Variations: Millionaires’ Problem

Two millionaires need to truthfully find out which one is richer, without telling how rich they are.

Securing information: authenticity

Task

Spammers need lots of webmail accounts. They write bots who visit Hotmail, Yahoo! etc. to open disposable accounts, to distribute spam.

Design a protocol for setting up a webmail account which will be able to tell apart bots from humans.

First computer

First authentication protocol

Turing test

challenge
Agent Bot Smith in the Middle

Security 1: Introduction
Dusko Pavlovic
Preamble
Examples
Authorization
Secrecy
Authentication
Voting
Security?
Structure

Agent Bot Smith in the Middle

Problem
Smart card relay attacks

Agent Bot Smith in the Middle

Security 1: Introduction
Dusko Pavlovic
Preamble
Examples
Authorization
Secrecy
Authentication
Voting
Security?
Structure

Agent Bot Smith in the Middle

This becomes much easier with NFC phones!
Securing social interactions and networks

Task
There are 11 voters and 3 candidates A, B, and C. The voters need to elect one candidate. They have different preferences. Describe a method to elect the candidate which satisfies most voters.

Suppose the preferences are distributed as follows:

<table>
<thead>
<tr>
<th>voters</th>
<th>preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>A &gt; B &gt; C</td>
</tr>
<tr>
<td>2</td>
<td>A &gt; C &gt; B</td>
</tr>
<tr>
<td>2</td>
<td>B &gt; C &gt; A</td>
</tr>
<tr>
<td>4</td>
<td>C &gt; B &gt; A</td>
</tr>
</tbody>
</table>

If each voter casts 1 vote, then the tally is 5:4:2 for A > C > B.

If each voter casts 1+1 votes, then the tally is 9:8:5 for B > C > A.

Problem
Suppose the preferences are distributed as follows:

<table>
<thead>
<tr>
<th>voters</th>
<th>preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>A &gt; B &gt; C</td>
</tr>
<tr>
<td>2</td>
<td>A &gt; C &gt; B</td>
</tr>
<tr>
<td>2</td>
<td>B &gt; C &gt; A</td>
</tr>
<tr>
<td>4</td>
<td>C &gt; B &gt; A</td>
</tr>
</tbody>
</table>

- If each voter casts 1 vote, then the tally is 5:4:2 for A > C > B.
- If each voter casts 1+1 votes, then the tally is 9:8:5 for B > C > A.
- If each voter casts 2+1 votes, then the tally is 12:11:10 for C > B > A.
A computer performs computation:

- computation as **calculation**:
  - data processing through language, symbols, calculators...

- computation as **communication**:
  - data processing with other people, other computers, web...
What is a computer?

Examples of computers
- pocket calculator, brake stabilizer, flight controller
- laptop, desktop, mainframe
- Google cluster, StormWorm botnet
- the Web
- networks: cell, tissue, organism
- social groups and networks...

They all have their security requirements, vulnerabilities, attackers and adversaries.

Software engineering

Program dependability
- **safety**: "bad things (actions) don't happen"
- **liveness**: "good things (actions) do happen"

In sequential computation
- all first order constraints are dependability properties

Security engineering: Systems

Resource security (access control)
- **authorization**: "bad resource calls don't happen"
- **availability**: "good resource calls do happen"

In an operating or a computer system
- all resource constraints are security properties

Security engineering: Systems

Information security
- **secrecy**: "bad information flows don't happen"
- **authenticity**: "good information flows do happen"

In network computation
- all information flow constraints are security properties
**Security engineering: Networks**

Social choice (voting) and market economy

- **neutrality**: "bad data aggregations don’t happen"
- **fairness**: "good data aggregations do happen"

In social data processing

- all aggregation constraints are security properties

---

**Security vs dependability**

<table>
<thead>
<tr>
<th>processing</th>
<th>dependability</th>
<th>security</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>centralized</td>
<td>distributed</td>
</tr>
<tr>
<td>observations</td>
<td>global</td>
<td>local</td>
</tr>
<tr>
<td>Environment</td>
<td>neutral</td>
<td>adversarial</td>
</tr>
<tr>
<td>threats</td>
<td>accidents</td>
<td>attacks</td>
</tr>
</tbody>
</table>

---

**Security implementation**

Protection and enforcement counter attacks in three phases

- **prevention**: security properties cannot be breached  
  - firewalls, cryptography
- **detection**: security breaches are detected  
  - intrusion detection, digital forensics
- **policy**: recovery, penalties, incentives  
  - legal measures (RIAA, MPAA), economics of security  
    (cost of an attack must be higher than the expected profit of success)

---

**Outline**

- **Preamble**
- Security examples
- What is computer security?
- Structure of the course

---

**Structure of the course**

- Security
- Systems sec.  
  - Resource sec.  
    - Part 2
  - Cryptography  
    - Part 3
- Networks sec.  
  - Information sec.
  - Social sec.  
    - Part 7
- Protocols  
  - Part 4
- Web sec.  
  - Part 5
- Pervasive sec.  
  - Part 6