Using sensor data to generate random bit-strings

Patrik Jansson
Magnus Rundlöf

Supervisor: Christian Rohner
General

• Generate random bit-strings from sensor data for use in cryptographic applications.

• If we could generate the identical sequence in two places at the same time we would solve the key distribution problem.
Applications

• Applicable only locally, not for encrypting traffic over the Internet.

• For example, between PDA’s, mobile phones and digital cameras.
Set-up

• Embedded Sensor Board (ESB)
  – Contiki Operating System

• PC
Tilt sensor

- How does it work?
- How does Contiki handle the sensor?
Sub goals

- Compile applications for Contiki.
- Send sensor data from ESB to PC.
- Process sensor data on PC side.
- Shake two sensors simultaneously and see if the same bit-string is created.
Solutions / ESB

• Poll sensor every 500/1000ms

• The difference between two consecutive polls is sent to the PC via RS232 (serial port).
Solutions / PC

- Read sensor data from RS232 and present incoming values in a histogram in real-time.
- Extract bits and construct the bit-string.
- Perform randomness tests on the bit-string.
Solutions / bit-string

- Two approaches to generate a bit-string from sensor data
  - Sensor values from fixed intervals
  - Count consecutive intervals satisfying some condition
Sensor values from fixed intervals

Sensor value

Bit index=2

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Count consecutive intervals

Sensor value

n>s

T T T F F T T T T

3 4

Bit index=2

0 0 1 0

1 0 0
Randomness tests

• Frequency test
  – Compares the number of 0’s and 1’s.

• Serial test
  – Compares overlapping occurrences of 00, 01, 10, 11.

• Poker test
  – Compares non-overlapping occurrences of different bit sequences of given length.
Conclusions

• The precision of the sensor was not adequate, i.e. the values differ.

• It would be preferable to use sensors with higher resolution.

• Trade-offs
  – Speed vs randomness
• However if we had higher precision and better resolution it could probably be done.
Demo