Shake well before use: Authentication based on Accelerometer Data

Pervasive 2007
15. May 2007, 09:25

Rene Mayrhofer, Hans Gellersen
Lancaster University, UK

An idea

Pairing small mobile wireless devices

- Take two (or multiple) devices together in one hand
- Shake...

[HMSABG 2001] L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, H.-W. Gellersen: "*Smart-its friends: A technique for users to easily establish connections between smart artefacts*". In: Proc. UbiComp 2001, Springer-Verlag, 2001

The problem

Wireless communication is insecure

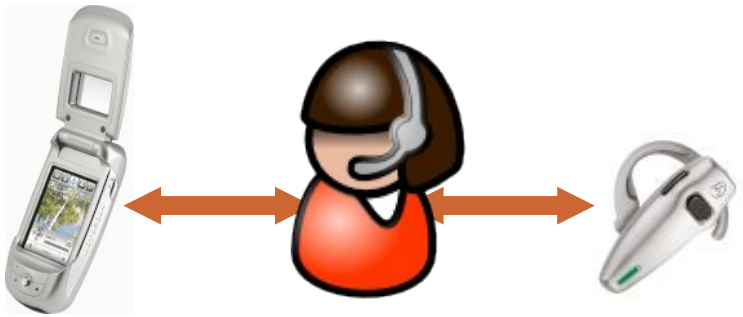
- Especially problematic for spontaneous interaction: no a priori information about communication partners available

⇒ User needs to establish shared secret between devices

Why is it a problem? (1)

Secret key exchange over wireless channels

- Can use Diffie-Hellman (DH) for key agreement
- Problem of Man-in-the-Middle (MITM) attacks:



⇒ Secret keys need to be authenticated

Why is it a problem? (2)

Options for authentication

- Entering PINs (e.g. Bluetooth), passwords (e.g. WEP/WPA)
- Verifying hashes of public keys (e.g. web site certificates)

Need user interface for that!

- A headset doesn't have a classical user interface (display + keypad)

Somebody needs to do it!

- Do you want to enter random password 10 – 100 times a day?

⇒ Problem of scalability

One possible approach

Solution for both problems: implicit context authentication

- Authenticate devices when they are in the same context
- Measure physical properties that are verifiable by the user

Proposed methods:

- Concept: Kindberg et al. “constrained channels”, Balfanz et al. “location-limited channels”, Hoepman “ephemeral pairing”
- Implementations: e.g. “Seeing-is-Believing”, “Loud and Clear”, “LoKey”, “Relate” spatial authentication
- Our approach combines service and physical device selection and authentication: “Shake well before use”

Shaking as shared context

Shaking is common movement

- both (all) devices will experience very similar movement patterns
- both (all) devices will experience very similar accelerations

⇒ not only use it as interaction technique, but also for generating keys

Acceleration is a local physical phenomenon

⇒ difficult for an attacker (MITM) to estimate or replicate

- Not used for identifying users, only as shared context!

Reasons for using shaking

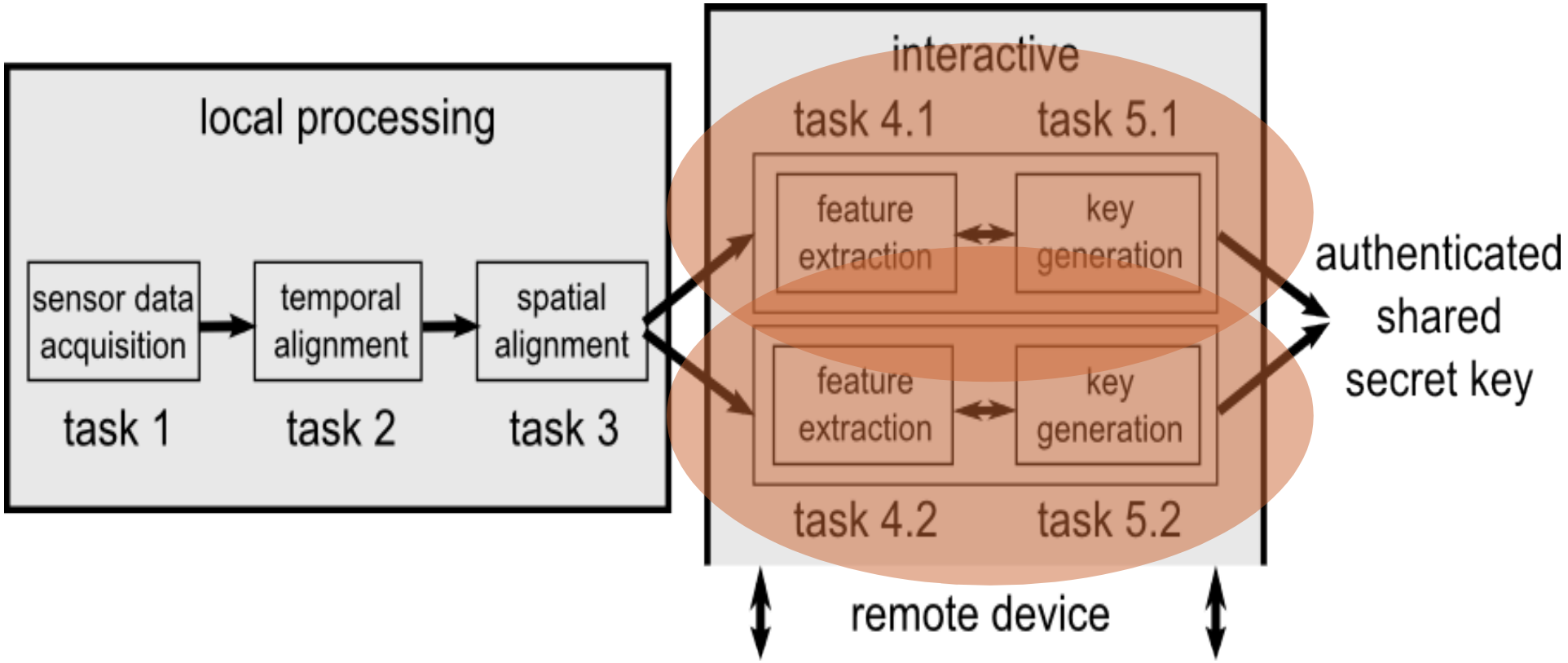
Shaking is

- intuitive
- vigorous
- varying

Accelerometers are

- small
- cheap
- (relatively) power-efficient

From sensor data to shared secret keys

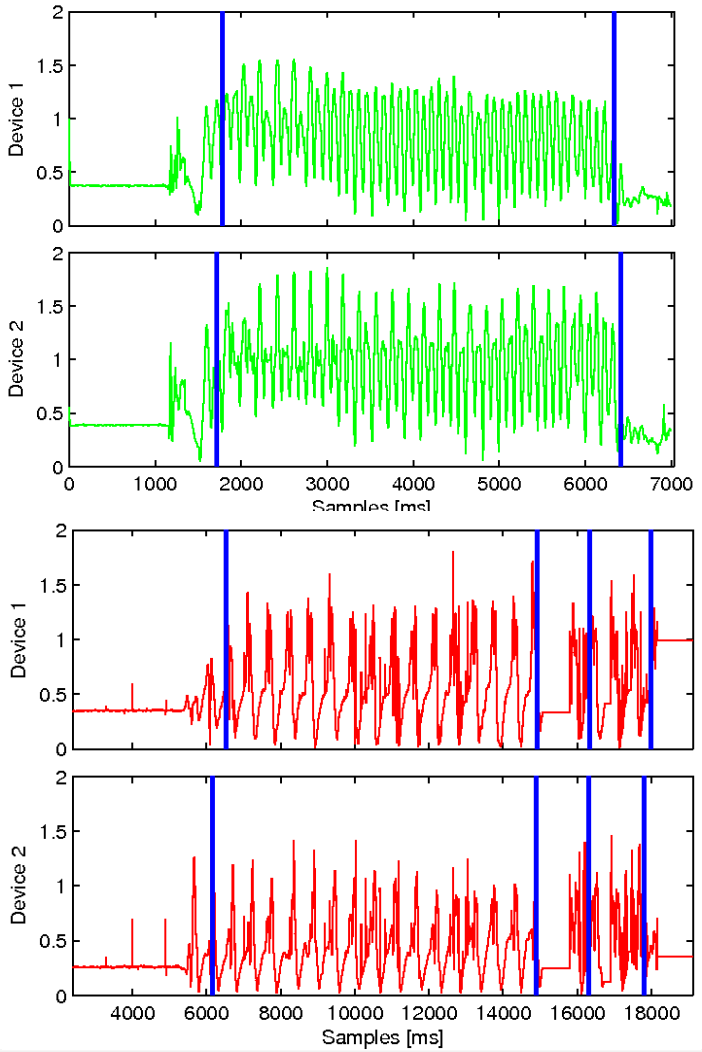


Pre-processing

1. Sensor data acquisition
   - Potential problem: side-channel attacks
2. Temporal alignment
   - Triggering
   - Synchronization

   ⇒ use motion detection
3. Spatial alignment
   - Devices arbitrarily aligned in 3D
   - Alignment changes when picked up (between “silent” and “active”)

   ⇒ reduce to 1 dimension (magnitude)



Feature extraction

Features for shaking:

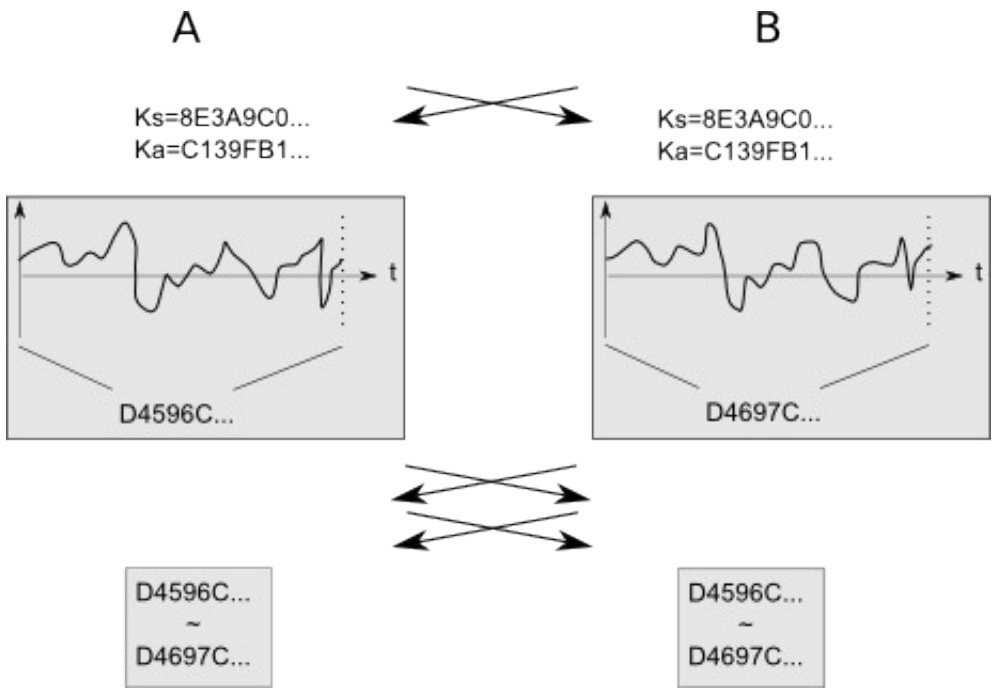
- Frequency domain
  - less accuracy required for synchronization
  - less sensitive to noise and alignment problems

- Coherence: measures power spectrum correlation between two signals split into overlapping slices, produces similarity value in [0; 1]
- Quantized FFT coefficients: pairwise added FFT coefficients quantized into exponential bands as feature vectors, compare equality

Cryptographic protocols (1)

Protocol 1:

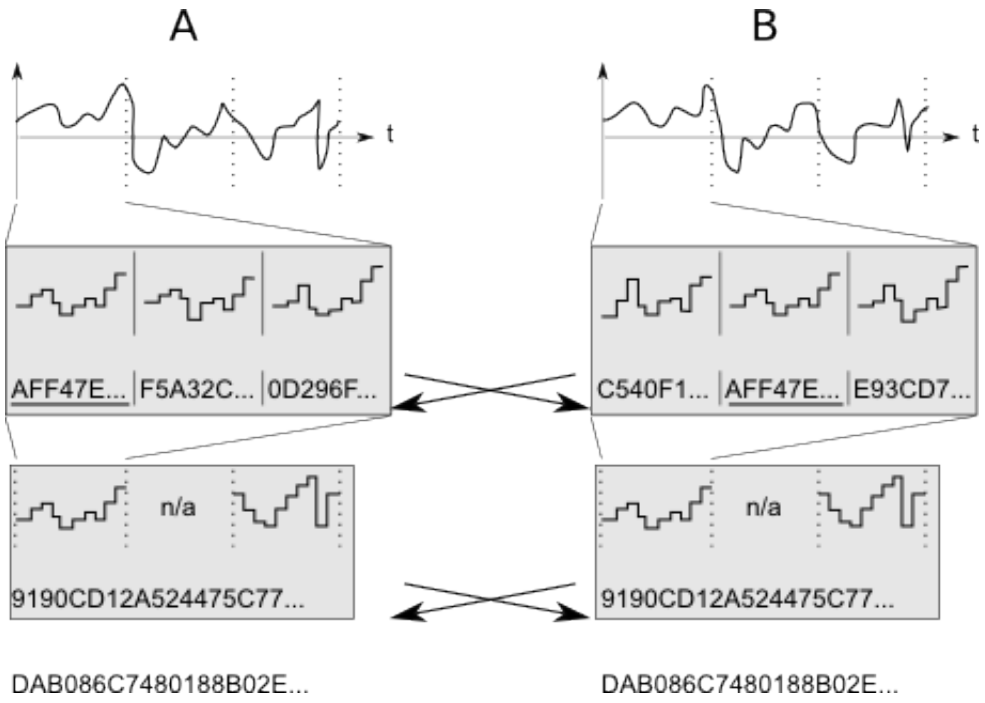
- Uses Diffie-Hellman for key agreement
- Exchange sensor time series after pre-processing with interlock* protocol
- Both devices check similarity locally with coherence



Cryptographic protocols (2)

Protocol 2:

- Generates secret shared keys directly from sensor data streams
- Computes feature vectors of quantized FFT coefficients
- Exchanges and compares hashes of feature vectors
  ⇒ candidate key parts
- Matching vectors concatenated
  ⇒ candidate keys



[May 2007b] R. Mayrhofer: "*The candidate key protocol for generating secret shared keys from similar sensor data streams*". In Proc. ESAS 2007: 4th European Workshop on Security and Privacy in Ad hoc and Sensor Networks. Springer-Verlag, July 2007

Protocol properties

Protocol 1

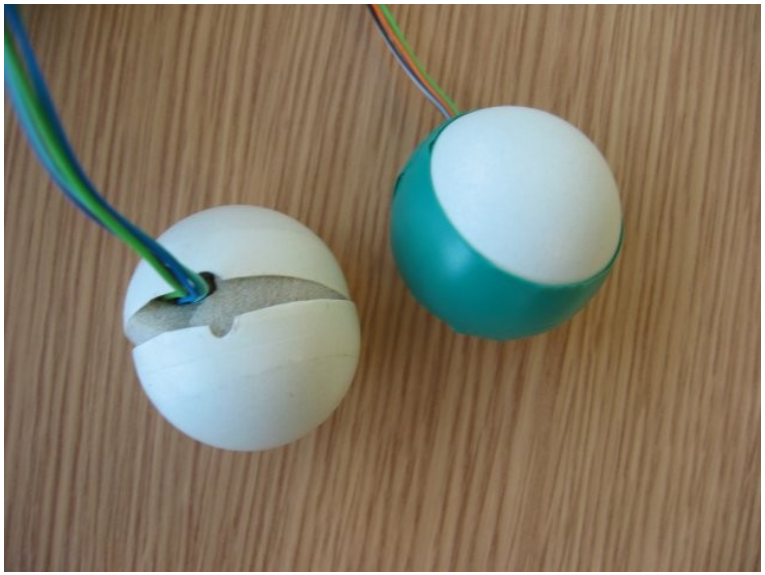
- Two phases:
  - Key agreement
  - Key verification
- Either with opportunistic key agreement or slight delay
- Only one-off chance for online attack
- Independent signal analysis

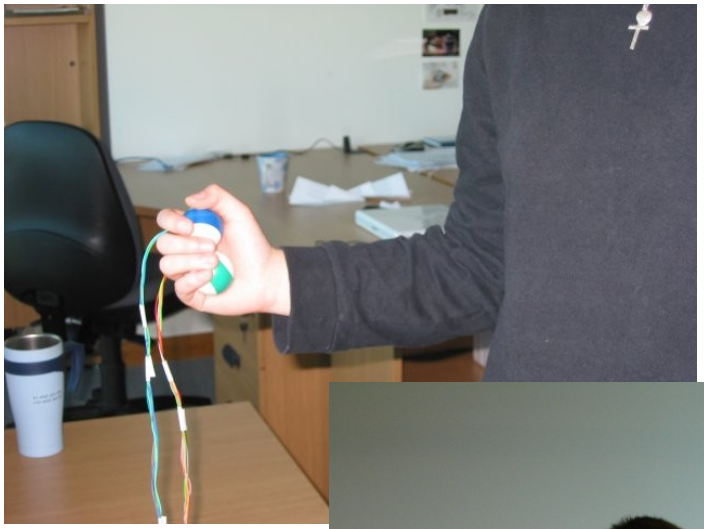
Protocol 2

- Single, continuous phase
- Devices "tune into" each other's key streams
- Multi-device authentication
- Offline lookup table attacks possible when feature vectors have insufficient entropy

In the beginning there were...

- 2x 2D accelerometer per “device”: ADXL202JE
- Sampled directly with the parallel port

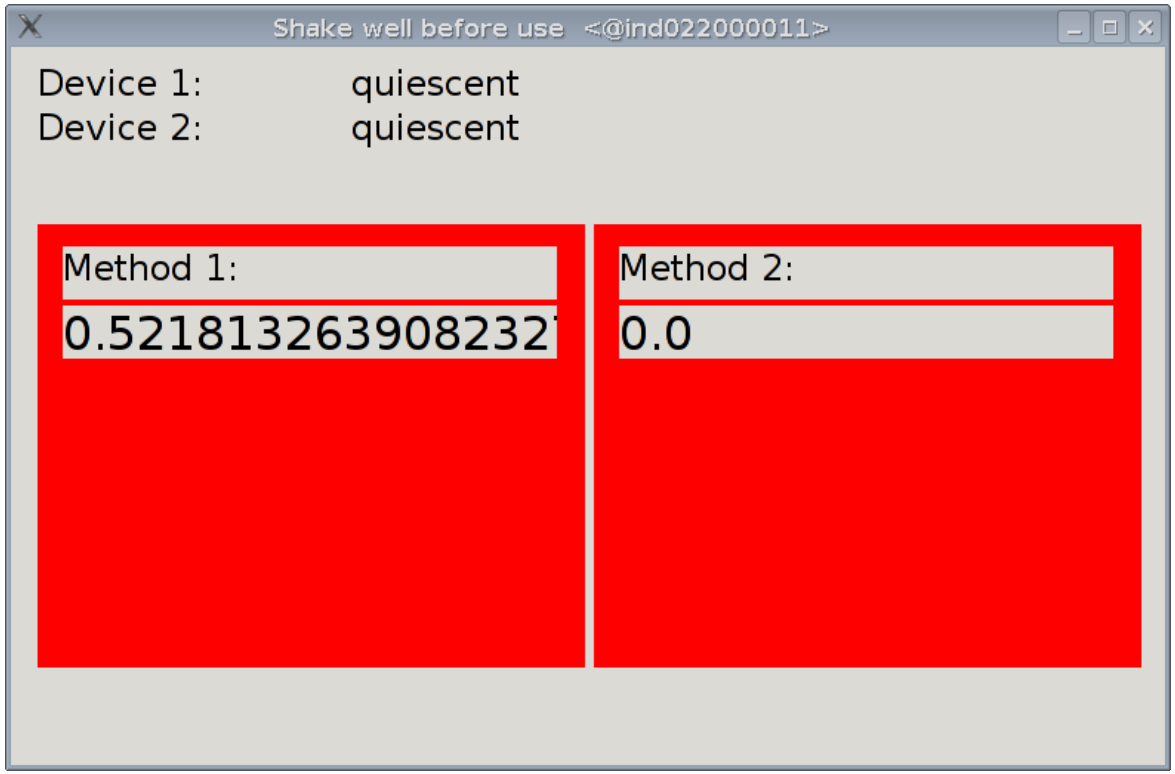




OpenUAT with a demonstrator

Current implementation of the method

- Sampling and logging with 600Hz
- "Real" implementation, but running on one host
- Java 1.2, compatible with J2ME
- Minimal GUI for live mode
- TCP channel for protocol 1
- UDP multicast for protocol 2



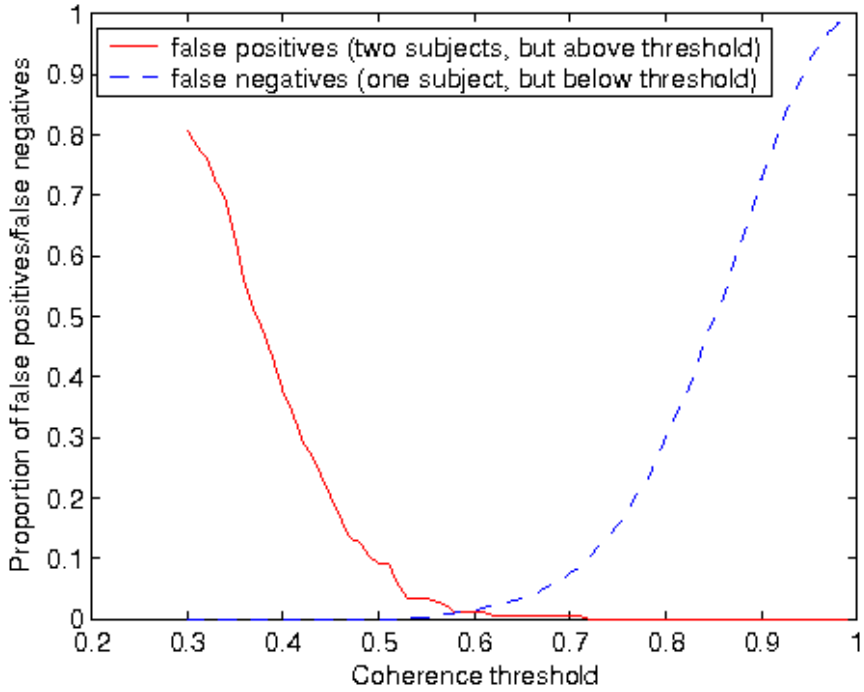
First experimental results

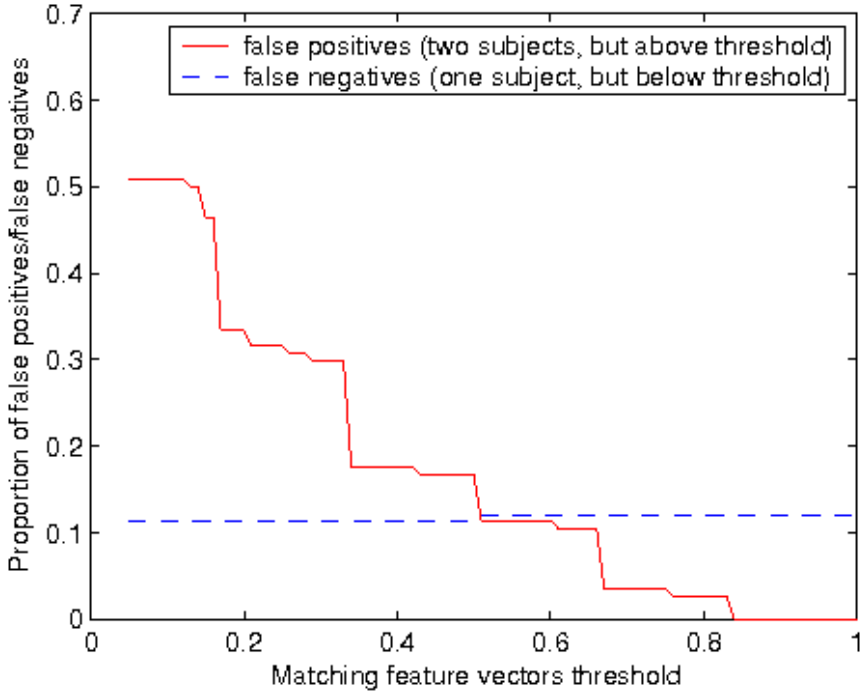
3 experiments:

- How do people shake?
- "Hacking" competition
- Live mode – does it work?

Results:

- Parameters for no false positives
- False negatives 10.24%, 11.96%
- 25/30 subjects successful

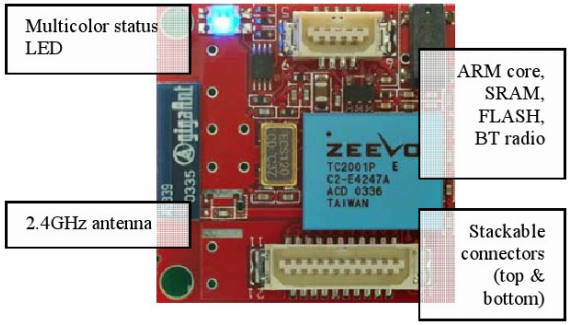




Scaling it down

Current developments:

- Implementing the method on embedded devices
  - "Nokia 5500 Sport" – includes 3D accelerometer with API
  - Intel iMote 1 with TinyOS – to emulate headset
- Bluetooth instead of TCP and UDP
  - different way of communication setup
  - no broadcast
- Improving classification accuracy



What have we done?

- Interaction method coupled with secure authentication
- Authentication without explicit user interface
- Implicit instead of explicit authentication

- Two protocols with significantly different properties
  - More secure, but heavyweight
  - More dynamic, multi-device

- Implementation under LGPL at http://www.openuat.org

“But what ... is it good for?”

Engineer at the Advanced Computing Systems Division of IBM, 1968, commenting on the microchip.

First International Workshop on Security for Spontaneous Interaction at Ubicomp 2007

Thank you for your attention!

Slides: http://www.mayrhofer.eu.org/presentations
Source code: http://www.openuat.org
Later questions: rene@mayrhofer.eu.org
Anonymized data sets available on request

OpenPGP key: 0xC3C24BDE
7FE4 0DB5 61EC C645 B2F1 C847 ABB4 8F0D C3C2 4BDE