Usability Analysis of Secure Pairing Methods

Ersin Uzun[12], Kristiina Karvonen[3], N. Asokan[23]

[1]University Of California, Irvine

[2]Nokia Research Center

[3]Helsinki University of Technology

Outline

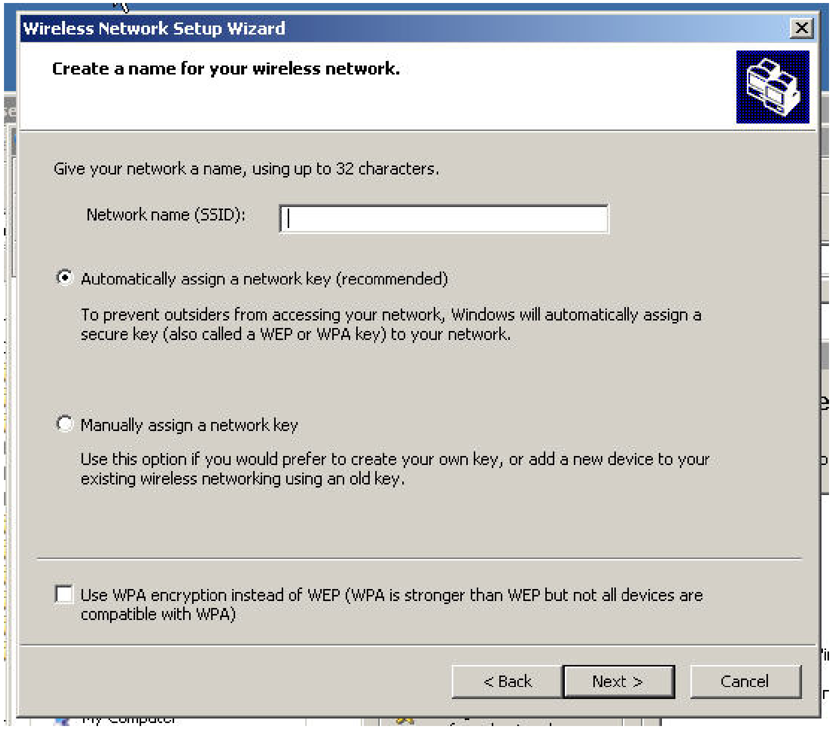
- What is secure pairing and why is it hard to secure?
- Current methods and ongoing efforts
- Usability study of different human mediated pairing methods.
- Conclusions and guidelines
- Discussion points
- Future work.

Secure pairing of personal devices

- **Pairing**: setting up the communication and security contexts for subsequent communication. E.g.,
  - Pairing a Bluetooth phone and headset
  - Enrolling a Phone or PC in the home WLAN
  - More instances to come: Wireless USB, WiMedia
- **Problem**: Secure pairing for personal devices
  - No prior context (no PKI, key servers etc.)
  - Ordinary non-expert users
  - Cost-sensitive commodity devices

(Uzun et al. USEC'07)

Current mechanisms are not intuitive



Wireless Network Setup Wizard

Create a name for your wireless network.

Give your network a name, using up to 32 characters.

Network name (SSID):

● Automatically assign a network key (recommended)

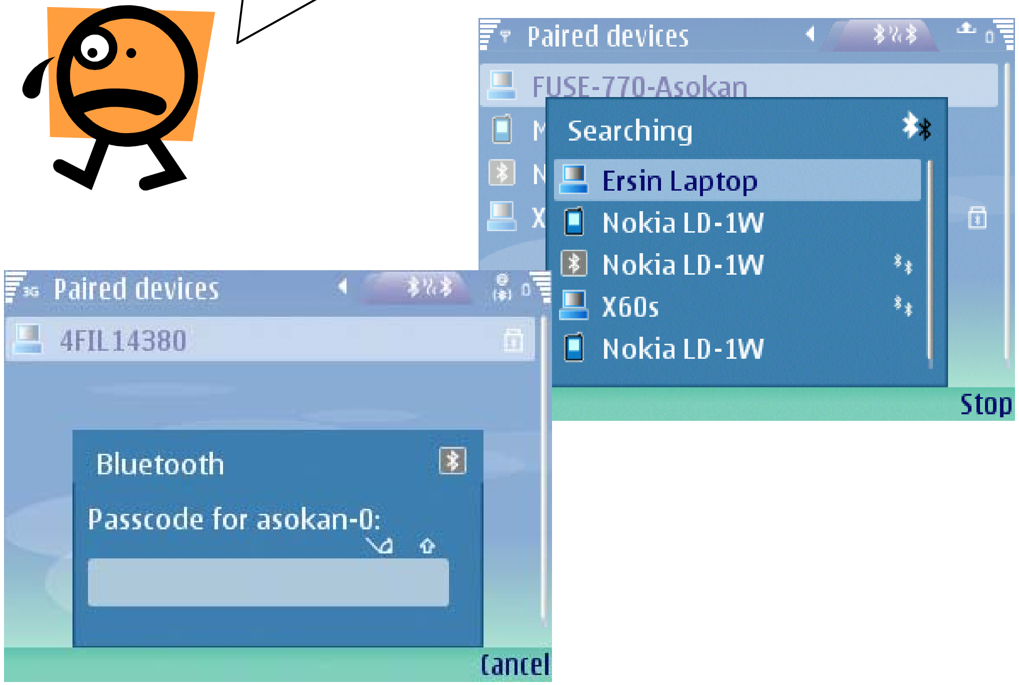
To prevent outsiders from accessing your network, Windows will automatically assign a secure key (also called a WEP or WPA key) to your network.

○ Manually assign a network key

Use this option if you would prefer to create your own key, or add a new device to your existing wireless networking using an old key.

□ Use WPA encryption instead of WEP (WPA is stronger than WEP but not all devices are compatible with WPA)

SSID? WPA?
Passcode!



Paired devices

FUSE-770-Asokan

Searching

Ersin Laptop

Nokia LD-1W

Nokia LD-1W

X60s

Nokia LD-1W

Paired devices

4FIL14380

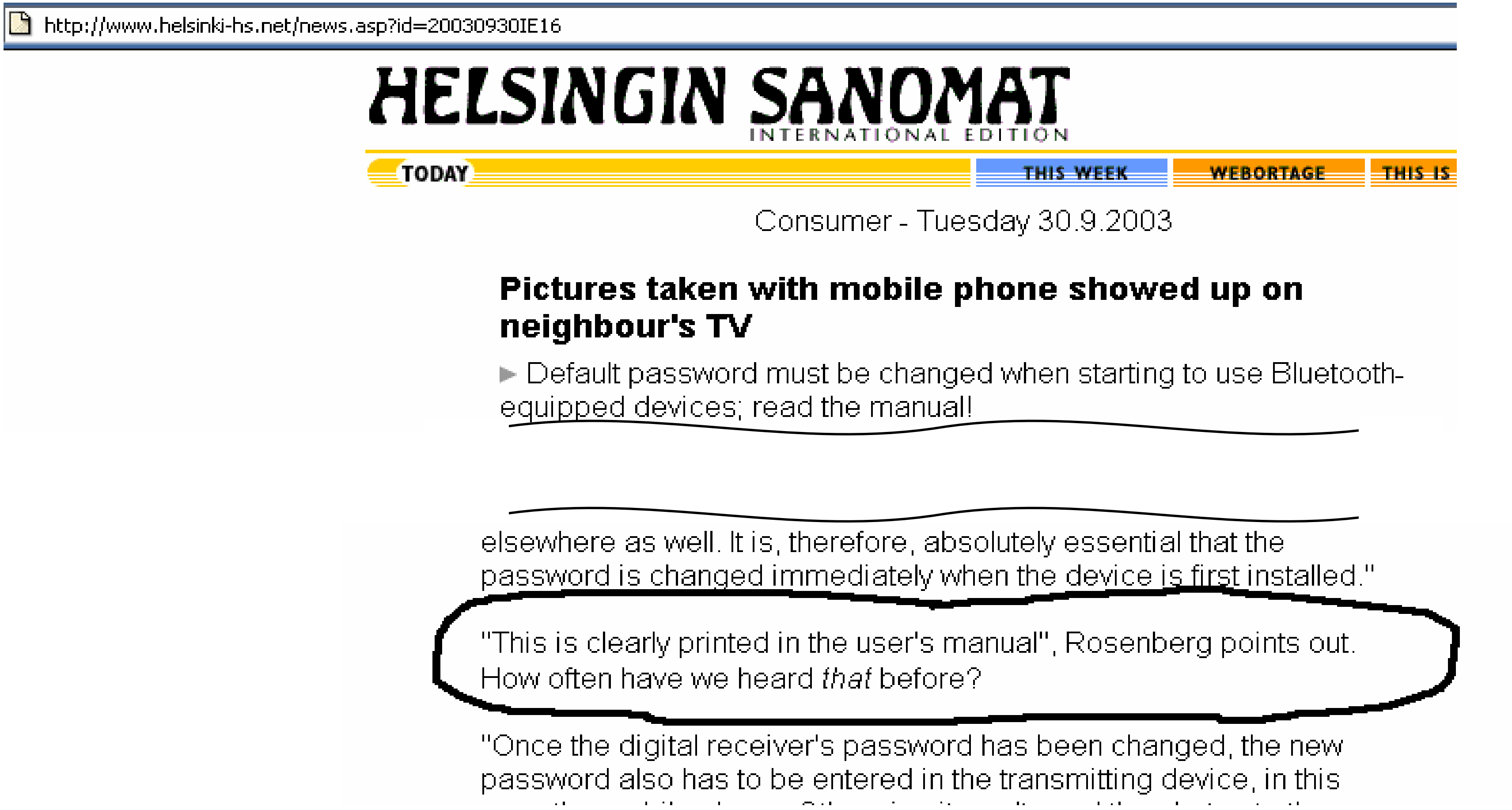
Bluetooth

Passcode for asokan-0:

... and not very secure!

(Uzun et al. USEC'07)

Naïve usability measures damage security



Consumer - Tuesday 30.9.2003

Pictures taken with mobile phone showed up on neighbour's TV

▶ Default password must be changed when starting to use Bluetooth-equipped devices; read the manual!

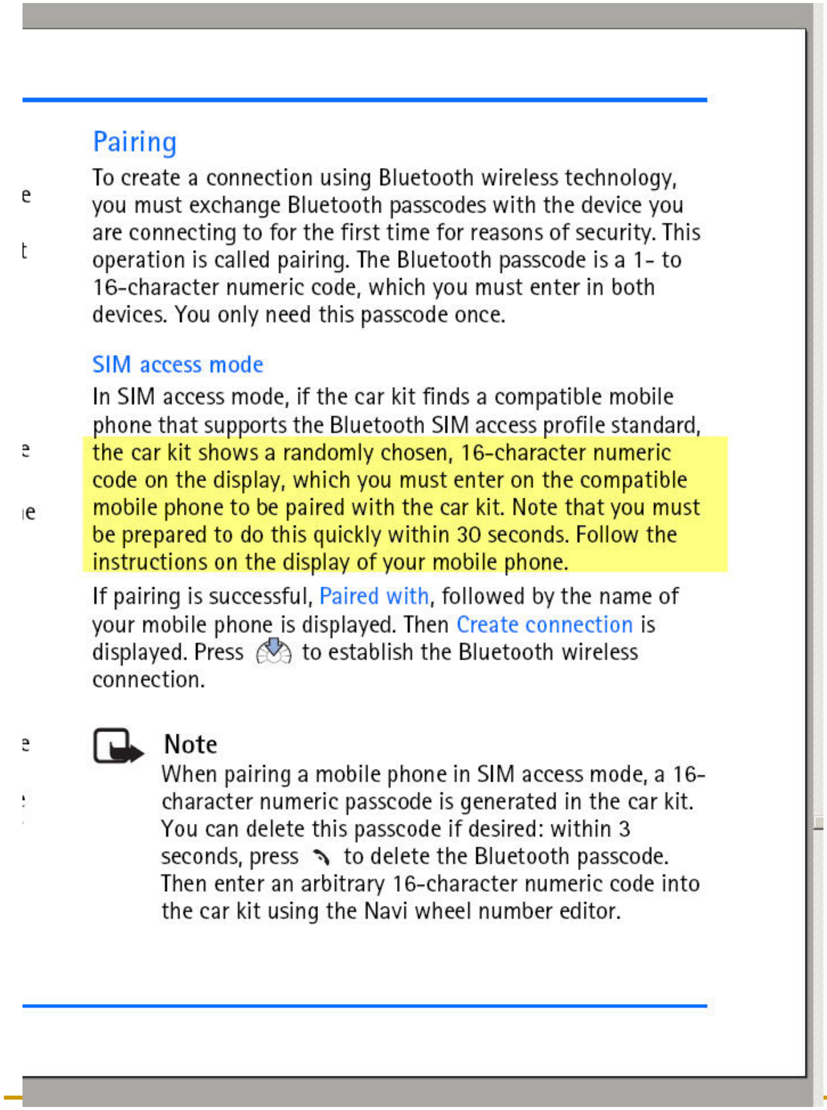
elsewhere as well. It is, therefore, absolutely essential that the password is changed immediately when the device is first installed."

"This is clearly printed in the user's manual", Rosenberg points out. How often have we heard *that* before?

"Once the digital receiver's password has been changed, the new password also has to be entered in the transmitting device, in this

(Uzun et al. USEC'07)

Naïve security measures damage usability



Pairing

To create a connection using Bluetooth wireless technology, you must exchange Bluetooth passcodes with the device you are connecting to for the first time for reasons of security. This operation is called pairing. The Bluetooth passcode is a 1- to 16-character numeric code, which you must enter in both devices. You only need this passcode once.

SIM access mode

In SIM access mode, if the car kit finds a compatible mobile phone that supports the Bluetooth SIM access profile standard, the car kit shows a randomly chosen, 16-character numeric code on the display, which you must enter on the compatible mobile phone to be paired with the car kit. Note that you must be prepared to do this quickly within 30 seconds. Follow the instructions on the display of your mobile phone.

If pairing is successful, Paired with, followed by the name of your mobile phone is displayed. Then Create connection is displayed. Press to establish the Bluetooth wireless connection.

Note
When pairing a mobile phone in SIM access mode, a 16-character numeric passcode is generated in the car kit. You can delete this passcode if desired: within 3 seconds, press to delete the Bluetooth passcode. Then enter an arbitrary 16-character numeric code into the car kit using the Navi wheel number editor.

- Bluetooth pairing was designed with moderate security in mind
- Car kits allow a car phone to retrieve and use session keys from a mobile phone smartcard
- Car kit requires higher level of security
  - users have to enter 16-character passcodes

More secure = Harder to use?

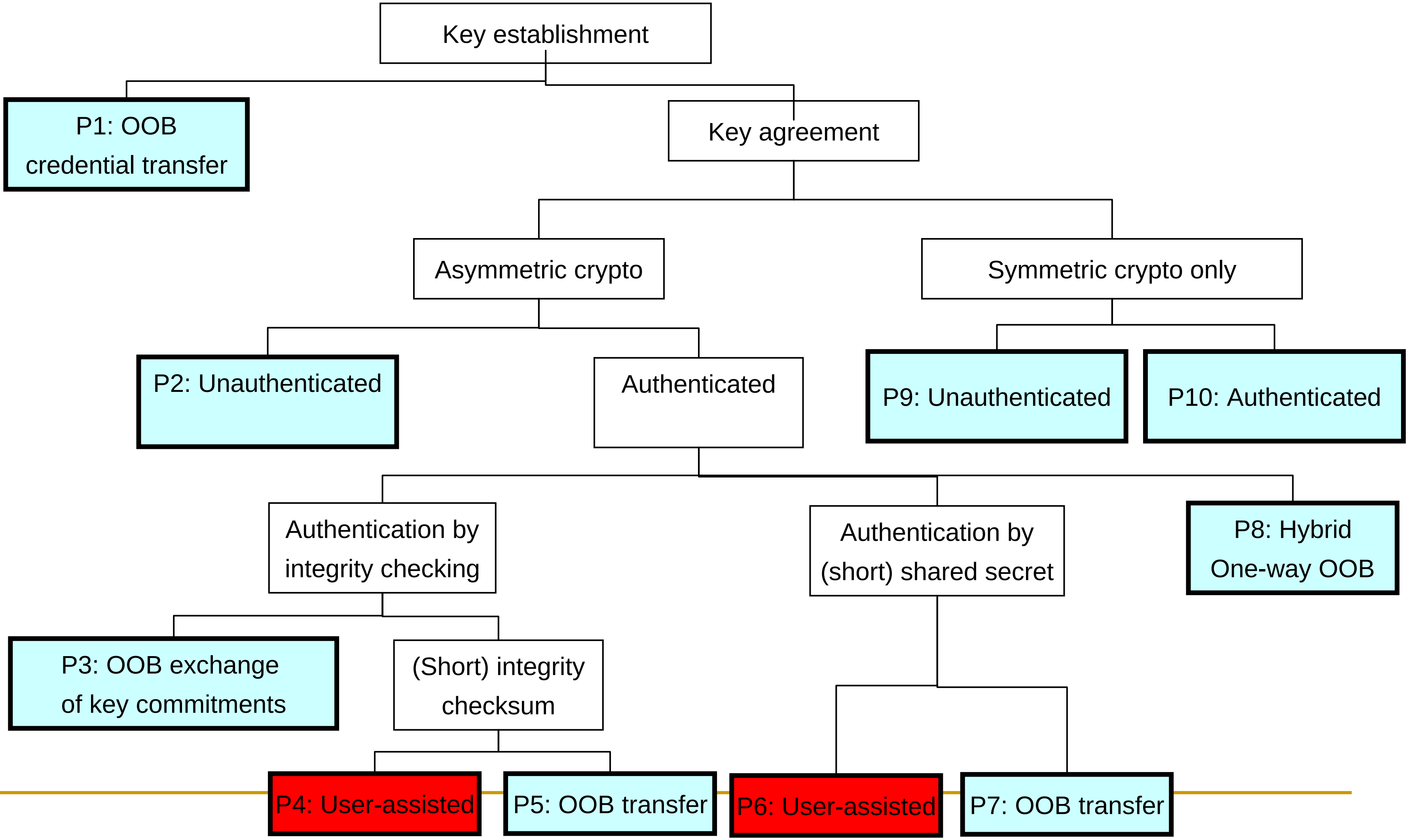
(Uzun et al. USEC'07)

Wanted: Secure, intuitive, inexpensive techniques for device pairing

- Two (initial) problems to solve
  - Discovery: finding the other device
  - **Authenticated key agreement**: setting up keys for subsequent communication
- Assumption: Peer devices are physically identifiable
- Idea: Use a secure channel to transport security-critical information
  - **Human user or** auxiliary secure channel

(Uzun et al. USEC'07)

User-mediated mechanisms for key establishment



(Uzun et al. USEC'07)

Suomalainen, Valkonen, Asokan [NRC-TR-2007-004]

Current Standardization Activities

- WiFi
  - WiFi Protected Setup (P1, P2, P3, P6, P8), Jan 2007
    - Announcement: http://www.wi-fi.org/news/pressrelease-081606-WiFiProtectedSetup/
  - Windows Connect Now (P1, P6)
    - Specifications: http://download.microsoft.com/download/a/f/7/af7777e5-7dcd-4800-8a0a-b18336565f5b/WCN-Netspec.doc
    - similar to WiFi Protected Setup
- Bluetooth Secure Simple Pairing, Feb 2007
  - White paper: http://bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf
- Wireless USB Association Models Supplement, 2006
  - http://www.usb.org/developers/wusb/wusb_2006_0302.zip (P1, P4)
- Others are in the works

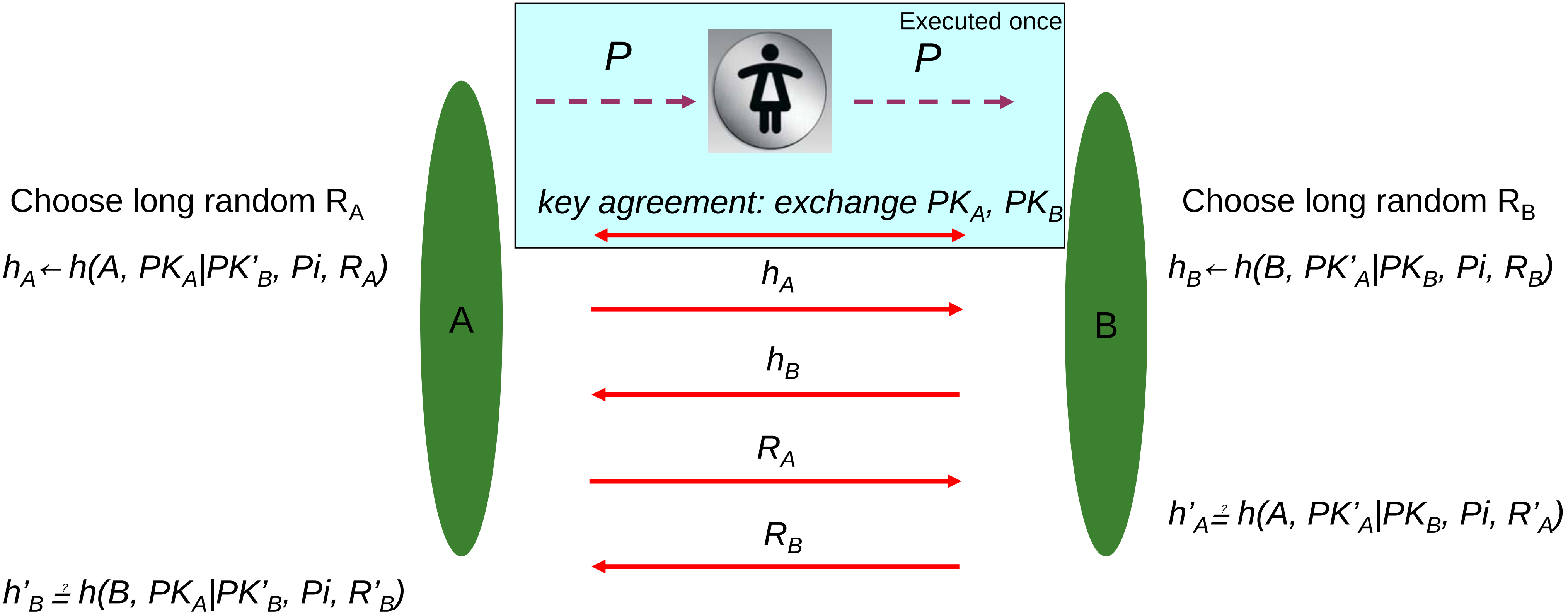
(Uzun et al. USEC'07)

“User as the secure channel” cases only

- Using a **short** secret Passkey (P6)
- Comparing **short** non-secret check codes (P4)
- Using a short key/code should not hamper long term security
  - Standard security against offline attacks
  - Good enough security against man-in-the-middle

(Uzun et al. USEC'07)

Authentication using secret short passkeys



Executed once

P

P

key agreement: exchange PK_A, PK_B

Choose long random R_A

$h_A \leftarrow h(A, PK_A|PK'_B, Pi, R_A)$

A

h_A

h_B

R_A

R_B

B

Choose long random R_B

$h_B \leftarrow h(B, PK'_A|PK_B, Pi, R_B)$

$h'_A \overset{?}{=} h(A, PK'_A|PK_B, Pi, R'_A)$

$h'_B \overset{?}{=} h(B, PK_A|PK'_B, Pi, R'_B)$

One-time passkey *P* is split into *i* parts (*i* > 1): next 4-round exchange repeated *i* times

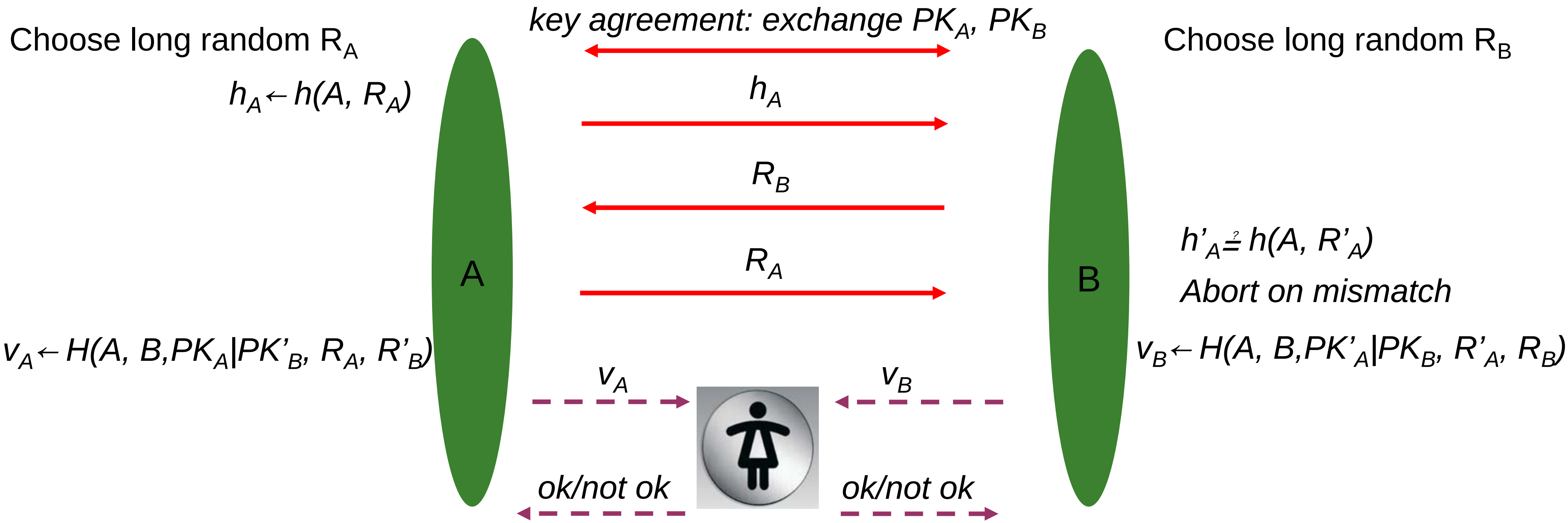
h() is a hiding commitment; in practice SHA-256

Up to $2^{-(k-1)}$ (unconditional) security against man-in-the-middle (k is the length of *P*)

Generalized version of MANAIII by Gehrmann, Nyberg, Mitchell [RSA Cryptobytes 2004]

(Uzun et al. USEC'07)

Authentication using non-secret short check codes



Choose long random R_A

$h_A \leftarrow h(A, R_A)$

key agreement: exchange PK_A, PK_B

Choose long random R_B

h_A

R_B

R_A

A

B

$h'_A \overset{?}{=} h(A, R'_A)$

Abort on mismatch

$v_A \leftarrow H(A, B, PK_A|PK'_B, R_A, R'_B)$

$v_B \leftarrow H(A, B, PK'_A|PK_B, R'_A, R_B)$

v_A

v_B

ok/not ok

ok/not ok

User approves acceptance if v_A and v_B match

h() is a hiding commitment; in practice SHA-256

H*()* is a mixing function; in practice SHA-256 output truncated to 4 digits

MANA IV by Laur, Asokan, Nyberg [IACR ePrint 2005] Laur, Nyberg [CANS 2006]

(Uzun et al. USEC'07)

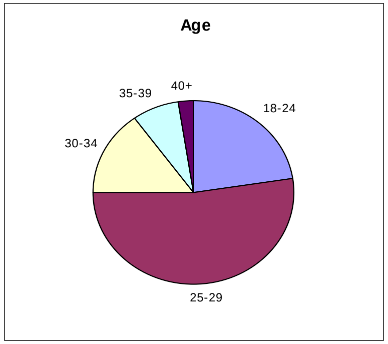
We conducted usability tests

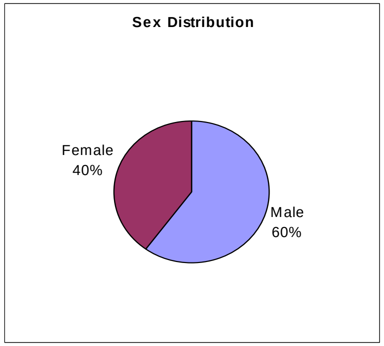
- Objectives: Study pairing proposals in emerging standards and
  - identify possible user-interaction methods
  - evaluate the methods by comparing them and
  - find implementation strategies that maximize their usability and security

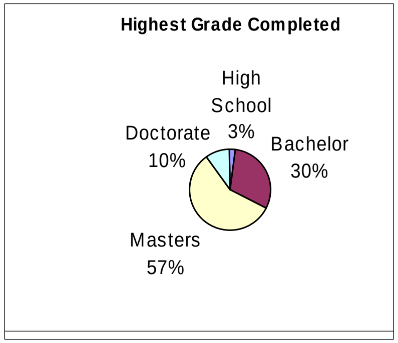
(Uzun et al. USEC'07)

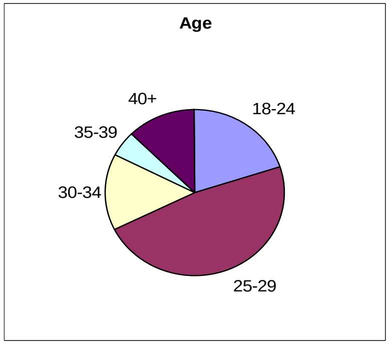
Who Tested the protocols (1/2)

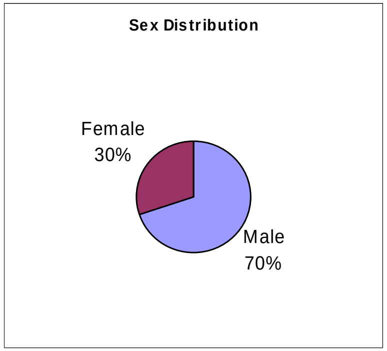
- Two groups of forty people with the following main demographics.

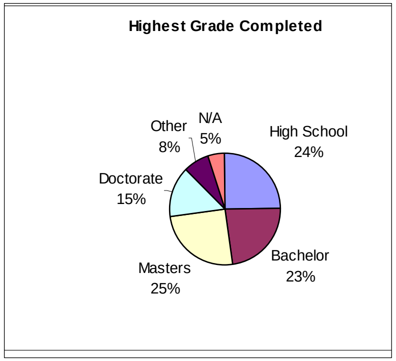












(Uzun et al. USEC'07)

Who Tested the protocols (2/2)

- Background of the test participants
  - On average, spending 7 hr/day in front of a computer.
  - All are mobile phone or PDA users.
  - 60% have a mobile device with Bluetooth, WI-FI, Infra-red capability.
  - 35% use Bluetooth, infrared or WI-FI regularly
    - Half of who doesn’t have Bluetooth or WI-FI in their device are planning to buy a new one in 6 months.
- Well educated and technology-aware user group!

(Uzun et al. USEC'07)

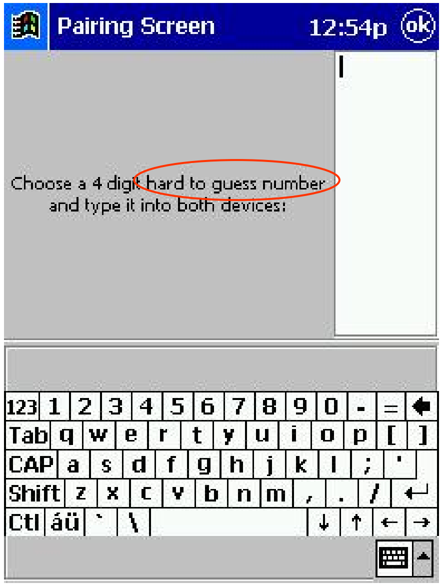
Tested user interaction methods

- Each pairing method admits different user interaction methods
- Comparing **short non-secret** check codes
  - Compare-and-Confirm
  - Select-and-Confirm
  - Copy-and-Confirm
- Using a **short secret** Passkey
  - Copy
  - Choose-and-Enter

(Uzun et al. USEC'07)

Choose-and-Enter (1/2)

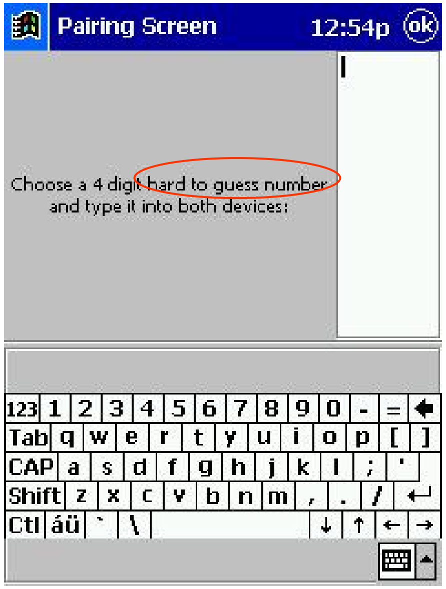
- User chooses number as passkey and types it into the both devices. (Like in current Bluetooth pairing in many phones)
  - Method: Specifically asked for a hard to guess 4-digit passkey



Pairing Screen 12:54p

Choose a 4 digit hard to guess number and type it into both devices:

ok
123 1 2 3 4 5 6 7 8 9 0 - =
Tab q w e r t y u i o p [ ]
CAP a s d f g h j k l ; '
Shift z x c v b n m , . /
Ctl áü ` \ ↓ ↑ ← →



Pairing Screen 12:54p

Choose a 4 digit hard to guess number and type it into both devices:

ok
123 1 2 3 4 5 6 7 8 9 0 - =
Tab q w e r t y u i o p [ ]
CAP a s d f g h j k l ; '
Shift z x c v b n m , . /
Ctl áü ` \ ↓ ↑ ← →

Short secret passkey

(Uzun et al. USEC'07)

Choose-and-Enter (2/2)

- Results
  - Participants considered it professional, and they liked it.
  - 15% percent explicitly complained about the hardness of **coming up with a random number.**
  - Took about 32 seconds on average. **Longest** among tested.
  - **42.5%** used very predictable repeating or in-sequence numbers. More severely, they all admitted reading the warning!
  - Provided **Worst security** among the tested.

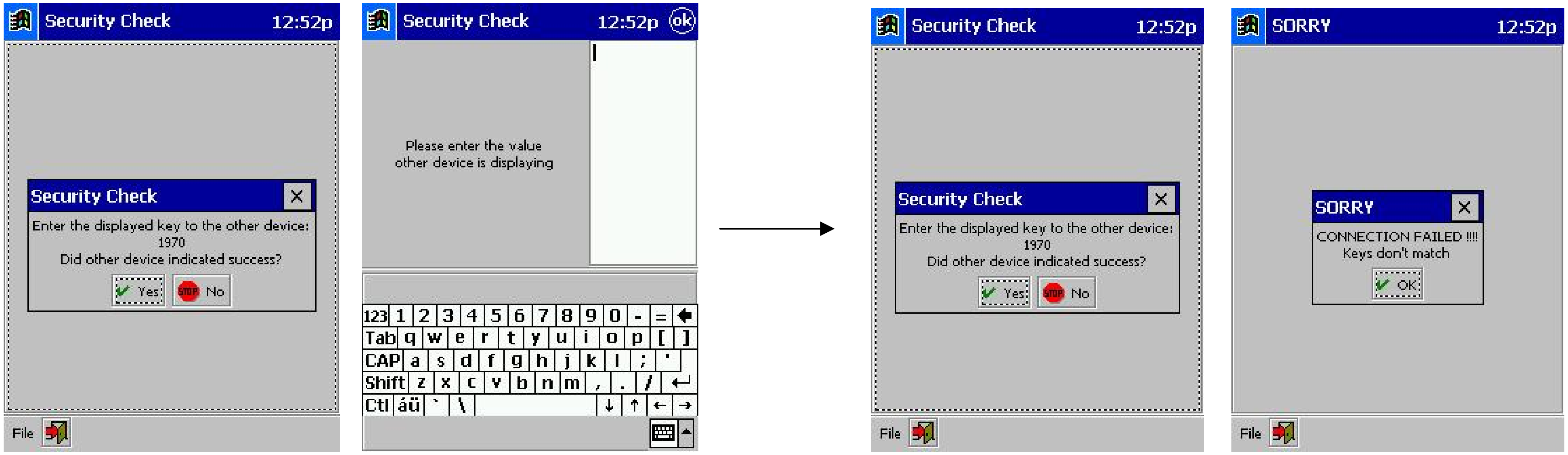
- This method is clearly out of picture for achieving usable security.

Short secret passkey

(Uzun et al. USEC'07)

Copy-and-Confirm (1/2)

- One device shows a number and asks user to type it into the second device. User confirms on the first device after seeing success on the second.
  - Method: first device shows a 4-digit number and a yes/no confirmation question



Enter the displayed key to the other device:
1970
Did other device indicated success?

Please enter the value
other device is displaying

Enter the displayed key to the other device:
1970
Did other device indicated success?

CONNECTION FAILED !!!
Keys don't match

Short non-secret checksum

(Uzun et al. USEC'07)

Copy-and-Confirm (2/2)

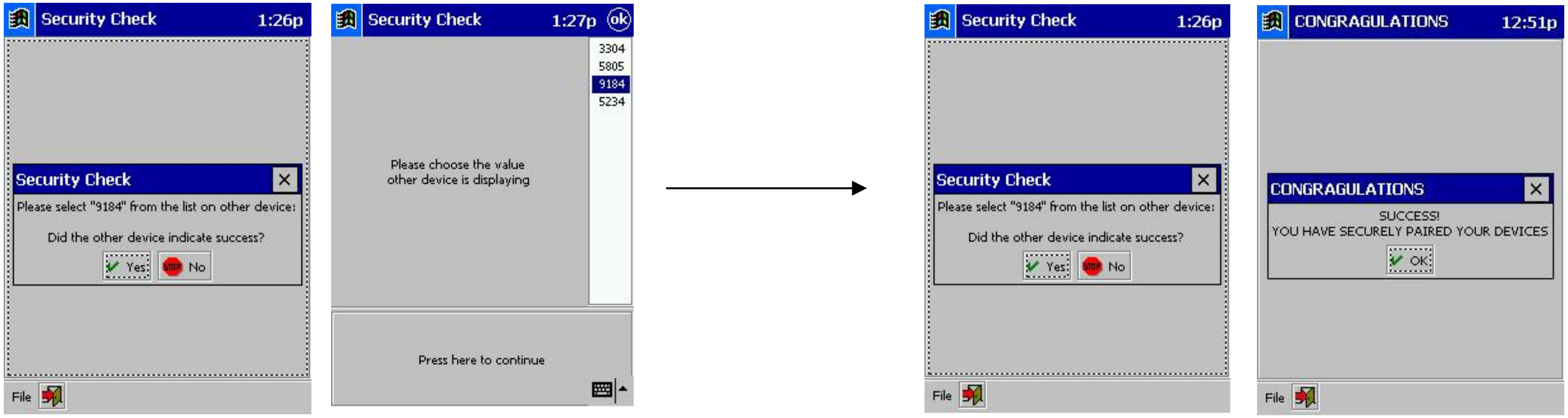
- Results
  - Users didn’t like two phase structure (copying first and confirming next)
  - Took around 27 seconds.
  - 10% didn’t wait for success indication before confirming on the first device.
- Better to use *Copy* without confirmation phase although *Copy* requires the passkey to be kept secret.

Short non-secret checksum

(Uzun et al. USEC'07)

Select & Confirm (1/2)

- One device shows a number and the other device shows a set of numbers. User selects the matching value and confirms on the first device after seeing success indication.
  - Method 1: 4-Digit number, 4 item selection list



Security Check 1:26p

Security Check

Please select "9184" from the list on other device:

Did the other device indicate success?

Yes No

File

Security Check 1:27p ok

3304
5805
9184
5234

Please choose the value other device is displaying

Press here to continue

Security Check 1:26p

Security Check

Please select "9184" from the list on other device:

Did the other device indicate success?

Yes No

File

CONGRAGULATIONS 12:51p

CONGRAGULATIONS

SUCCESS!
YOU HAVE SECURELY PAIRED YOUR DEVICES

OK

File

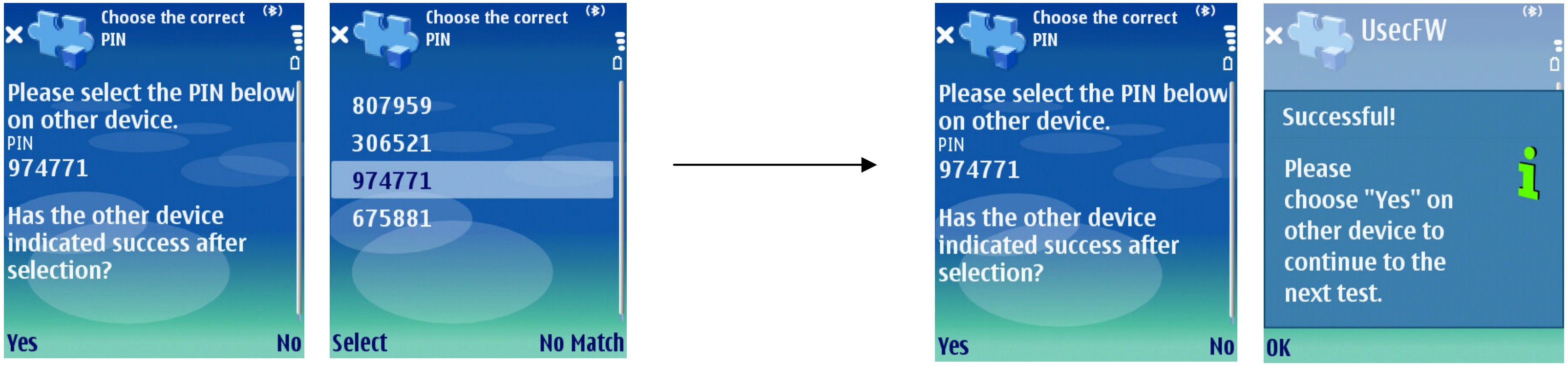
  - Results
    - 7.5% error on choosing the correct value.
    - 12.5% confirmation without seeing the success indication.

Short non-secret checksum

(Uzun et al. USEC'07)

Select & Confirm (2/2)

- Method 2: 6-digit number, 4 item selection list, improved UI.



Choose the correct PIN

Please select the PIN below on other device.

PIN

974771

Has the other device indicated success after selection?

Choose the correct PIN

807959

306521

974771

675881

Choose the correct PIN

Please select the PIN below on other device.

PIN

974771

Has the other device indicated success after selection?

UsecFW

Successful!

Please choose "Yes" on other device to continue to the next test.

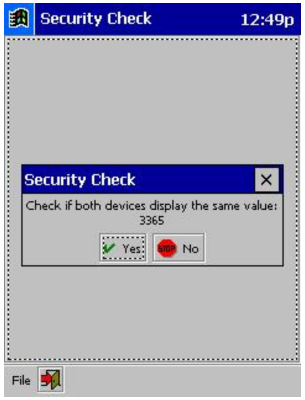
- Results
  - Despite GUI improvements, still 5% didn’t wait for the success indication.
  - 2.5% error on choosing the correct value.
  - Users find it fun to use but two-phase interaction is still confusing for some users

Short non-secret checksum

(Uzun et al. USEC'07)

Compare-and-Confirm (1/2)

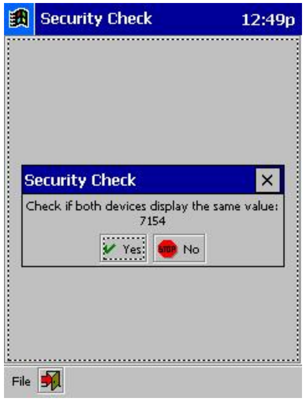
- Each device shows a number and asks user to compare shown values.
  - Method 1: 4-digit numbers; straight-forward implementation of YES/NO question.



Security Check 12:49p

Security Check

Check if both devices display the same value:
3365



Security Check 12:49p

Security Check

Check if both devices display the same value:
7154

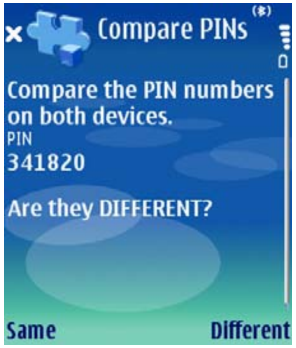
  - Results
    - Takes around 15 seconds.
    - **85% found it easiest** but only 10% found it professional!
    - **20%** pressed “yes” on non-matching values without reading instructions!

Short non-secret checksum

(Uzun et al. USEC'07)

Compare-and-Confirm (2/2)

- Method 2
  - 6-digits
  - Different question, uncommon answers (same/different).
  - Putting the negative answer as default key action.

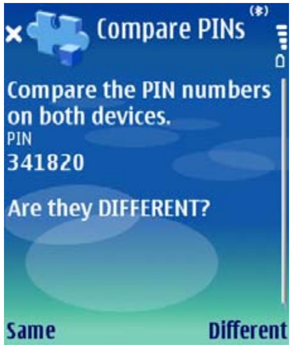


Compare the PIN numbers on both devices.

PIN

341820

Are they DIFFERENT?



Compare the PIN numbers on both devices.

PIN

341820

Are they DIFFERENT?

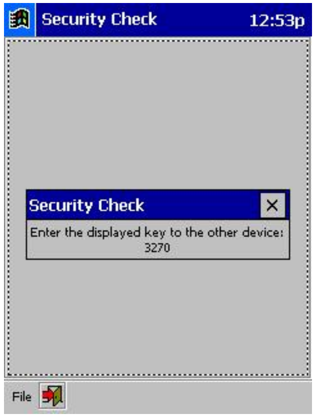
- Results
  - Takes around 17 seconds
  - **100% security** achieved, nobody said “same” on non-matching values.
  - 2.5% erroneously cancelled the connection (still on the safe side!)

Short non-secret checksum

(Uzun et al. USEC'07)

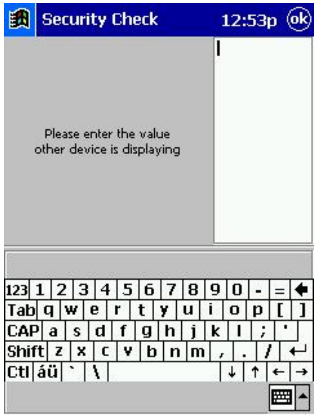
Copy (1/2)

- One device shows a number as a passkey and user types it into the second device. Devices accept or cancel automatically.
  - Method 1: 4-digit passkey



Security Check

Enter the displayed key to the other device:
3270


Security Check
12:53p
ok
123 1 2 3 4 5 6 7 8 9 0 - =
Tab q w e r t y u i o p [ ]
CAP a s d f g h j k l ; '
Shift z x c v b n m , . /
Ctl áü ` \ ↓ ↑ ← →

Please enter the value
other device is displaying

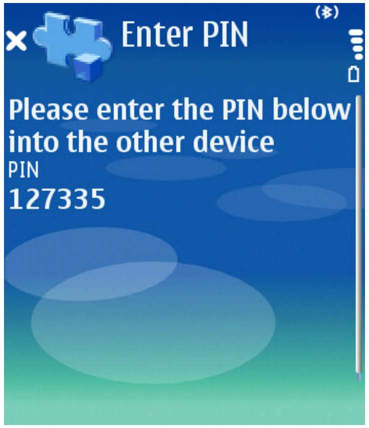
  - Method 2
    - 8-digit passkey
- Results
  - Users find this method hard to use but professional, they like and want to see it on their devices.
  - 4-digit doesn’t provide enough security for most cases and 95% of users found 8-digit too much.

Short secret passkey

(Uzun et al. USEC'07)

Copy (2/2)

- Method 3
  - 6-digit passkey

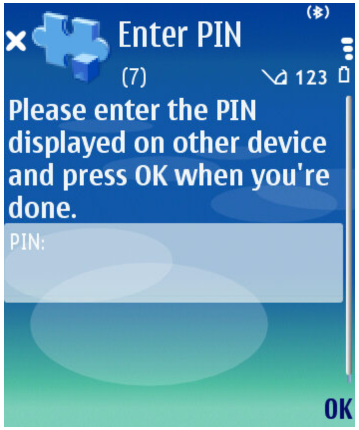


Enter PIN

Please enter the PIN below into the other device

PIN

127335



Enter PIN

(7)

123

Please enter the PIN displayed on other device and press OK when you're done.

PIN:

OK

- Results
  - 6-digit seems to have the balance but still rated as hardest.
  - Using 6-digit takes around 13 seconds in phones and provides **97% success rate**.
  - Naturally Secure. Not easy to make it insecure by simple user mistakes.

Short secret passkey

(Uzun et al. USEC'07)

Conclusions

- Security protocols should give extra importance to usability since there is no room for any error.
    - Users' cognitive abilities and tendencies are the key concepts.
- Some lessons learnt:
    - Avoid multi-step interaction where user can change the assumed order
        - If security relies on a certain order of steps, make sure that users cannot change the order
        - Don't rely on instructions you give, they may not read!
    - Follow the Saltzer-Schroeder "Fail-safe defaults" principle: Always put the safest option as the next default
    - Make questions clear and short, if possible guide clearly about the next action (E.g. press YES on the other device).
        - In practice, this is difficult without standardizing UIs
    - Avoid familiar labels, especially those that have direct negative or positive associated meaning. Instead use words specific to the required task.
        - E.g., SAME/DIFFERENT rather than YES/NO, CANCEL/CONTINUE
        - But impact of learning effect needs to be studied further
    - Demand as less brain intensive work as possible from users.
        - Don't expect that a user will like copying 16 digits to pair a car-kit, they'll hate even 8-digits (magic number 7).

(Uzun et al. USEC'07)

Discussion Points

- Concentrating on 6-digit on the second round was guided by the
  - first round results
  - FIPS 140-2 requirements
- Many changes are done between rounds for pragmatic reasons, resulting in difficulty on pinpointing the exact cause of improvement in some cases.
- Users perception of easy-to-use may not be supported by objective measurements
  - E.g. *Copy* rated as the hardest although it didn't take any more time than the other two.
- Should the things be made as easy as possible?
  - Does "easy" lead to "careless"?
  - Users tend to associate easy with insecure

(Uzun et al. USEC'07)

What is next?

- We are in the process of doing more small scale controlled tests to better understand the effects of different improvements
- We are also testing other pairing methods that uses auxiliary secure channels with less user involvement.
  - ❑ Touching devices to each other
  - ❑ Recording the video of the other devices flashing its screen or LED.
  - ❑ Devices talking (over audio) to each other, or user comparing what he hears with what he sees.
  - ❑ User identifying synchronized audio, blinking or vibration patterns or composition of them. (still uses human as secure channel, but they rely on more basic abilities)
- We plan to test more sophisticated attack scenarios when the devices have no trusted path to the user.
- We plan the modify our test framework to enable conducting longer term tests in user’s familiar environment.

(Uzun et al. USEC'07)

Selected Related Work & Pointers

- Security Associations in Personal Networks: A Comparative Analysis (Suomalainen et al.)
- Low-cost Manufacturing, Usability, and Security: An Analysis of Bluetooth Simple Pairing and Wi-Fi Protected Setup (Kuo et al.)
- Schemes using different auxiliary channels
    - Seeing-Is-Believing (McCune et al.)
    - Secure Device Pairing based on a Visual Channel (Saxena et al.)
    - Loud and Clear: Human-Verifiable Authentication Based on Audio (Goodrich et al.)
    - Talking to Strangers (Balfanz et al.)

Thanks!

- Questions?