The case for Usable Mobile Security

N. Asokan,
Nokia Research Center

Joint work with Cynthia Kuo (NRC)

August 2012

NOKIA

http://www.dilbert.com (11/16/2007)

NOKIA

Outline

- Why worry about usable security?
- What is special about mobile?
- Some examples of mobile usable security problems we face
  - A look back: The “First Connect” story
  - Current problems
    - Local (user) authentication
    - Mobile CAPTCHA
    - Trustworthy installation
    - Theft resistance and data/credential recovery
    - ….
- Conclusions

Why worry about usable security

Lack of security usability

- harms security, eventually
- lowers overall attractiveness of the device/service, eventually
- costs money!

In many cases, the source of the "cost" is surprising

Why?

NOKIA

What is special about mobile?

Your mobile phone: Not a smaller version of your PC

!=

Your mobile phone: Not a smaller version of your PC

Mobile phone applications have different requirements due to

1. Smaller physical screen size

   → Less room for security indicators, notifications etc.

2.4″

3.5″

10.1″

20″

Mobility

Your mobile phone: Not a smaller version of your PC

Mobile phone applications have different requirements due to

1. Smaller physical screen size
2. Different input mechanisms

Directional pad + keyboard

Touch screen

Keyboard + mouse

Mobility

Your mobile phone: Not a smaller version of your PC

Mobile phone applications have different requirements due to

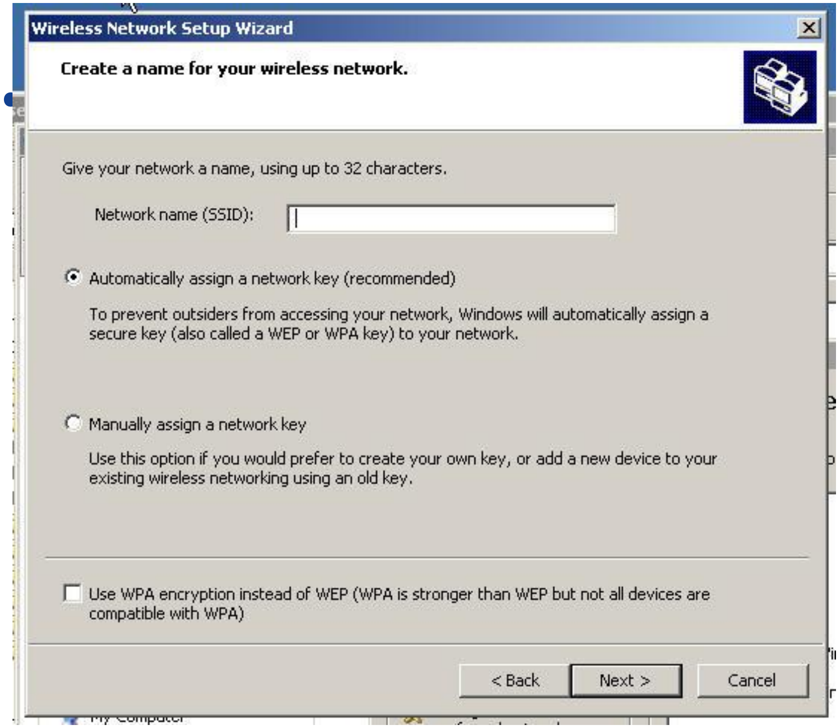
1. Smaller physical screen size
2. Different input mechanisms
3. Limited battery life
4. More prone to theft/loss
5. Slower and less reliable network connectivity
6. (Comparatively) limited computational power

Example: Setting up the first connection

- **First Connect**: setting up contexts for subsequent communication.
  - Typically for proximity communications between personal devices, 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 (circa 2006)**: Secure First Connect for personal devices
  - Initializing security associations (as securely as possible)
  - No security infrastructure (no PKI, key servers etc.)
  - Ordinary non-expert users
  - Cost-sensitive commodity devices

Prevalent mechanisms were not intuitive



SSID? WPA?
Passcode?

Example: First Connect

… and not very secure

Cracking the Bluetooth PIN*

Yaniv Shaked and Avishai Wool

School of Electrical E
Tel Aviv University, Ram
shakedy@eng.tau.ac.il,

Abstract

This paper describes the implementation of an attack on the Bluetooth security mechanism. Specifically, we de-

Security Weaknesses in Bluetooth

Markus Jakobsson and Susanne Wetzel

Lucent Technologies - Bell Labs
Information Sciences Research Center
Murray Hill, NJ 07974
USA
{markusj,sgwetzel}@research.bell-labs.com

Abstract. We point to three types of potential vulnerabilities in the Bluetooth standard, version 1.0B. The first vulnerability opens up the system to an attack in which an adversary under certain circumstances is able to determine the key exchanged by two victim devices, making

Naïve usability measures damage security

http://www.helsinki-hs.net/news.asp?id=20030930IE16

HELSINGIN SANOMAT
INTERNATIONAL EDITION

TODAY | THIS WEEK | WEBORTAGE | THIS IS

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

Example: First Connect

NOKIA

Naïve security erodes 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.

- 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?

Cost:

Calls to Customer Support

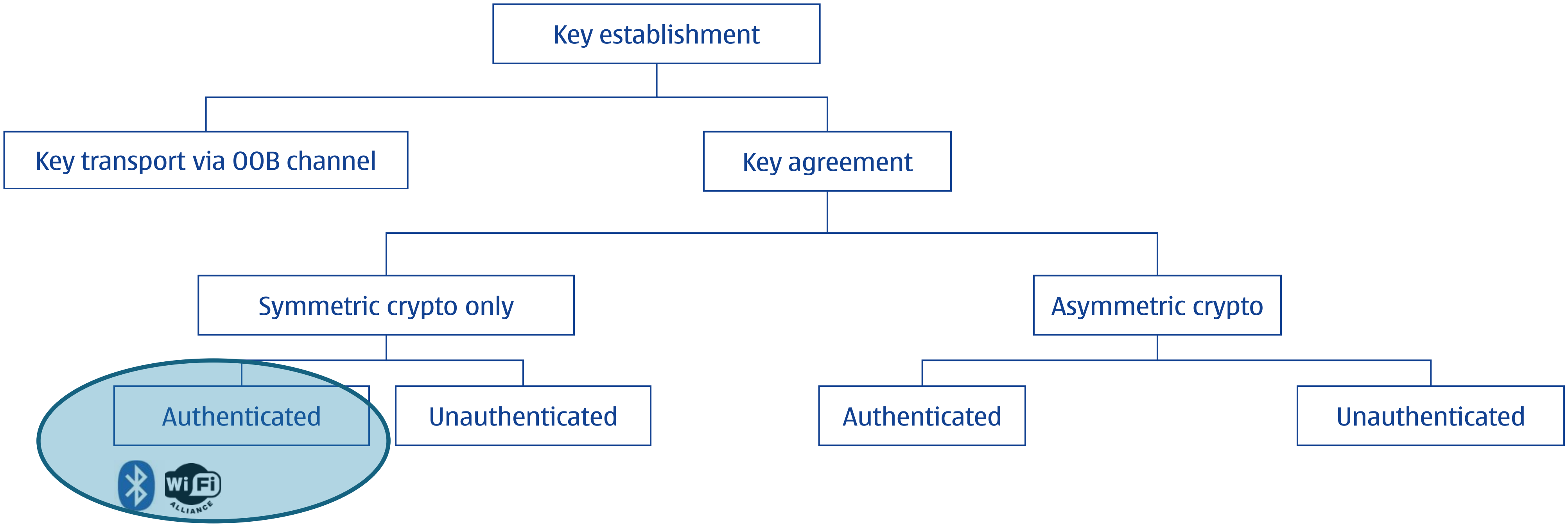
Example: First Connect

Wanted: intuitive, inexpensive, secure first connect

- Two (initial) problems to solve
  - Peer discovery: finding the other device
  - **Authenticated key establishment**: setting up a security association

- Assumption: Peer devices are physically identifiable

Key establishment for first connect ~2006



Short keys vulnerable to passive attackers

Secure against passive attackers

Example: First Connect

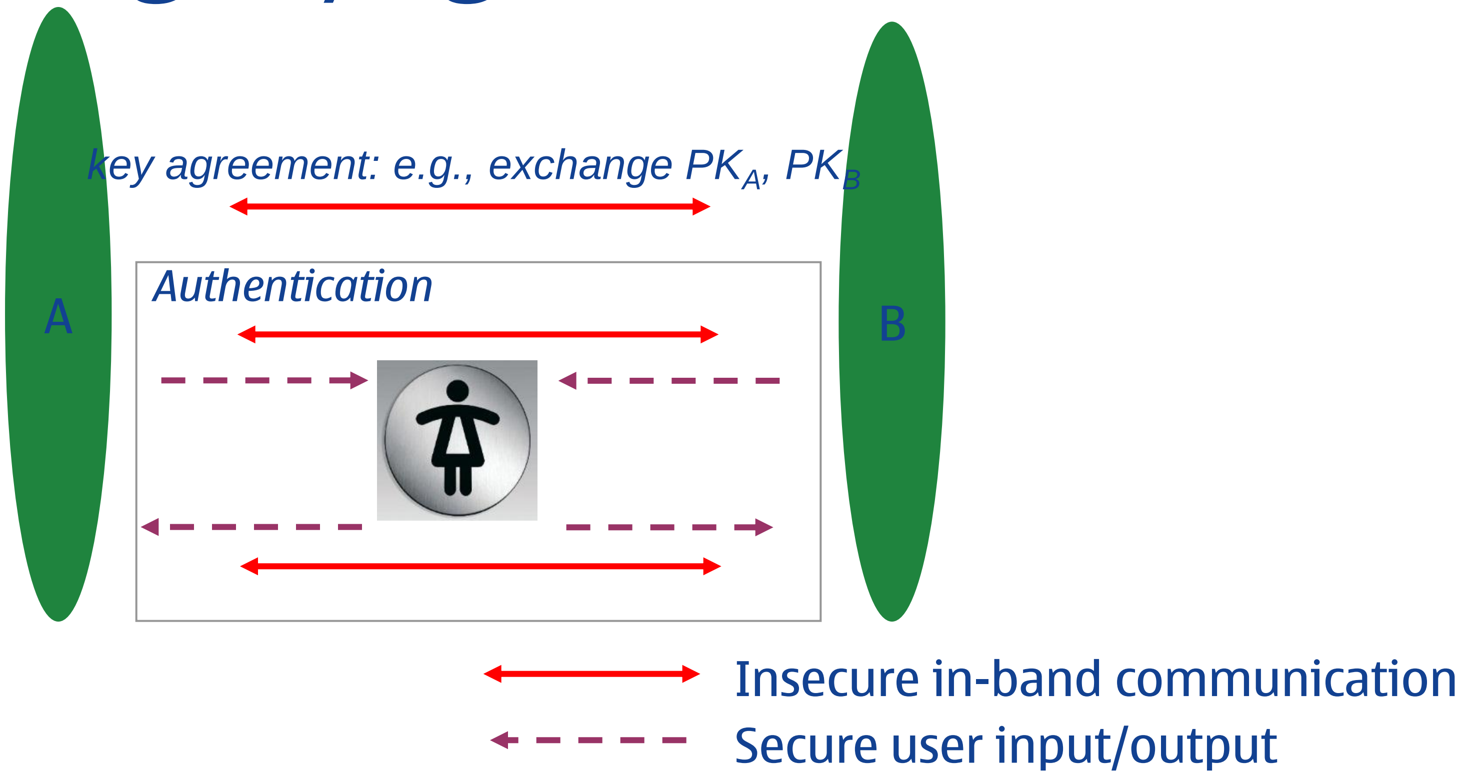
NOKIA

Authenticating key agreement

- Use an auxiliary channel to transfer information needed for authentication
- Two possibilities for realizing secure auxiliary channel
  - User assistance
  - Other out-of-band secure communication channels:
    - E.g., Near Field Communication, infrared, …

Example: First Connect

Authenticating key agreement: user-assisted



- User "bandwidth" is low (4 to 6 digits)
- Directionality depends on available hardware (1-way or 2-way)
- Security properties (integrity-only, or integrity+secrecy)

Example: First Connect

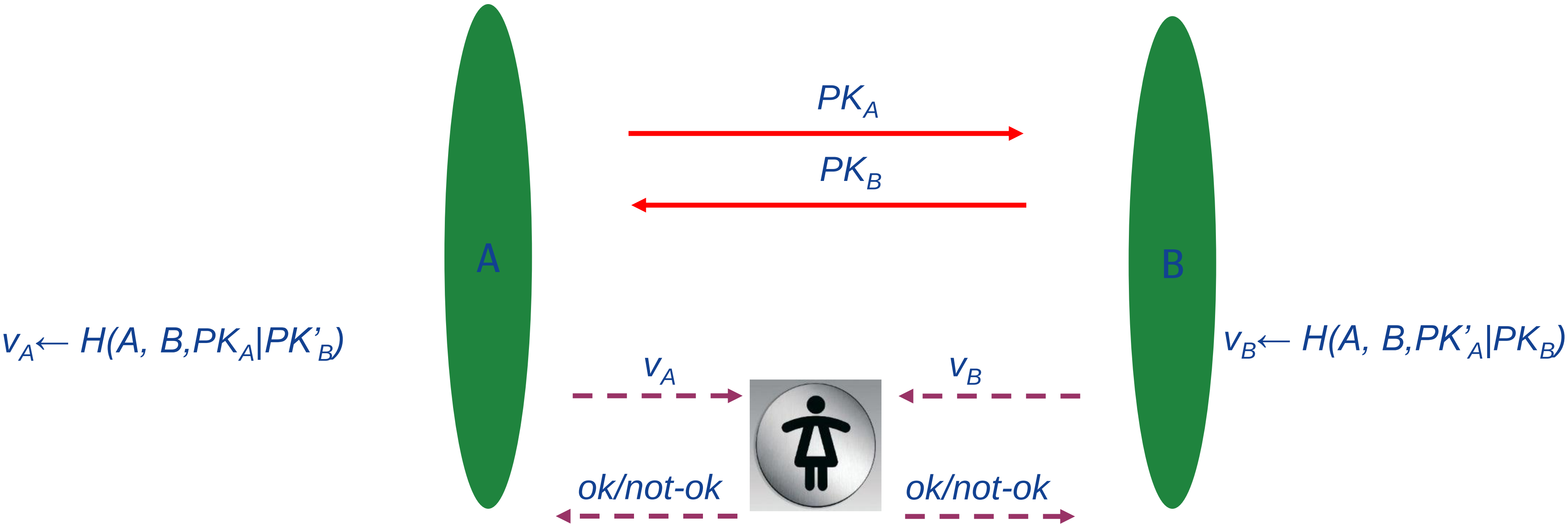
NOKIA

User as the secure channel

- Peer discovery by “user conditioning”: introduce a special first connect mode
  - E.g., Press a button to put device into the special mode
  - Demonstrative/indexical identification
- Authentication of key agreement by
  - Comparing **short** non-secret check codes (aka “short authentication string”), and
  - entering a **short secret** Passkey
- Short key/code should not hamper security
  - Standard security against offline attacks
  - Good enough security against active man-in-the-middle

Example: First Connect

Authentication by comparing short strings



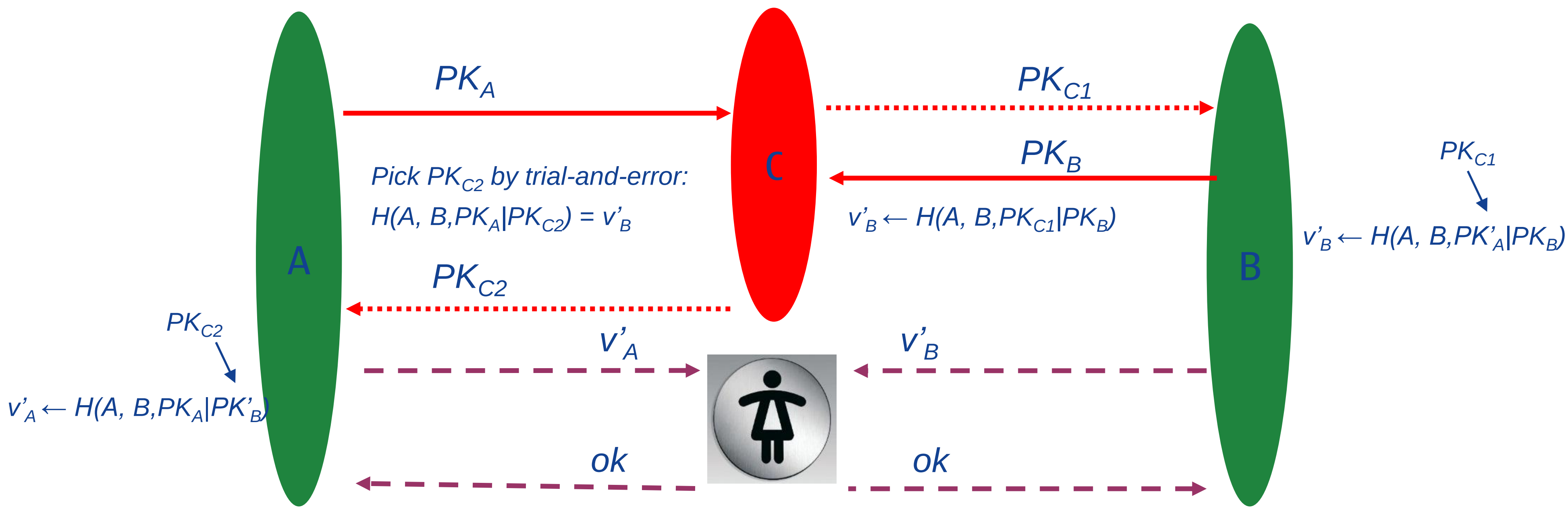
v_A and v_B are short strings (e.g., 4 digits),

User approves acceptance if v_A and v_B match

A man-in-the-middle can easily defeat this protocol

Example: First Connect

MitM in comparing short strings



Guess a value SK_{C2}/PK_{C2} until $H(A, B, PK_A|PK_{C2}) = v'_B$

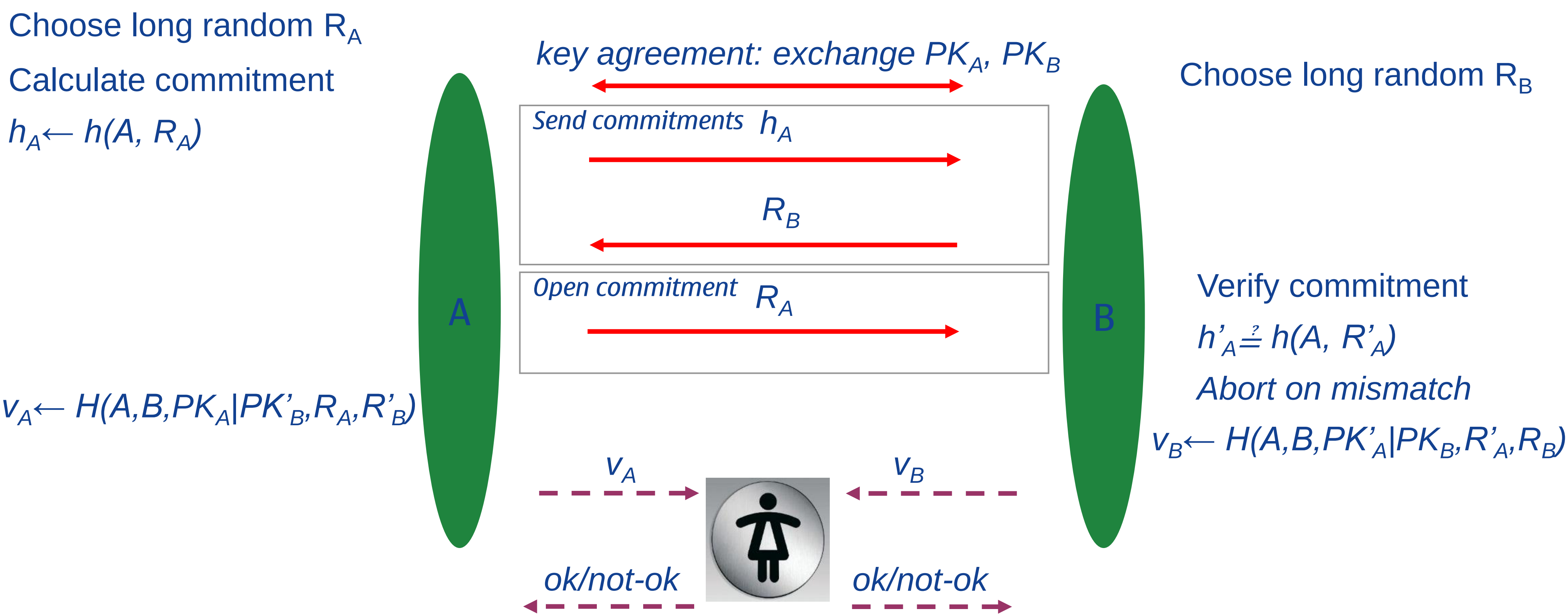
If v'_B is n digits, attacker needs at most 10^n guesses; Each guess costs one hash calculation

A typical modern PC can calculate 100000 MACs in 1 second

Example: First Connect

NOKIA

Authentication by comparing short strings



User approves acceptance if v_A and v_B match

2^{-l} (“unconditional”) security against man-in-the-middle (l is the length of v_A and v_B)

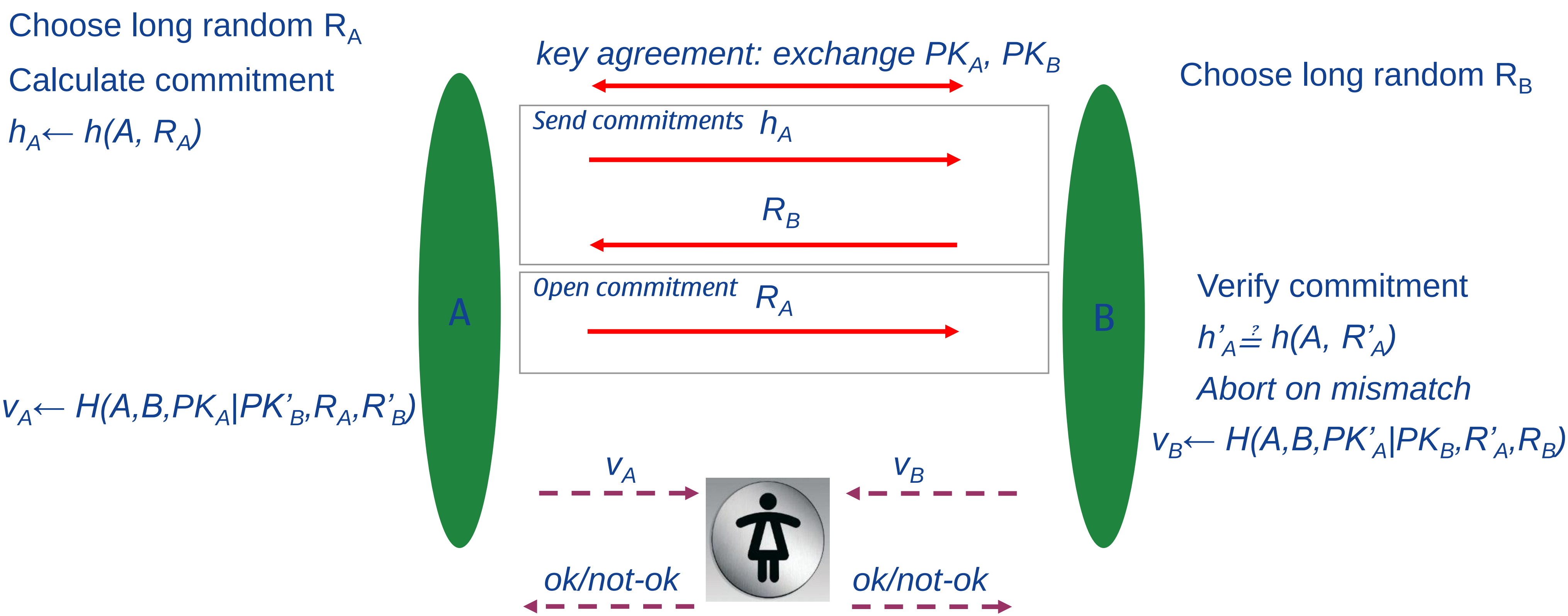
h() is a hiding commitment; in practice SHA-256

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

Example: First Connect

NOKIA

Authentication by comparing short strings



User approves acceptance if v_A and v_B match

2^{-l} ("unconditional") security against man-in-the-middle (l is the length of v_A and v_B)

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

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

Authentication using interlocking short passkeys

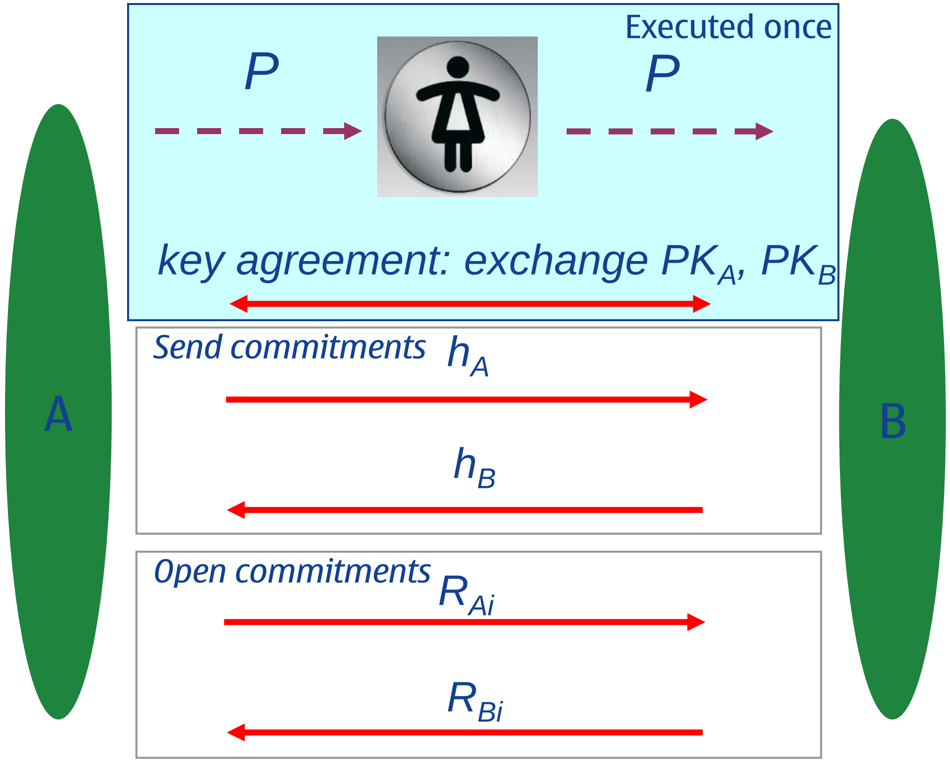
Choose long random R_{Ai}

Calculate commitment

$h_A \leftarrow h(A, PK_A|PK'_B, Pi, R_{Ai})$

Verify commitment

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



Choose long random R_{Bi}

Calculate commitment

$h_B \leftarrow h(B, PK'_A|PK_B, Pi, R_{Bi})$

Verify commitment

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

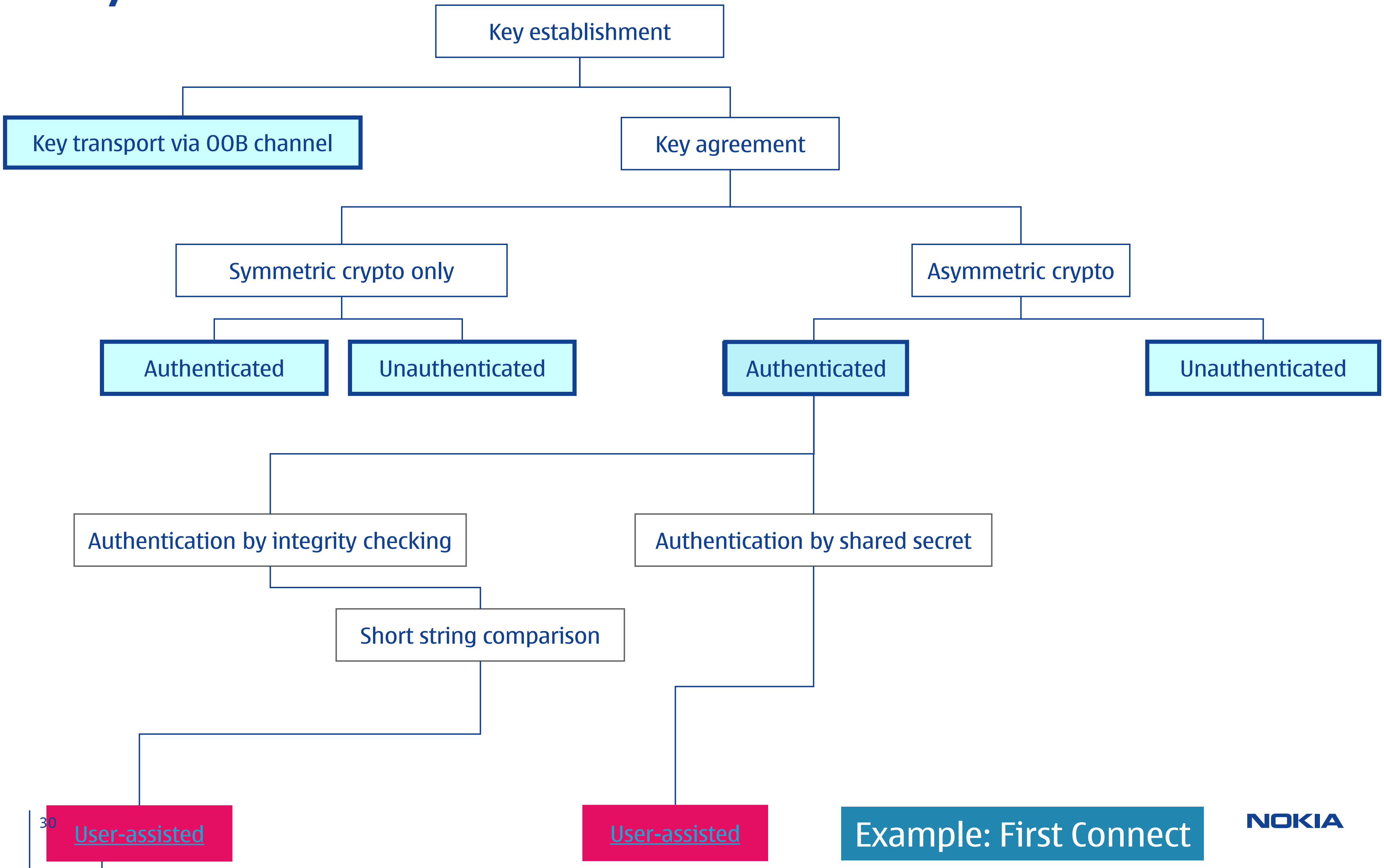
One-time passkey *P* is split into *k* parts (*k* > 1): next 4-round exchange repeated *k* times

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

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

Originally proposed by Jan-Ove Larsson [2001]: essentially multi-round MANA III

Key establishment for first connect



Example: First Connect

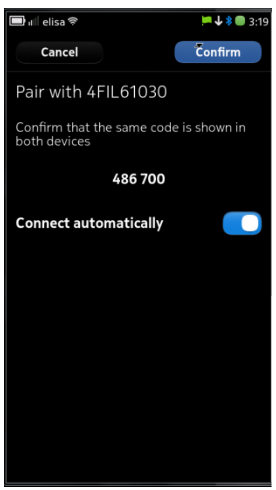
Key establishment for first connect ~2008

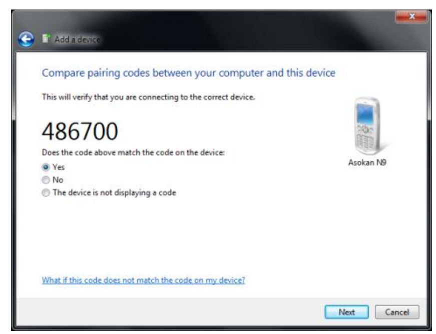
	Unauthenticated Diffie-Hellman	Authenticated Diffie-Hellman		
		short-string comparison	short PIN	Out-of-band channel
WiFi Protected Setup	“Push-button”		√	NFC
Bluetooth 2.1	“Just-works”	√	√	NFC
Wireless USB		√		USB Cable

“Security associations for wireless devices” (Overview, book chapter)
“Standards for security associations in personal networks: a comparative analysis”IJSN 4(1/2):87-100 (survey of standards)

First Connect: today

- Widely deployed (Bluetooth SSP, WiFi Protected Setup)
- **Improving usability/security → fundamental protocol changes**
  - Did it really help?
- Recent research exploiting properties of radio communication looks promising
  - Čapkun et al/TDSC 2008:5(4), Gollakota et al/Usenix Security ‘11





Example: First Connect

First Connect: A cautionary tale

Short pass keys were intended to be **one-time**

- Fixed pass keys are sometimes unavoidable
- Use of fixed pass key must be accompanied by suitable techniques to thwart online guessing attacks
  - Enter a 1-minute lock-out period after 3 failed guesses (WiFi Protected Setup)
  - Use an authenticated tunnel (a la server-authenticated TLS)
    - fixed public key (+ authenticator) to protect
    - Can you work out such a protocol?
    - (WUSB 1.1 Fixed Passkey Association Model)

December 27, 2011

Wi-Fi Protected Setup PIN brute force vulnerability

Filed under: advisories — Stefan @ 3:00 am

A few weeks ago I decided to take a look at the Wi-Fi Protected Setup (WPS) technology. I noticed a few really bad design decisions breaking the security of pretty much all WPS-enabled Wi-Fi routers. As all of the more recent router models come with WPS enabled by

I reported this vulnerability to CERT/CC and provided them with a list of (confirmed) affected vendors. CERT/CC has assigned VU#7237 To my knowledge **none** of the vendors have reacted and released firmware with mitigations in place.

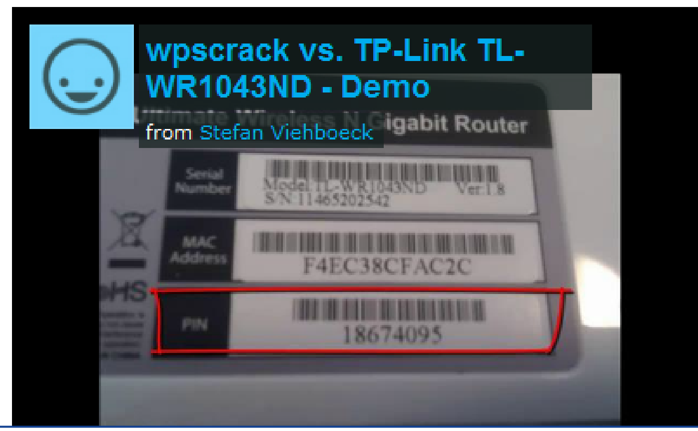
Detailed information about this vulnerability can be found in this paper: **Brute forcing Wi-Fi Protected Setup** – Please keep in mind th affected devices.

I would like to thank the guys at CERT for coordinating this vulnerability.

Update (12/29/2011 – 20:15 CET)
As you probably already know, this vulnerability was **independently** discovered by Craig Heffner (/dev/ttyS0, Tactical Network Solutic and released information about it first. Craig and his team have now released their tool "Reaver" over at Google Code.

My PoC Brute Force Tool can be found here. It's a bit faster than Reaver, but will not work with all Wi-Fi adapters.

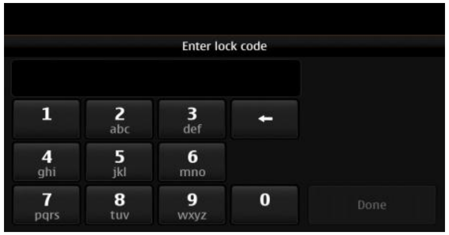
Update (12/31/2011 – 14:25 CET)



http://sviehb.wordpress.com/2011/12/27/wi-fi-protected-setup-pin-brute-force-vulnerability/
http://www.kb.cert.org/vuls/id/723755

Example: First Connect

Local user authentication: need new methods



[SOUPS '10 paper]

Need alternatives that are:

- Faster
- More enjoyable
- Secure enough

Biometrics

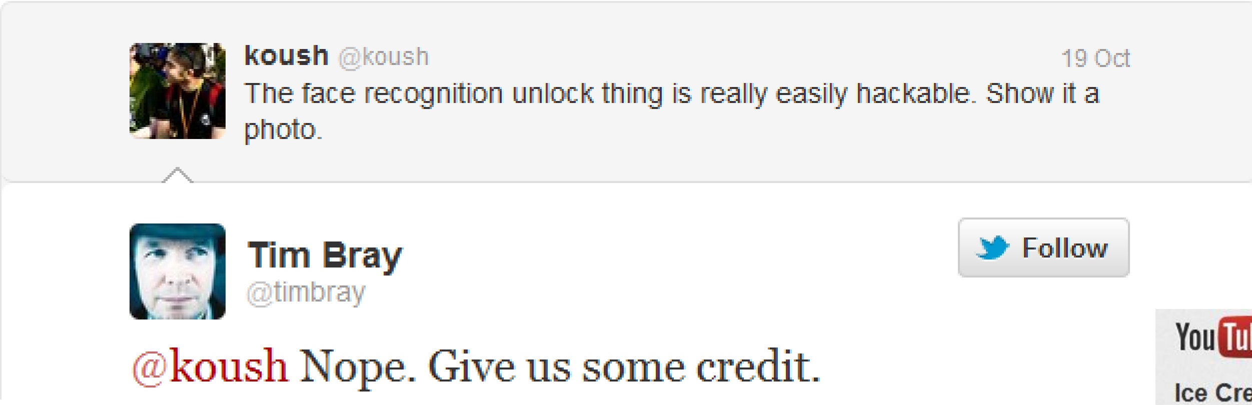
Wearables

?

Cost: users avoid using apps that mandate local authentication (work e-mail!)

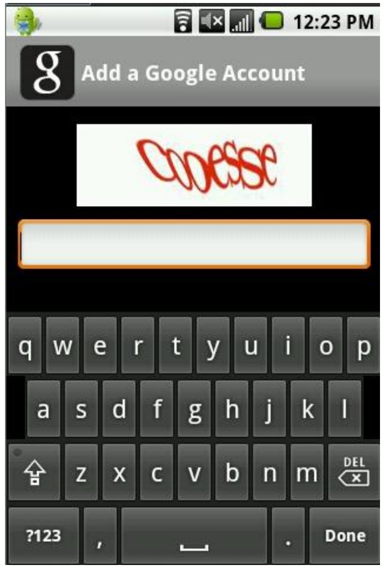
Cost: weak PINs

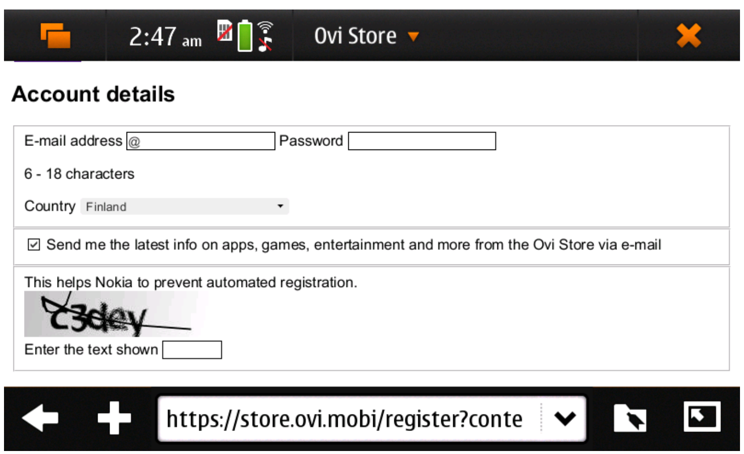
Local user authentication: a cautionary tale



http://youtu.be/BwfYSR7HttA

CAPTCHA on mobile devices





Cost:

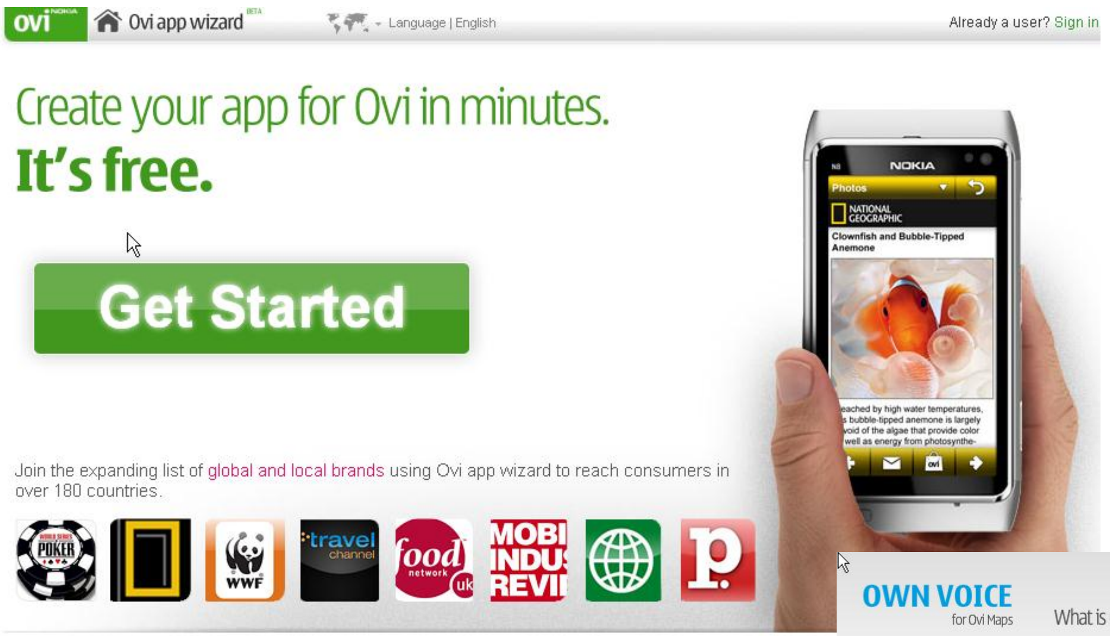
Estimated 15% drop-off rate when encountering a CAPTCHA on mobile devices

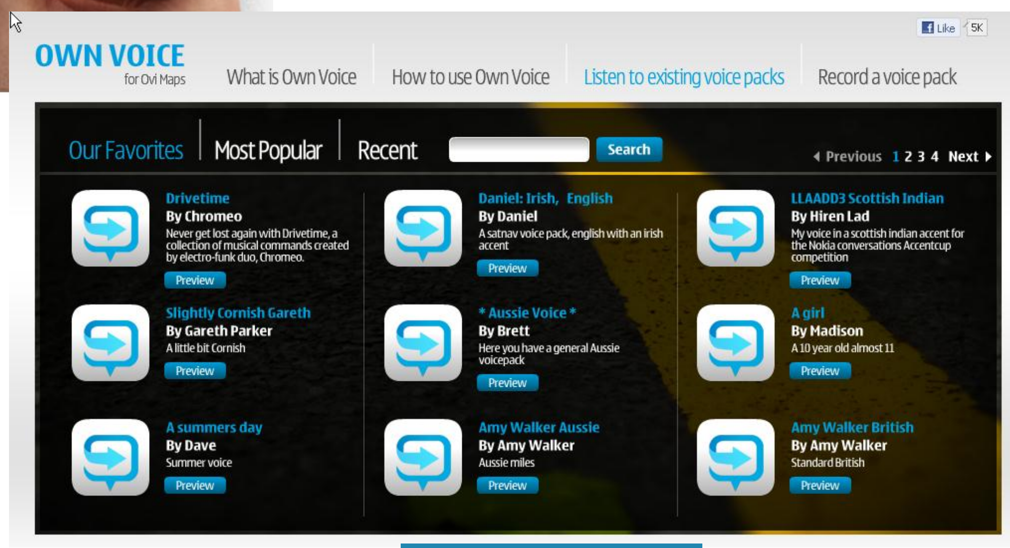
Alternatives to standard CAPTCHA?

- The problem is real
- Can it be solved without CAPTCHA? (device authentication)
- Mobile-friendly CAPTCHA variants?

Mobile CAPTCHA by Alex Smolen, Becky Hurwitz, Dhawal Mujumdar, UC Berkeley i213 Spring 2010

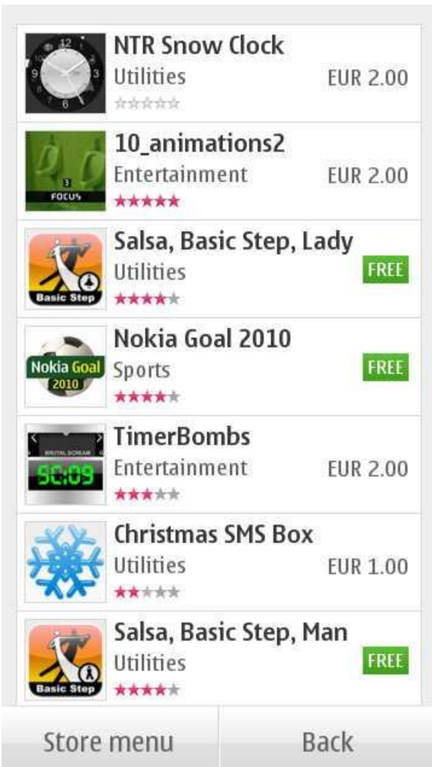
Long tail: app/content creation made easier

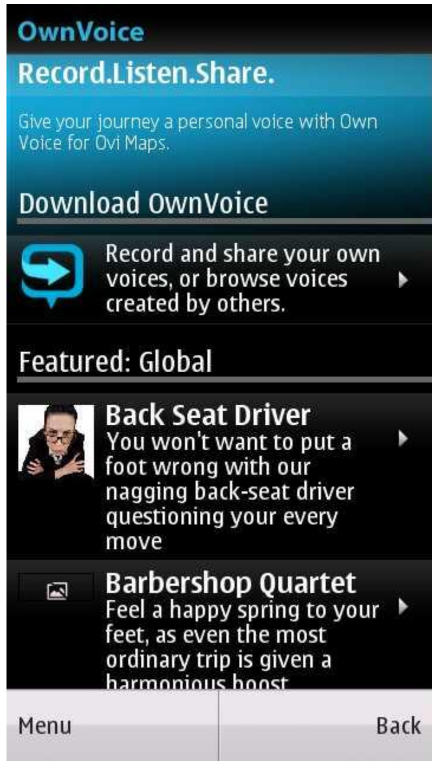


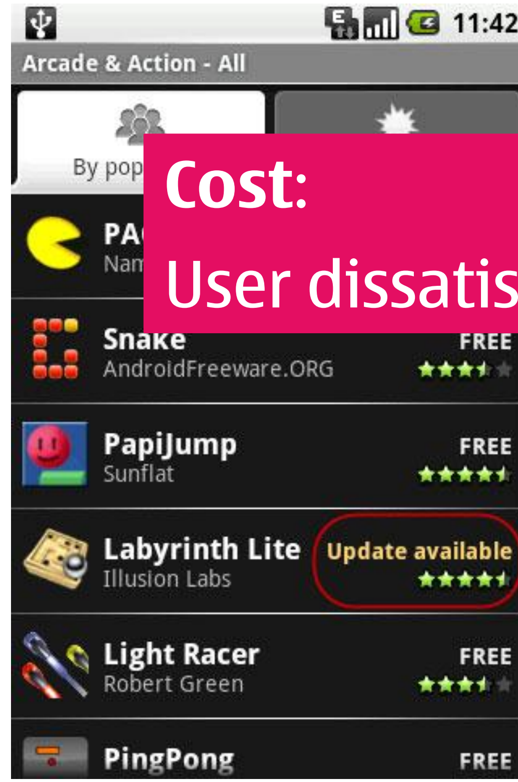


Installation

Plenty of choice for the user







Cost:
User dissatisfaction?

[”Is this App Safe?”](#)

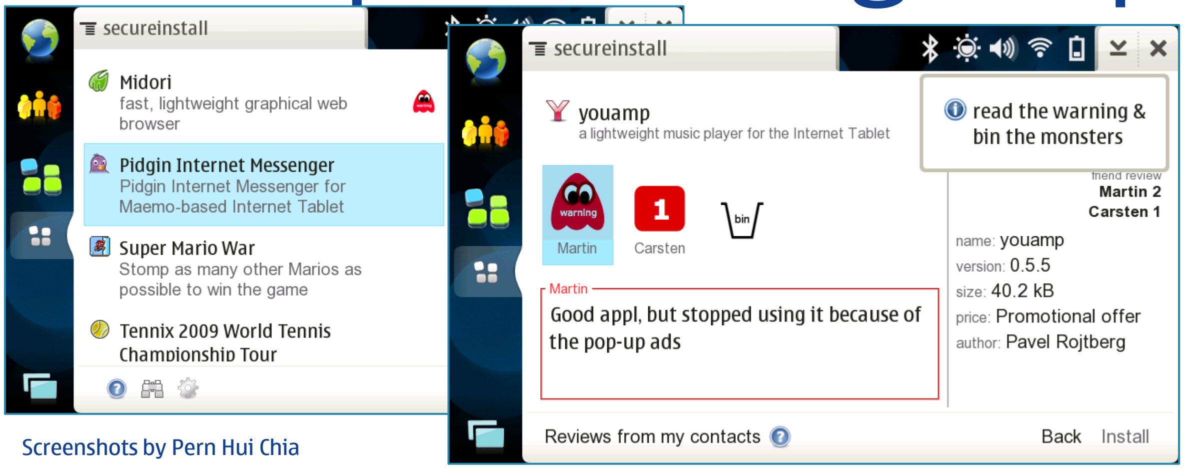
[A Large Scale Study on Application Permissions and Risk Signals](#)

([WWW 2012](#))

Installation

NOKIA

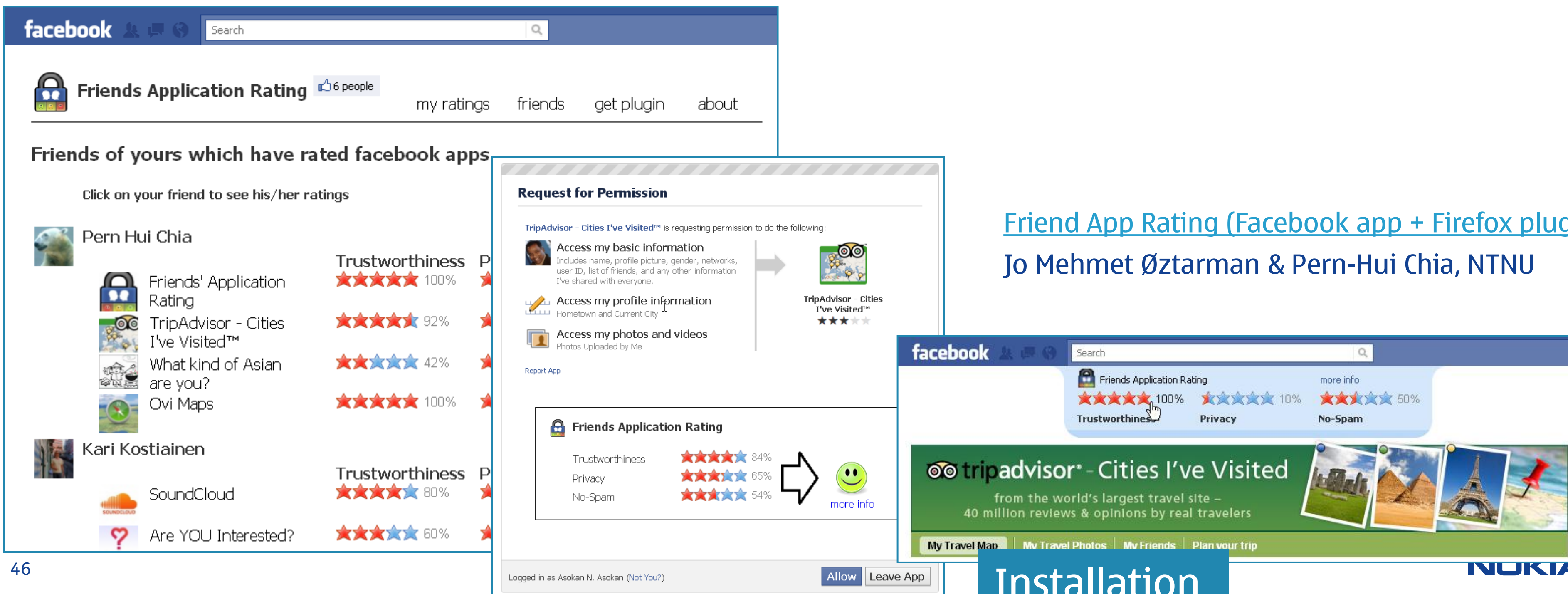
Can “clique-sourcing” help?



Screenshots by Pern Hui Chia

[Secure Installer for Nokia N810](#)

Pern-Hui Chia (NTNU) et al



[Friend App Rating (Facebook app + Firefox plugin)](#)

Jo Mehmet Øztarman & Pern-Hui Chia, NTNU

Installation

Mobile devices can help security/privacy

- Mobility and portability can help in surprising ways: e.g.,
  - PayPal Bump
  - "Mobility helps security in ad hoc networks", Čapkun et al, MobiHoc '03
  - …
- Mobiles can sense location, motion, ambient light, noise level, …
  - Cues from context/history to set sharing, access control policies
  - "CRePE: Context-Related Policy Enforcement for Android", Conti et al, ISC '10
  - ISAC (Intuitive and Sensible Access Control) project at NRC
    - SocialCom '12 Paper (to appear), older tech report, PerCom '11 Demo AISec '10 position paper.

Summary

- Usable mobile security is a challenging goal
  - Lack thereof results in surprising costs
  - Requires changes under-the-hood (protocols, algorithms, …)
- No satisfactory solutions yet for a number of specific instances
  - First Connect?
  - Local (user) authentication
  - Mobile CAPTCHA
  - Trustworthy installation
  - [Theft resistance and data/credential recovery]
  - ….
- One promising avenue: intuitive security/privacy policy configuration by using the context and history of the user's mobile device

How to make it possible to build trustworthy information protection mechanisms that are simultaneously easy-to-use and inexpensive to deploy while still guaranteeing sufficient protection?