Security For End Users

From personal devices to Internet of Things

N. Asokan
Kari Kostiainen
Cynthia Kuo

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

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
- Some usability challenges in securing Internet of Things (IoT)
- Conclusions

Why worry about usable security

Lack of security usability

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

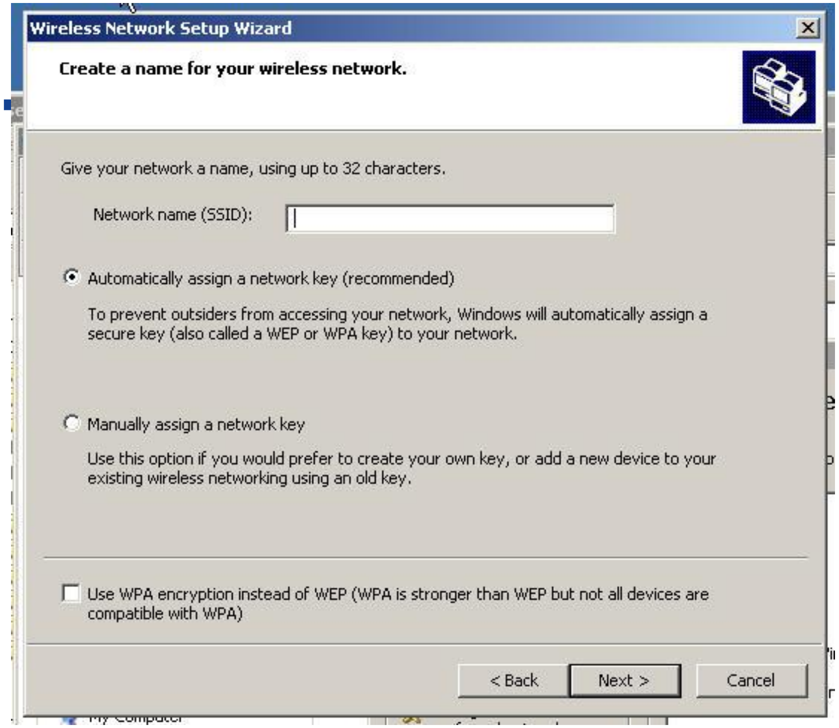
In many cases, the source of the ”cost” is surprising

Why?

Example: Setting up the first connection

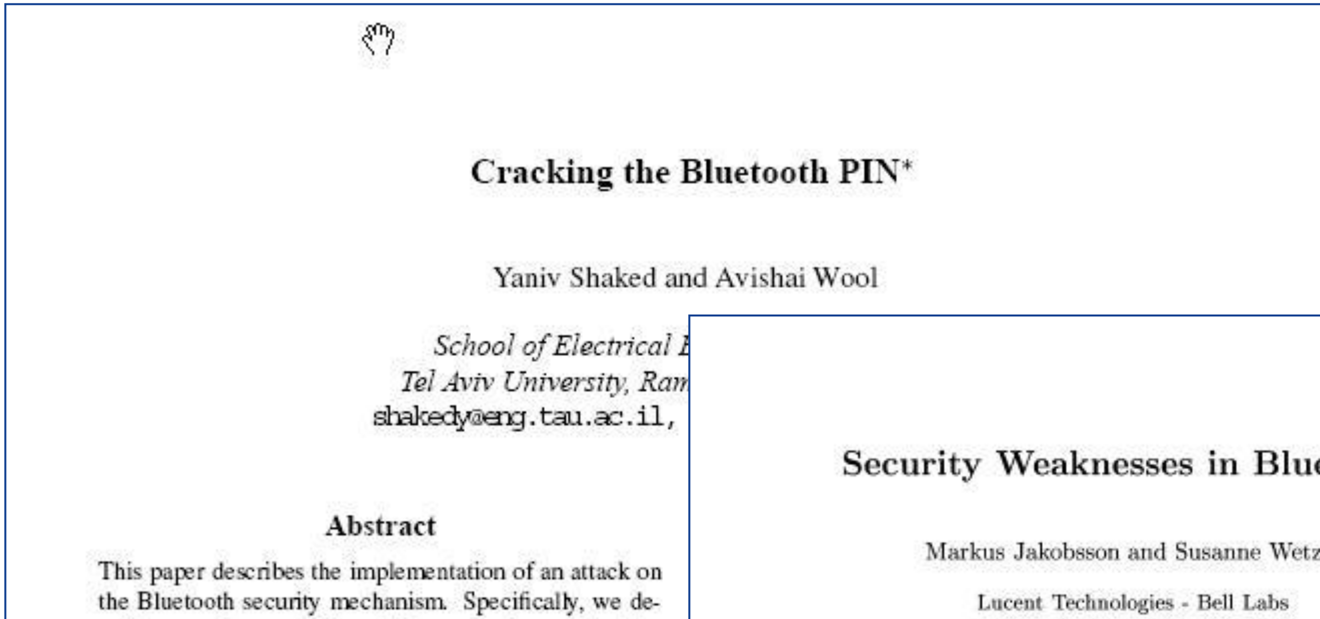
- **First Connect**: setting up contexts for subsequent communication.
  - Typically for proximity communications between personal devices, e.g.:
  - Pairing a Bluetooth phone and headset
  - Enrolling a Phone or PC in the home WLAN
  - More instances to come: Wireless USB, WiMedia
- **Problem (circa 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?

… and not very secure

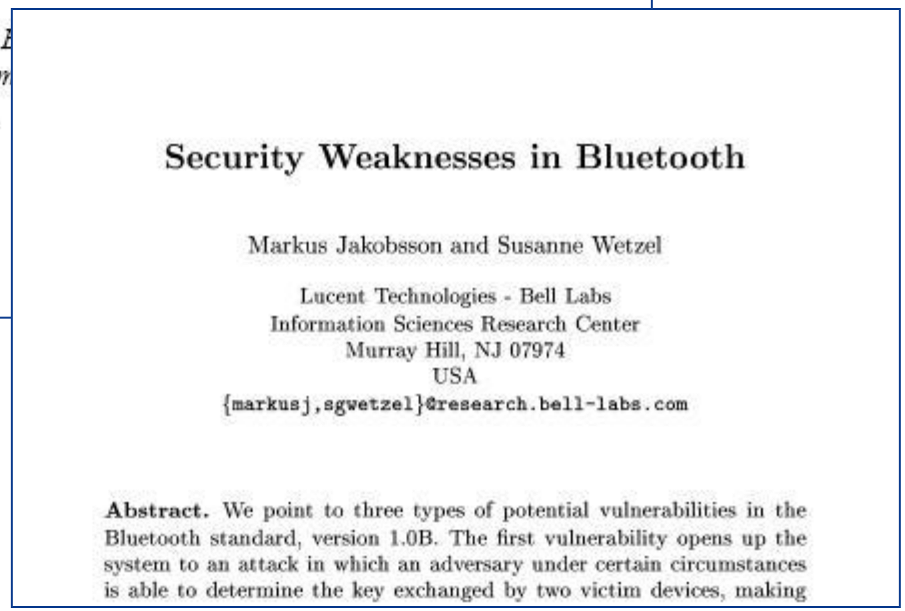

Cracking the Bluetooth PIN*

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!

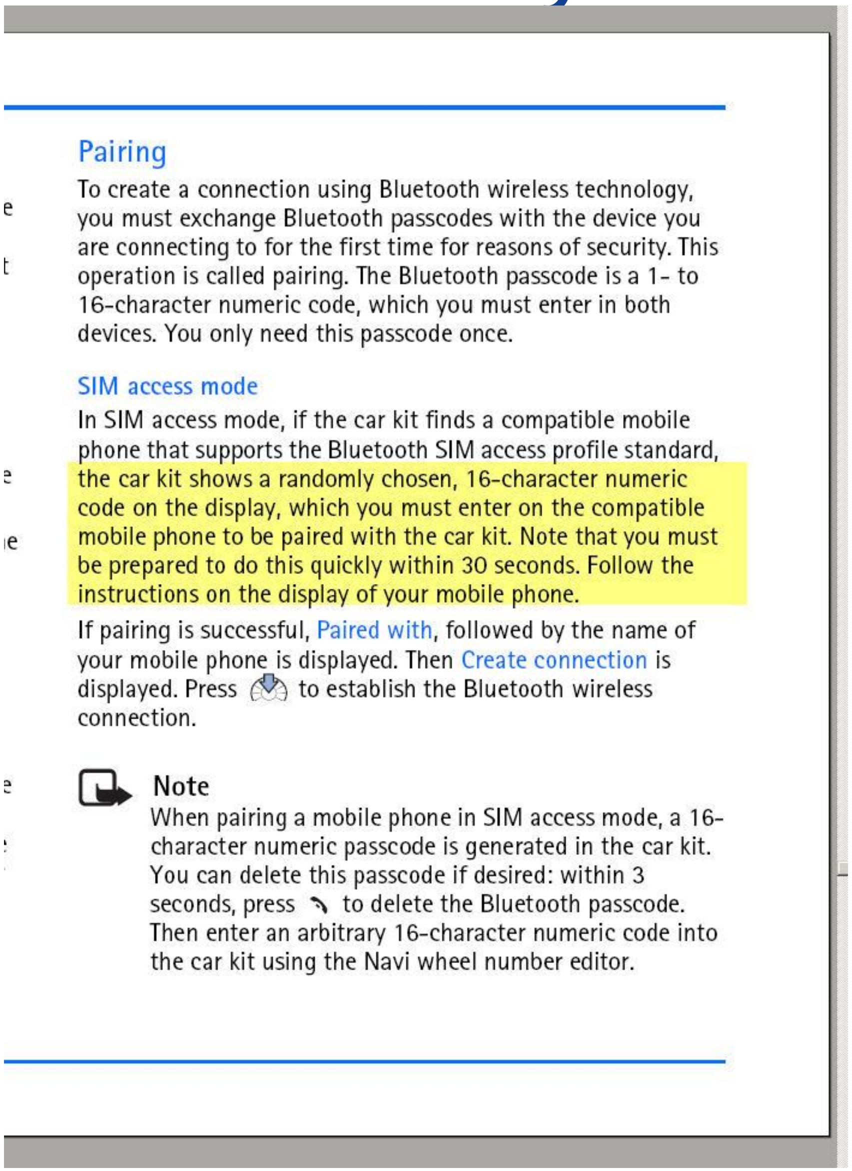
elsewhere as well. It is, therefore, absolutely essential that the password is changed immediately when the device is first installed."

"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

First Connect: background

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

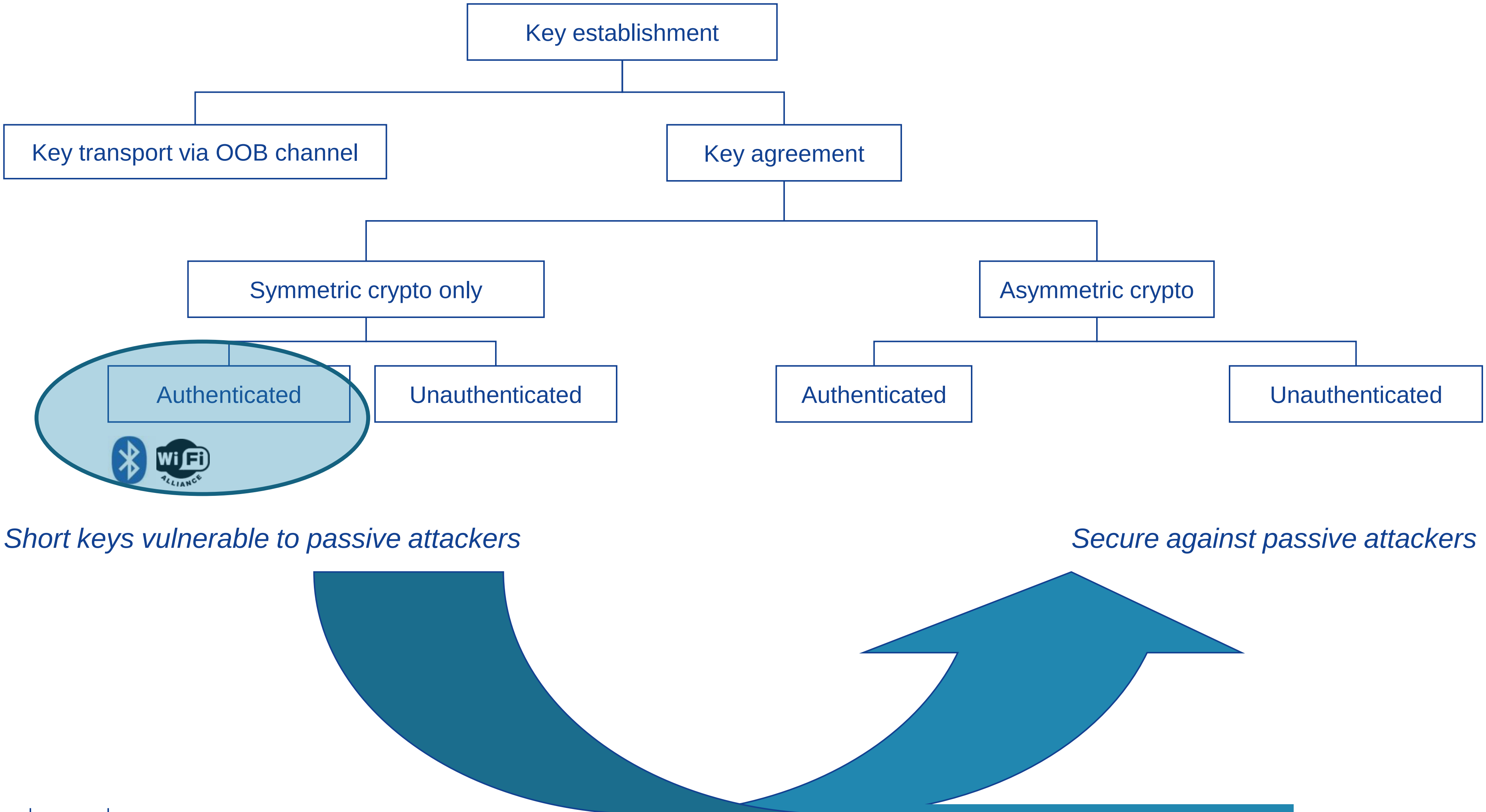
First Connect: background

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

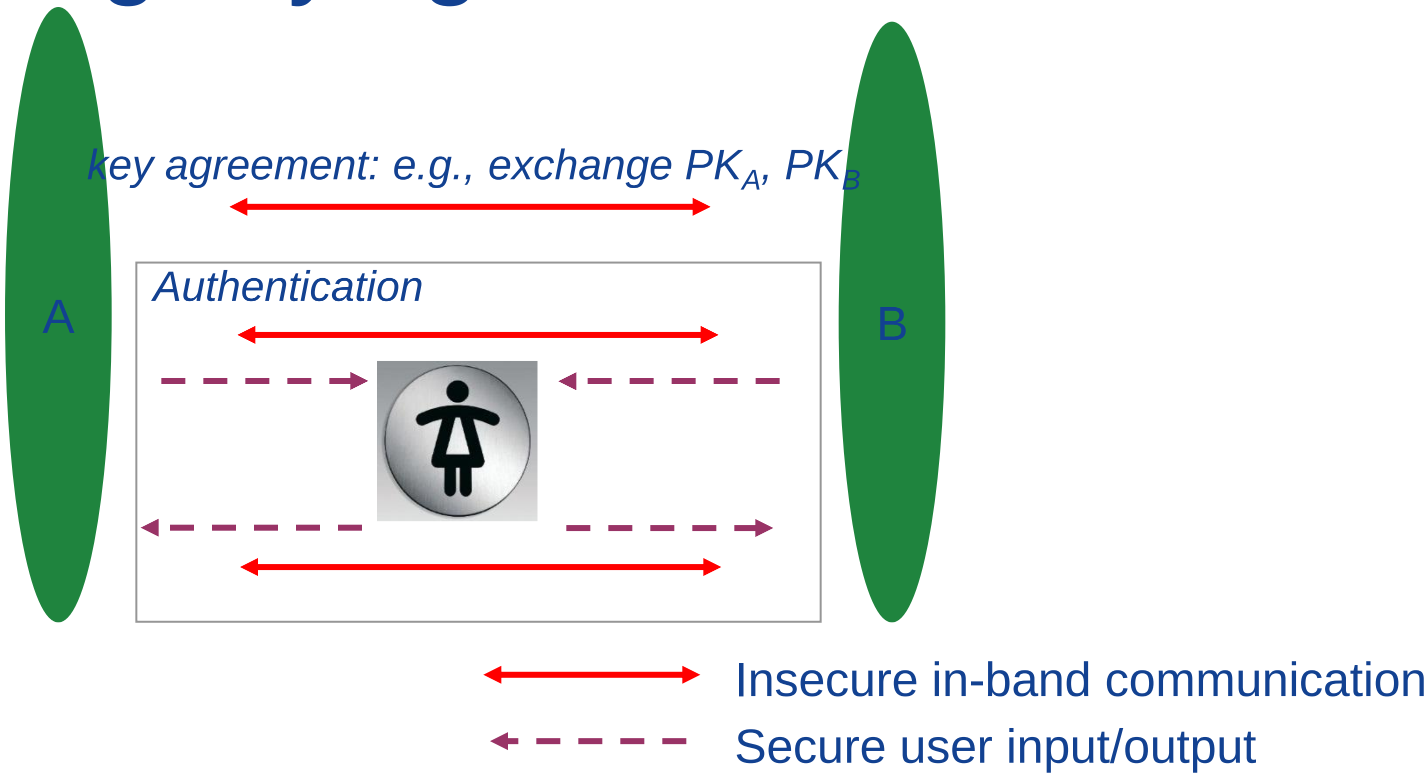


First Connect: background

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, …

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)

User as the secure channel

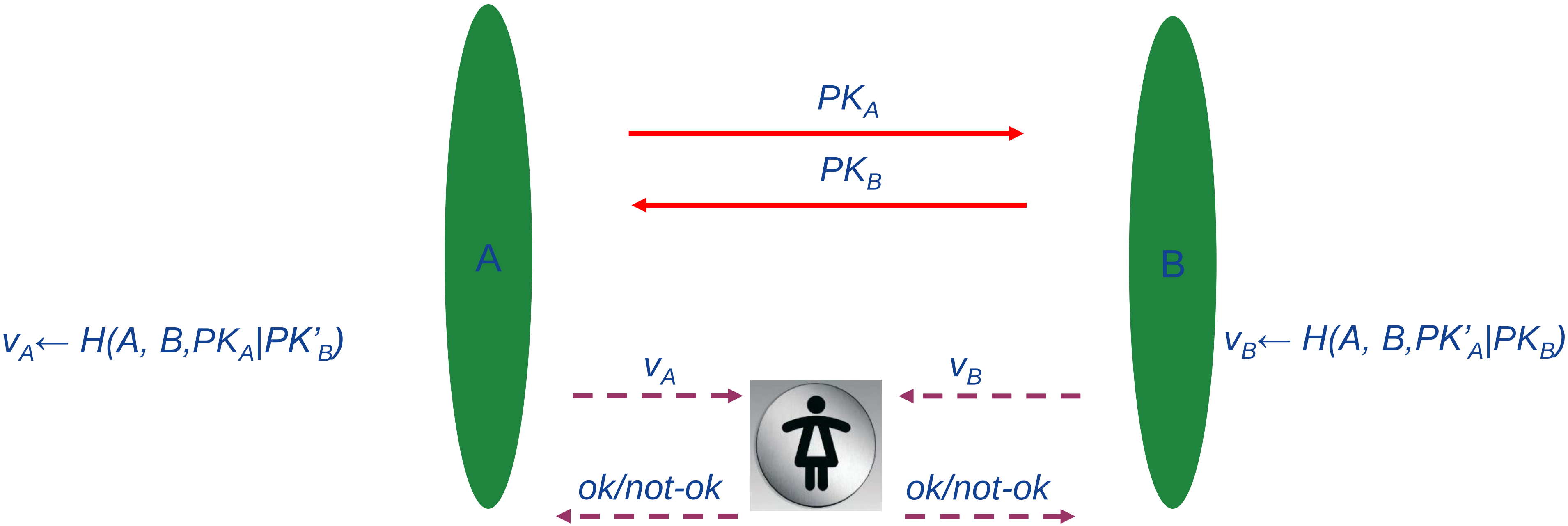
- Peer discovery by “user conditioning”: introduce a special first connect mode
  - E.g., Press a button to put device into the special mode
  - Demonstrative/indexical identification

- 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

First Connect: protocols in standards

Authentication by comparing short strings

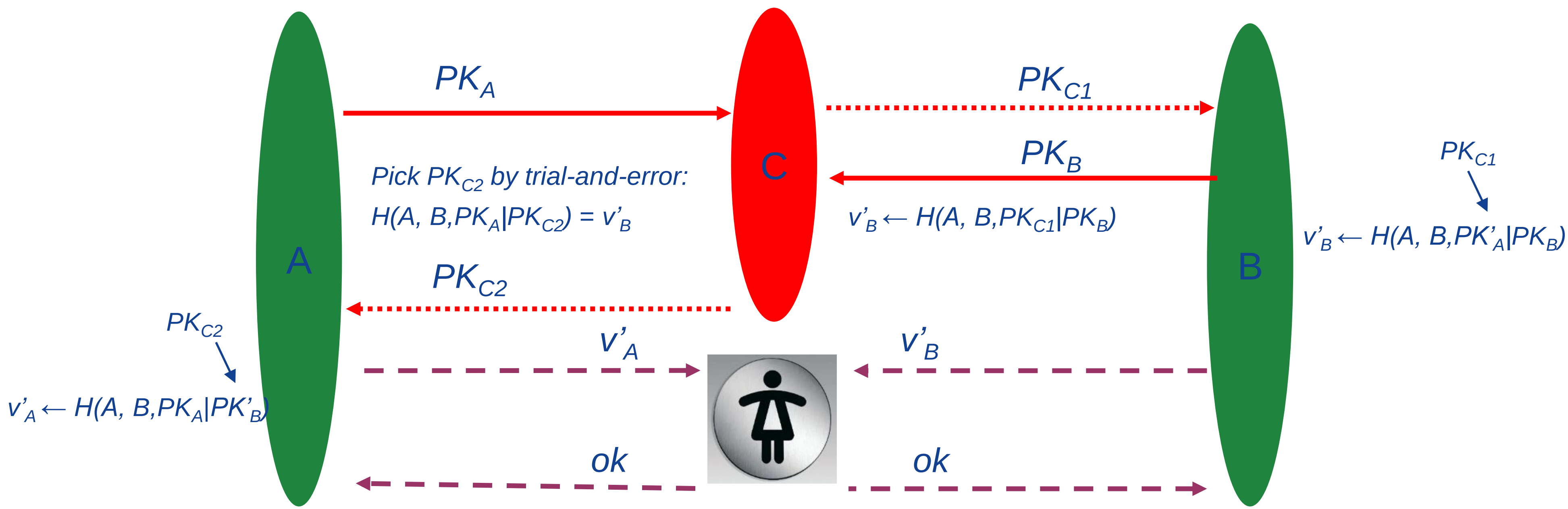


v_A and v_B are short strings (e.g., 4 digits),

User approves acceptance if v_A and v_B match

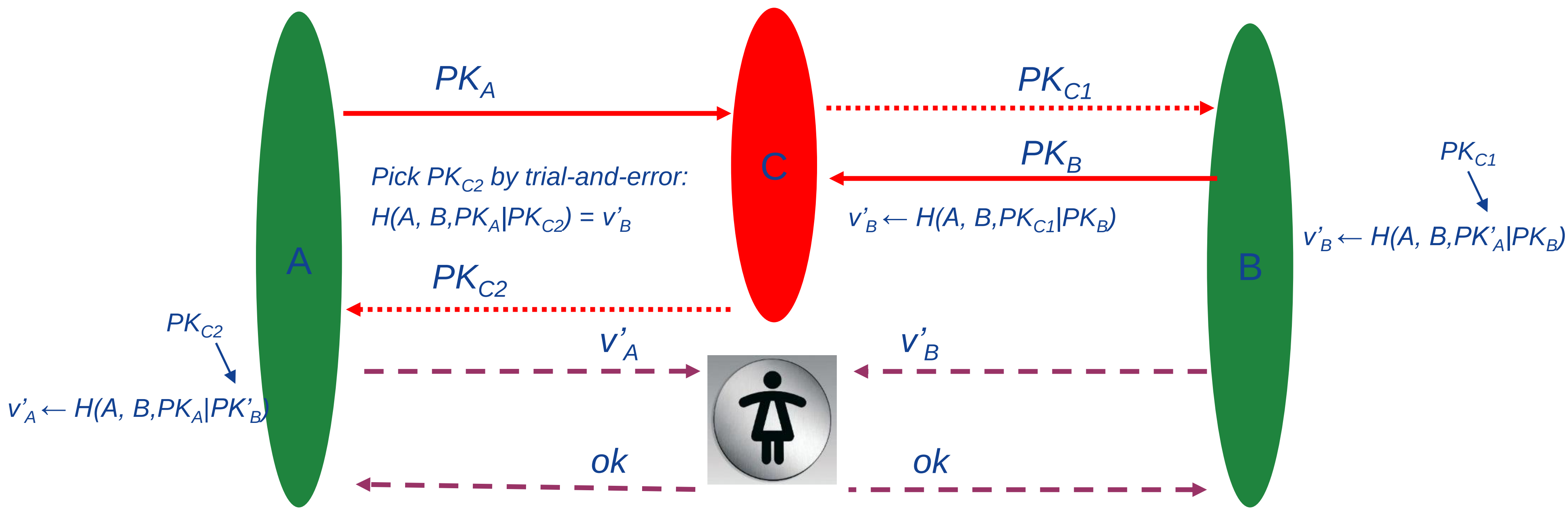
A man-in-the-middle can easily defeat this protocol

MitM in comparing short strings



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

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

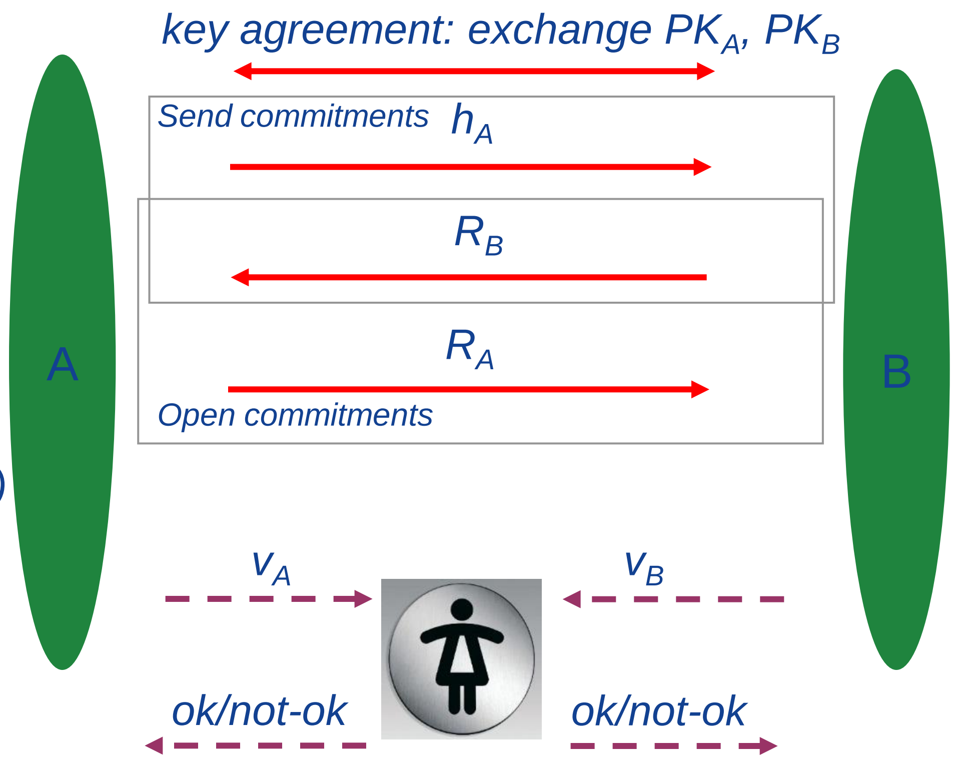
Authentication by comparing short strings

Choose long random R_A

Calculate commitment

$h_A \leftarrow h(A, R_A)$

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



Choose long random R_B

Verify commitment

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

Abort on mismatch

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

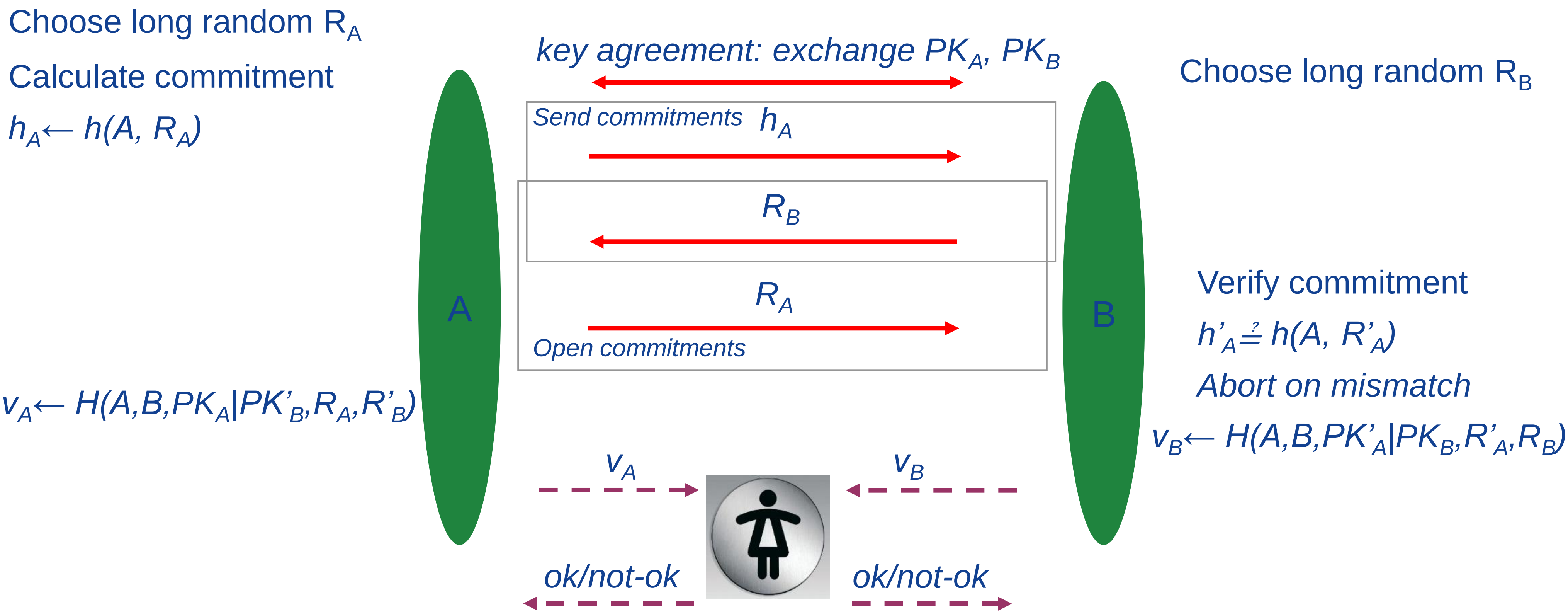
User approves acceptance if v_A and v_B match

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

Authentication by comparing short strings



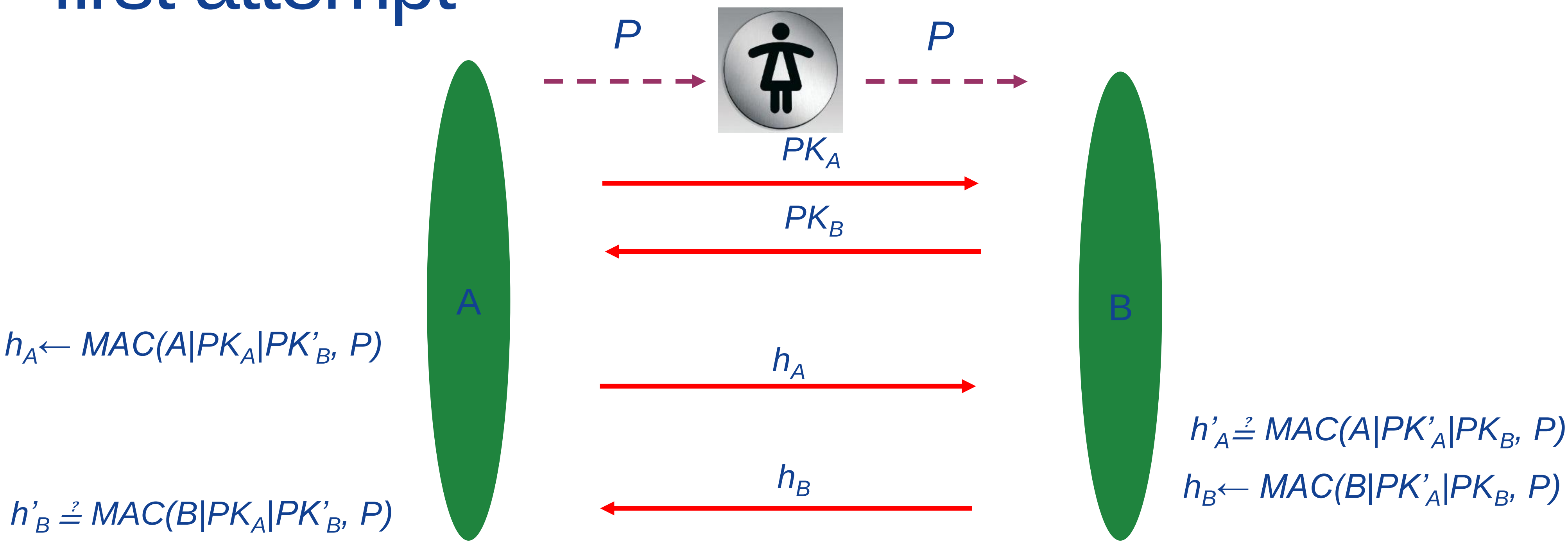
User approves acceptance if v_A and v_B match

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 a short passkey: a first attempt

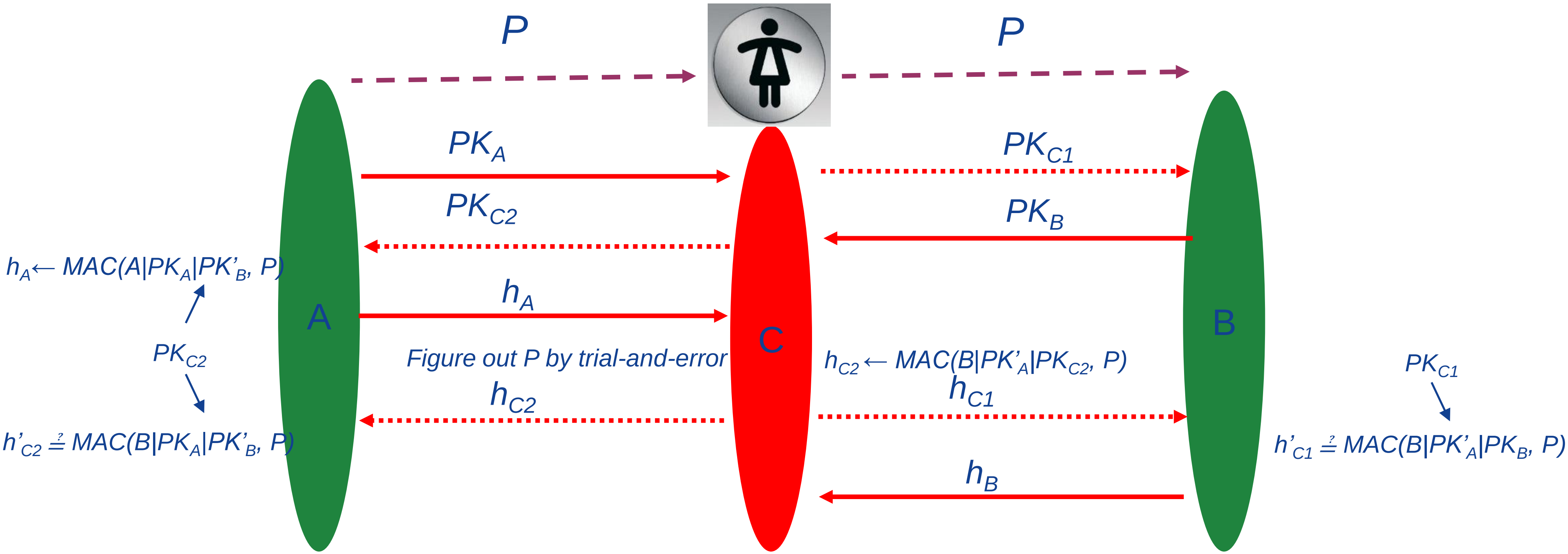


P is a short passkey (e.g., 4 digits)

MAC() is a message authentication code: e.g., HMAC-SHA1

But a man-in-the-middle can easily defeat this protocol!

MitM in using a short passkey



Guess a value x for P; calculate $h_x = MAC(A|PK'_A|PK_{C2}, X)$; Check $h_A \stackrel{?}{=} h_x$

If P is a n-digit PIN, attacker needs at most 10^n guesses; Each guess costs one MAC calculation

A typical modern PC can calculate over 1000000 MACs in 1 second

Authentication using interlocking short passkeys

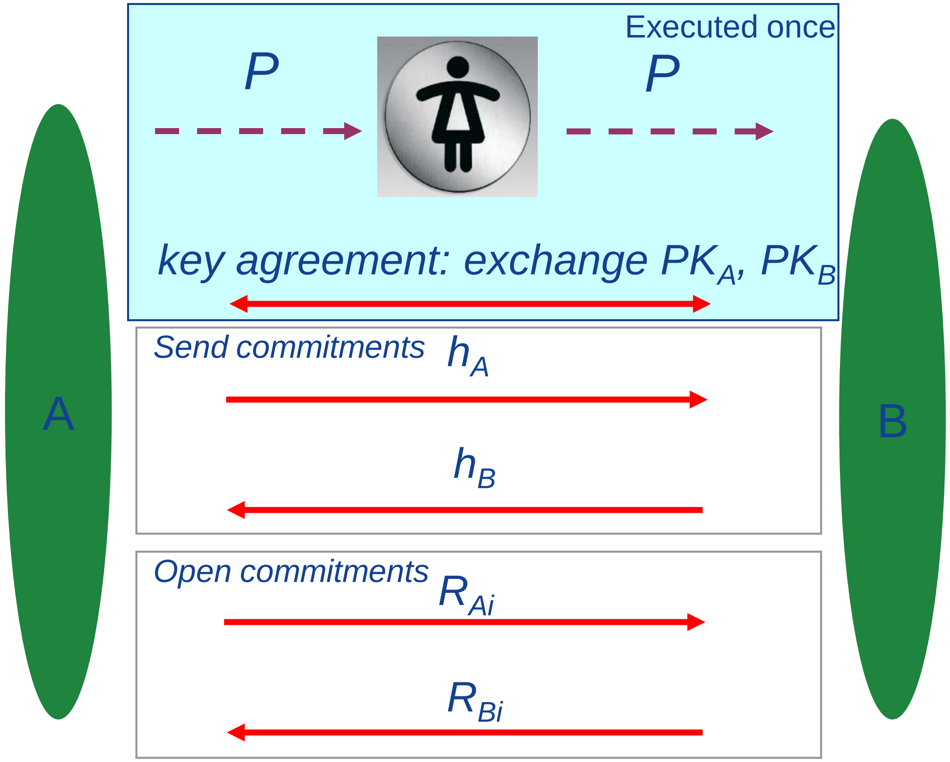
Choose long random R_{Ai}

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

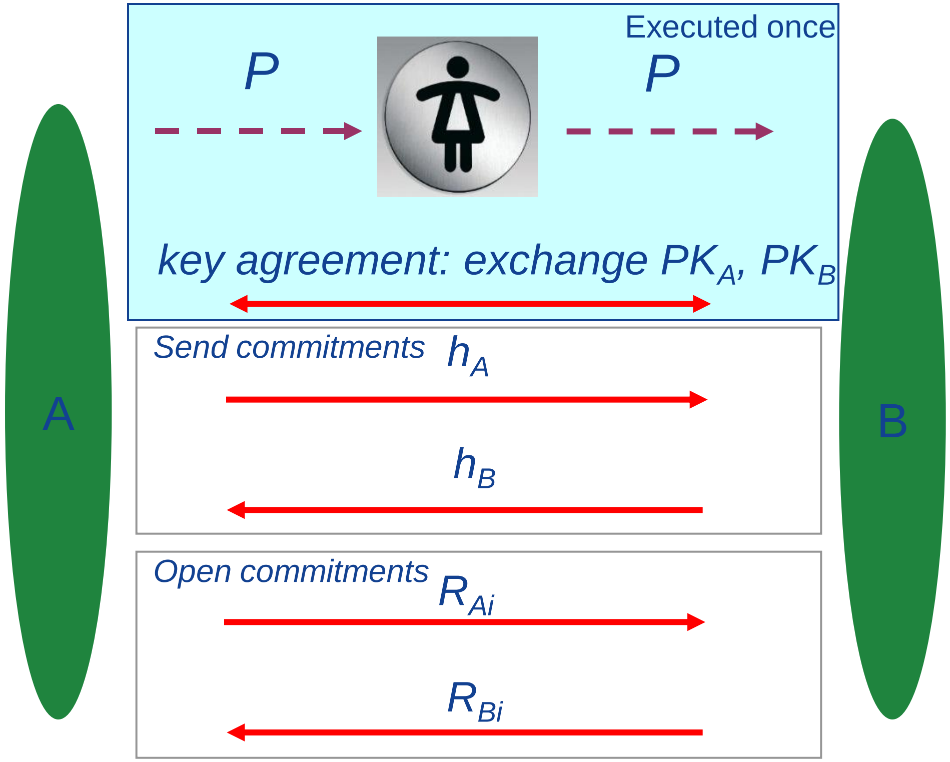
$h'_A \stackrel{?}{=} h(A, PK'_A|PK_B, Pi, R'_{Ai})$

One-time passkey *P* is split into *k* parts ($l \geq k > 1$): 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*)

Authentication using interlocking short passkeys



A:

Choose long random R_{Ai}

Calculate commitment

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

Verify commitment

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

B:

Choose long random R_{Bi}

Calculate commitment

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

Verify commitment

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

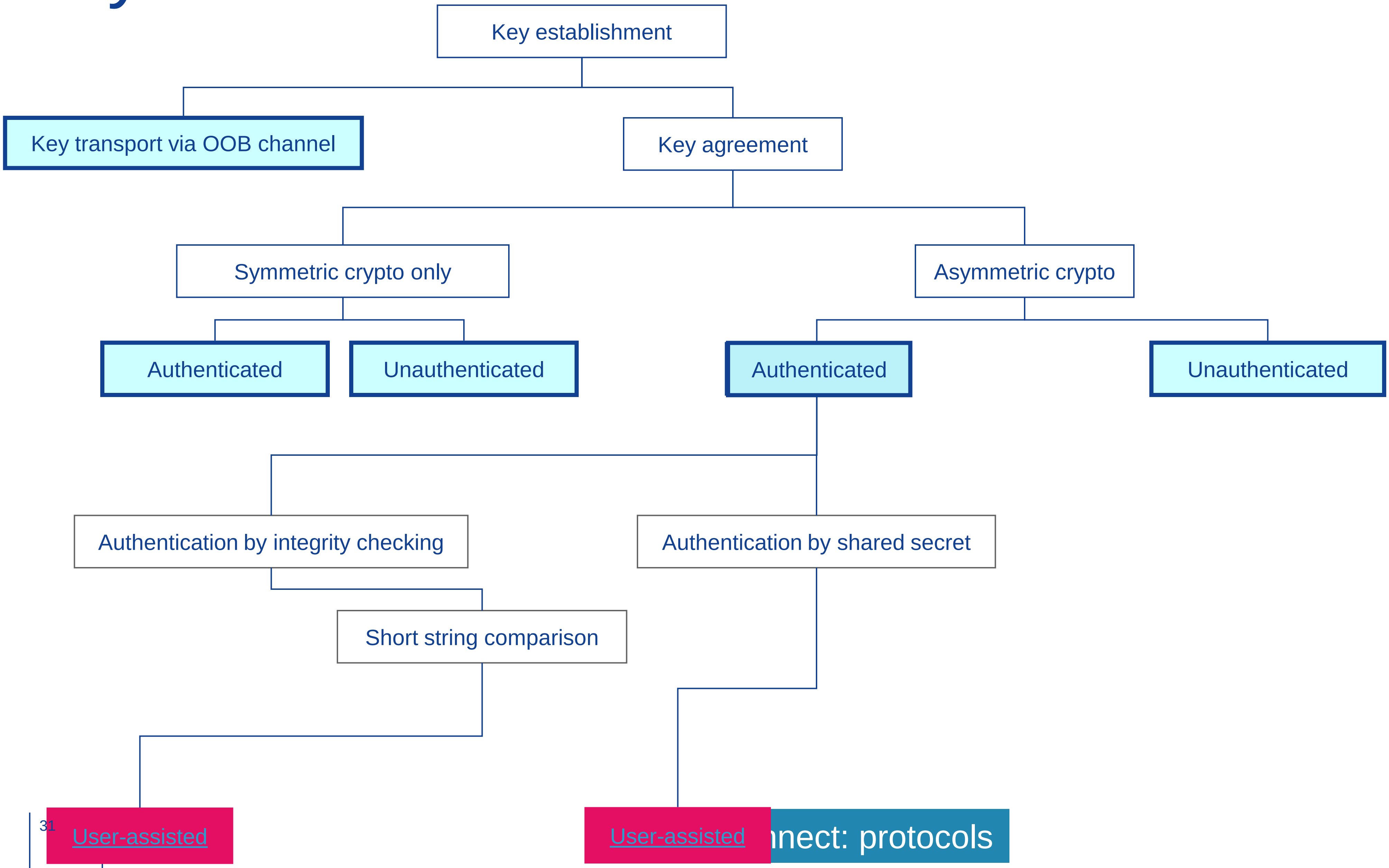
One-time passkey *P* is split into *k* parts (*k* > 1): next 4-round exchange repeated *k* times

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



Problems with user-as-secure-channel

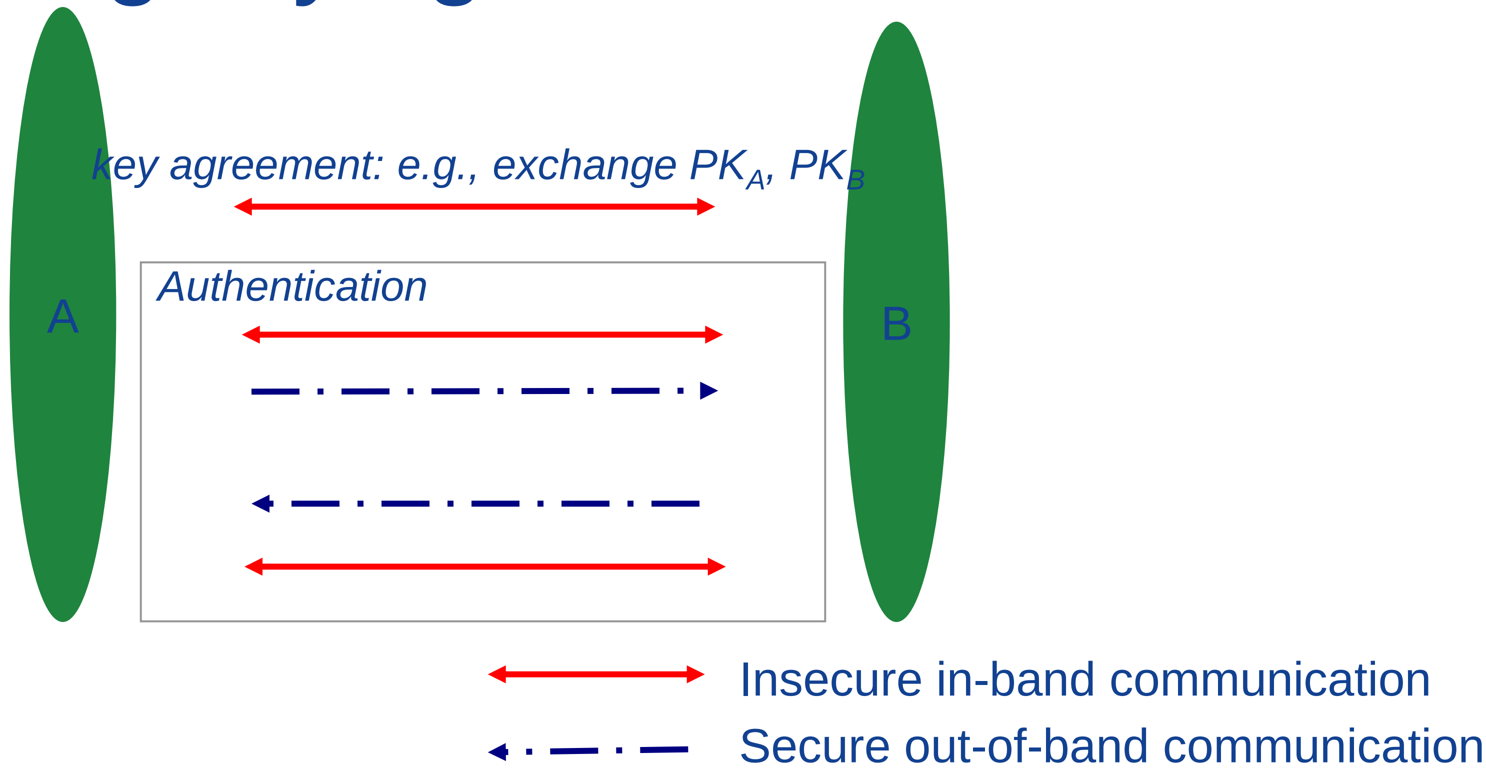
- Relies on availability of specific hardware (display, keypad, buttons, …)
- Needs a negotiation protocol
- What about usability?

Skip to “problems with OOB channels”

Out-of-band secure channel

- Idea: use a physically secure channel to transfer security critical information
  - Minimize user involvement → better usability, … and security
- Peer discovery is intuitive
  - Demonstrative/indexical identification
- Channel must have certain security properties
  - integrity (tampering with messages can be detected)
  - Sometimes secrecy as well

Authenticating key agreement: out-of-band channel



Different out-of-band channels have different

- Bandwidth
- Directionality (1-way or 2-way)
- Security properties (integrity-only, or integrity+secrecy)

First Connect: protocols in standards

What OOB channels can you think of?

- Near Field Communication
  - “touch” to connect

- Audio

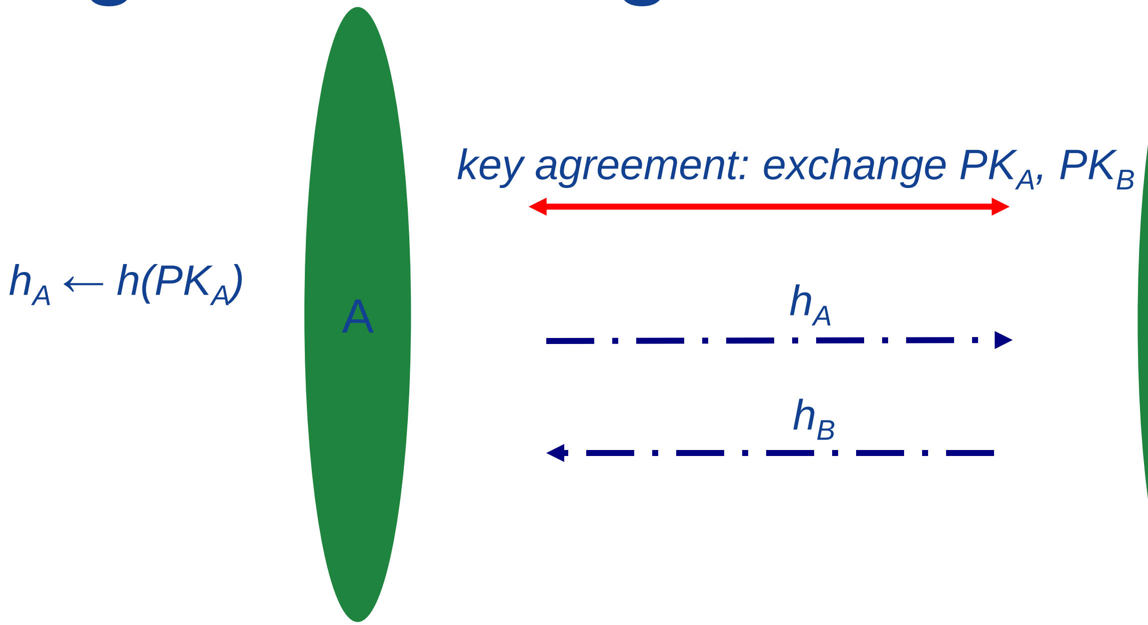
Visual Channel with minimal additional hardware?

- Visual

- Body-area communica
  - *touch* to connect
- …

First Connect: protocols in research papers

Seeing Is Believing



McCune et al,
[IEEE S&P 2005]

Rohs, Gfeller
[PervComp'04]

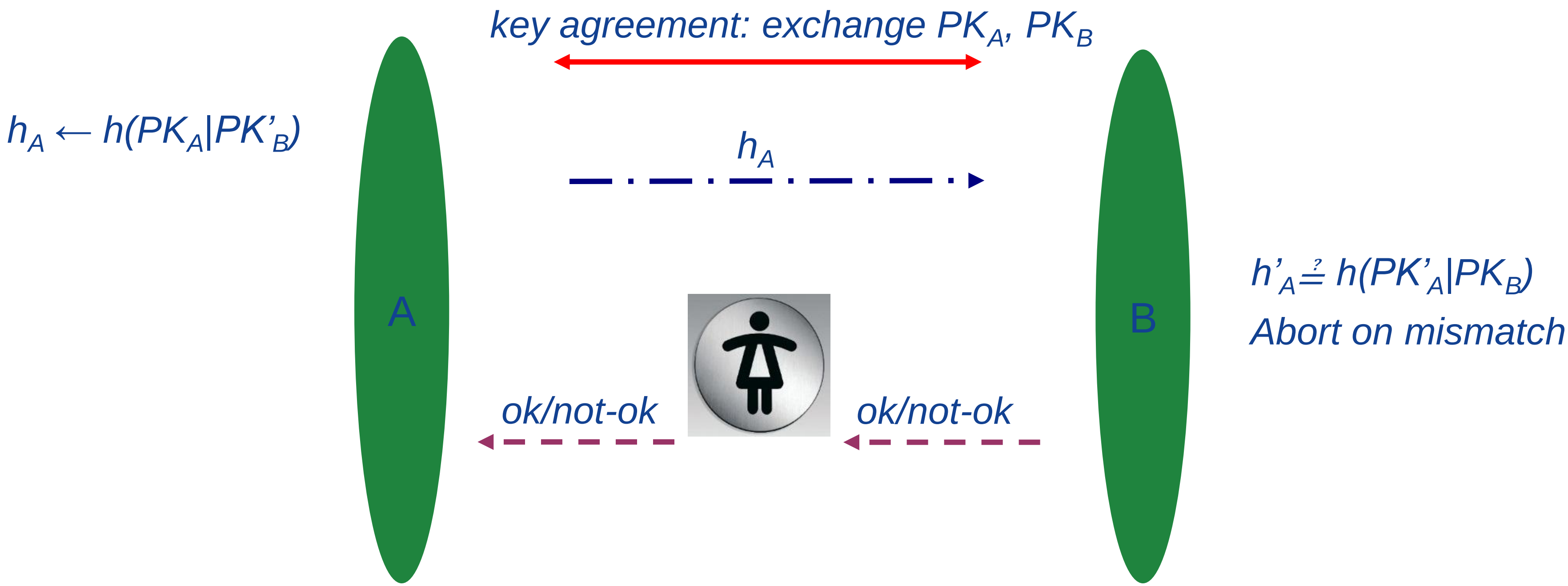
First Connect: protocols in research papers

Drawbacks of SiB

1. Mutual authentication requires that <u>both</u> devices have cameras and switch roles
   ➔ Slow and difficult for the user!
   Potential solution: one-way visual channel + user confirmation
2. Not all devices have big enough displays to show two-dimensional bar codes
   - Typically these constrained devices do not have cameras either

Problem: secure first connect for constrained devices with **minimal additional hardware**?

Mutual authentication with one-way visual channel



Supporting display constrained devices

Use a short authentication string protocol like MANA IV

Choose long random R_A

$h_A \leftarrow h(A, R_A)$

key agreement: exchange PK_A, PK_B

h_A

R_B

R_A

v_A

Choose long random R_B

A

B

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

Abort on mismatch

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

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

Check $v'_A \stackrel{?}{=} v_B$ *show ok/not-ok*

Abort if $v'_A \neq v_B$

ok/not-ok

ok/not-ok

Saxena, Ekberg, Kostiainen, Asokan [IEEE S&P 2006]

First Connect: protocols in research papers

Supporting display constrained devices

Pairing phone and laptop with LED

Pairing two phones

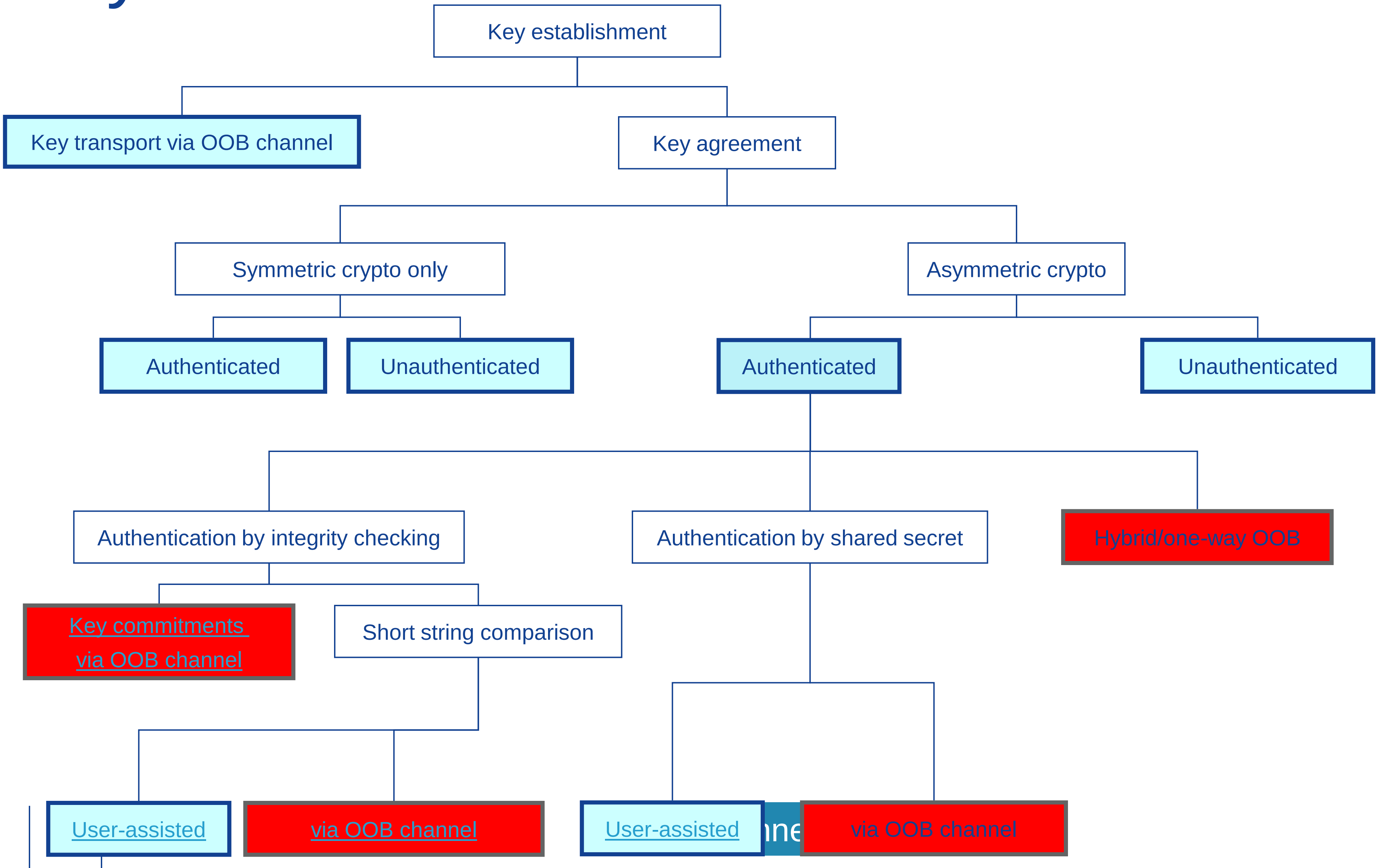
Suitable for access points, wireless headsets

Hardware needed:

- Single LED (cheap)
- Video camera (common on smartphones)

Saxena, Ekberg, Kostiainen, Asokan [IEEE S&P 2006]

Key establishment for first connect



Problems with out-of-band channels

- Cost
  - Availability of specific (possibly new) hardware interfaces
- Deployability
  - Universally deployed auxiliary channel needed
  - Else how to discover common aux. channels between devices?
    - Leave-it-to-the-user: visible well-known logos
    - Negotiation protocol

Can we use the radio interface itself for authentication?

- In-band integrity checking
  - Assumption: genuine device emits energy during transmission; a distant attacker cannot easily drown this out
  - I-codes by Čagalj et al
- Common radio environment
  - Assumption: genuine devices hear the same radio signals; a distant attacker likely hears something different
  - Amigo by Varshavsky et al
- Spatial indistinguishability
  - Assumption: a distant attacker cannot tell which device is transmitting
  - Shake-them-up by Castelluccia et al

Integrity protection in-band: I-Codes

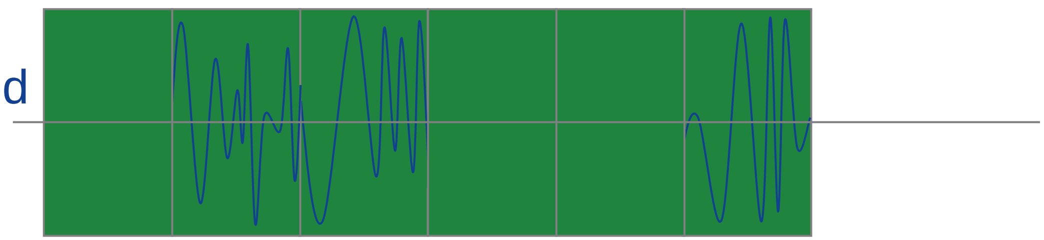
Message 0 1 0 …

↓ Manchester coding

Encoded message 0 1 1 0 0 1 …

↓ On-off keying

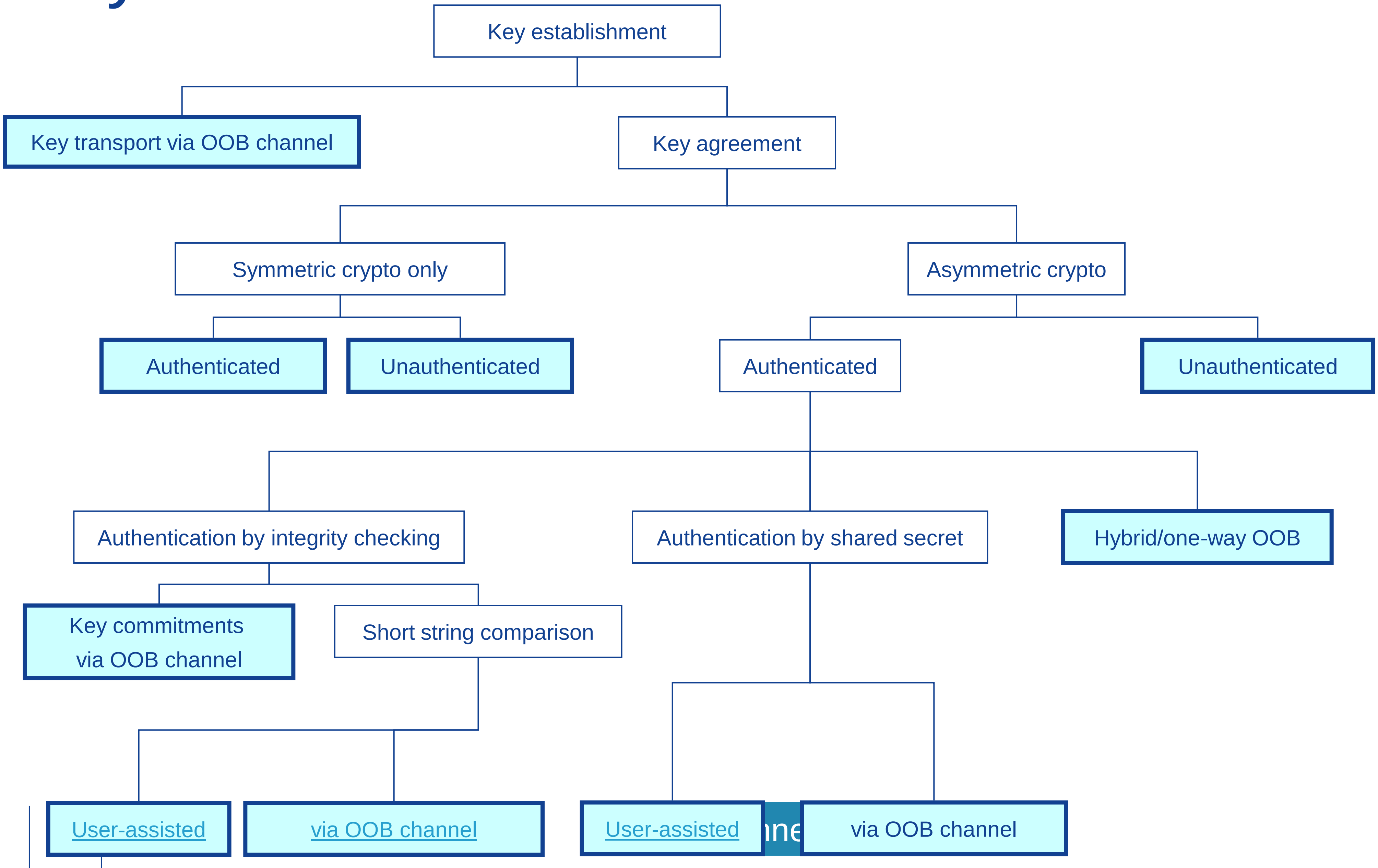
Transmitted signal



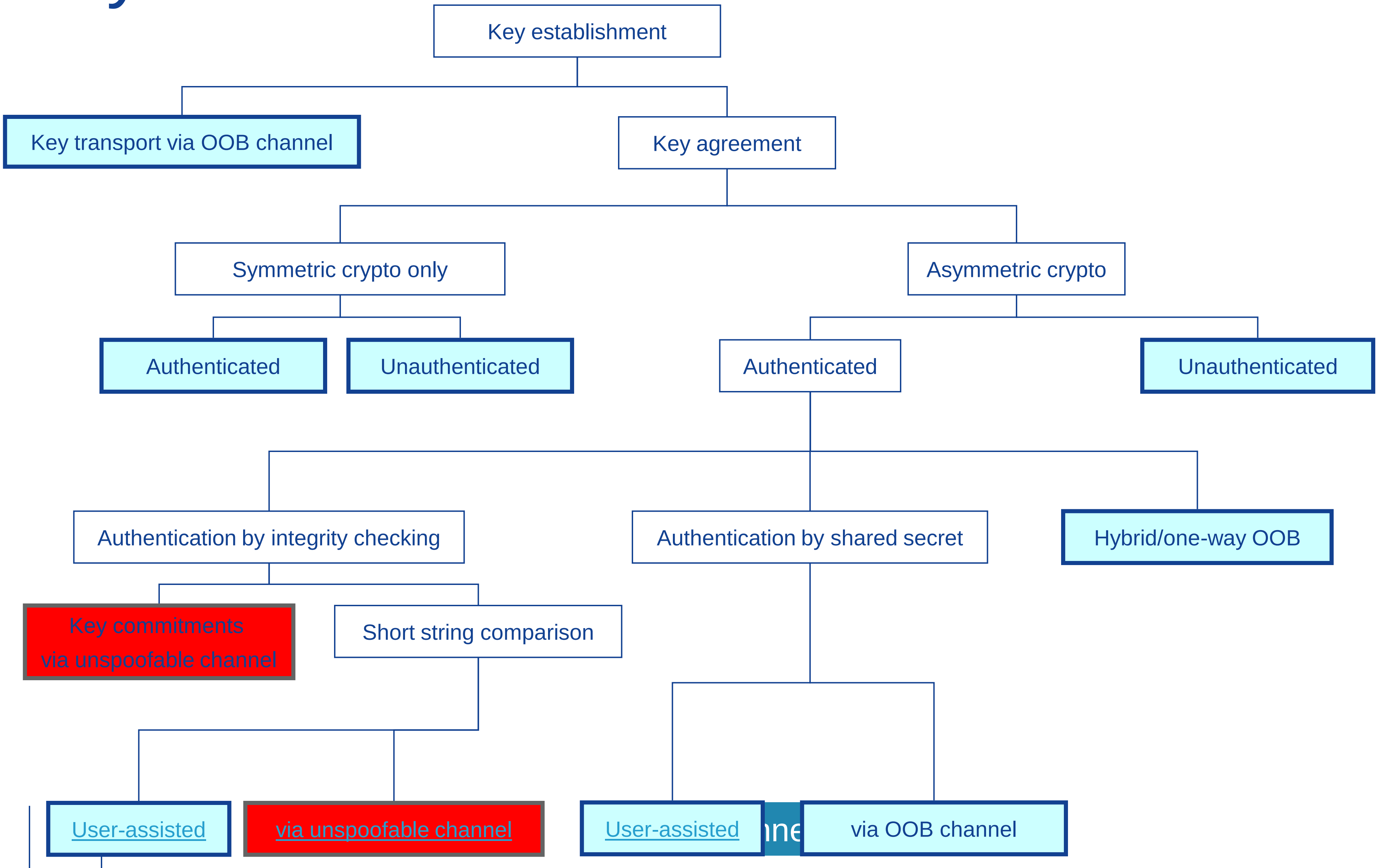
- Recipient measures the presence/absence of energy (1-bit/0-bit)
- Attacker cannot change 1→0
- Issues
  - Modifications to lower layers in the communication stack
  - No genuine radio interference

Čagalj, Čapkun, Rengaswamy, Tsigkogiannis, Srivastava, Hubaux [IEEE S&P 2006]

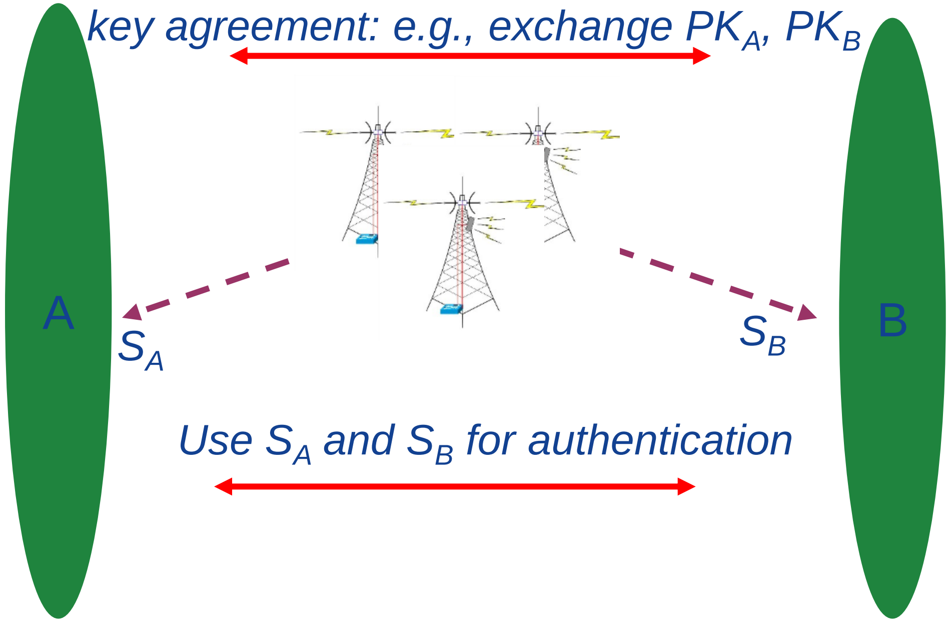
Key establishment for first connect



Key establishment for first connect



Authenticating key agreement: secret extraction from common environment



Sensing common private environment

- Measure some environmental features
    - For co-located (in space and time) sensors measurements should be *almost* identical
    - For anyone else, measurement must be unpredictable
- Radio signal strength [Varshavsky, Scanneli, LaMarca, de Lara, HotMobile 2007, UBICOMP 2007]
- Accelerometer readings [Mayrhofer and Gellersen, Pervasive 2007, TMC 2009]

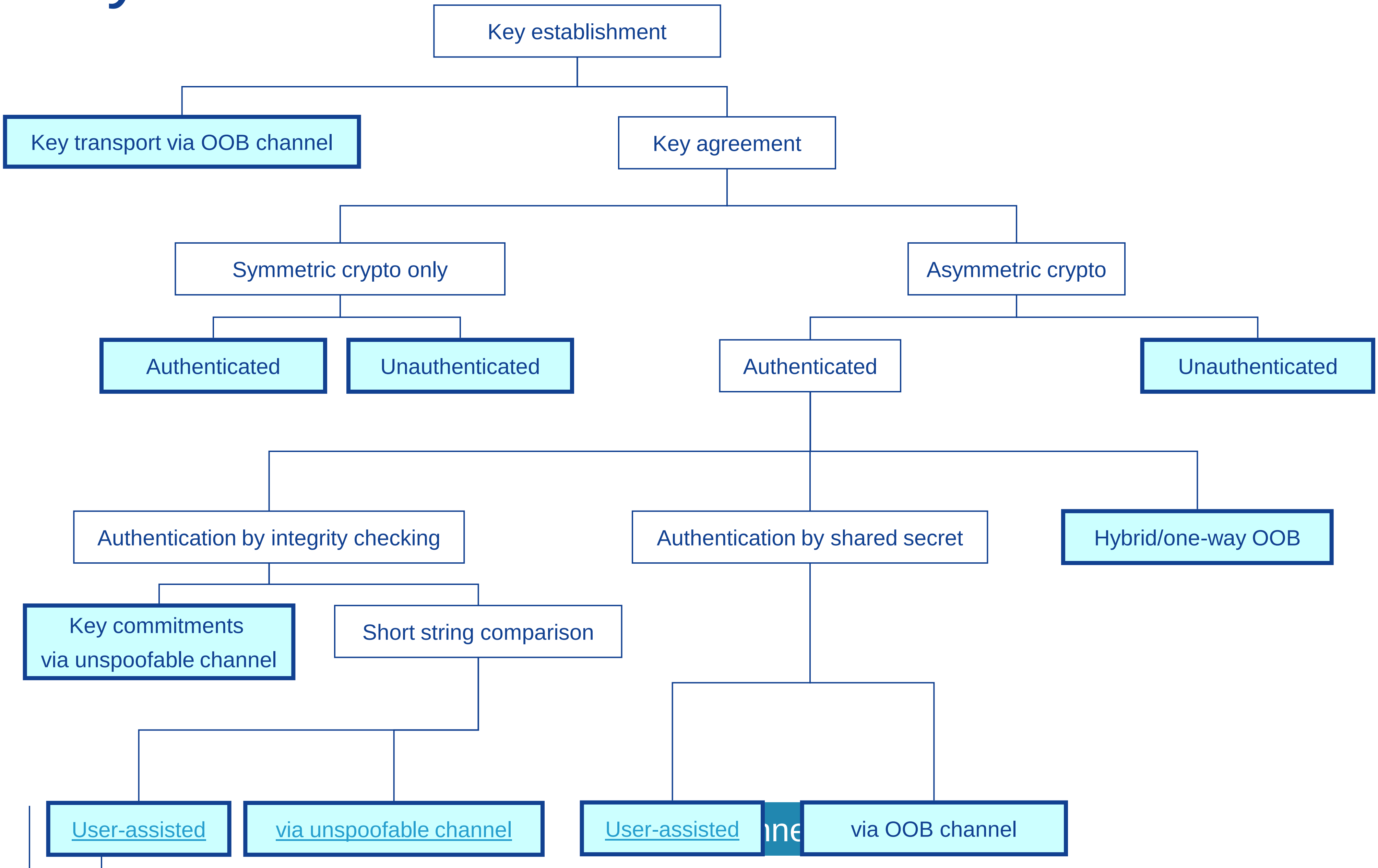
First Connect: protocols in research papers

Issues with secret extraction

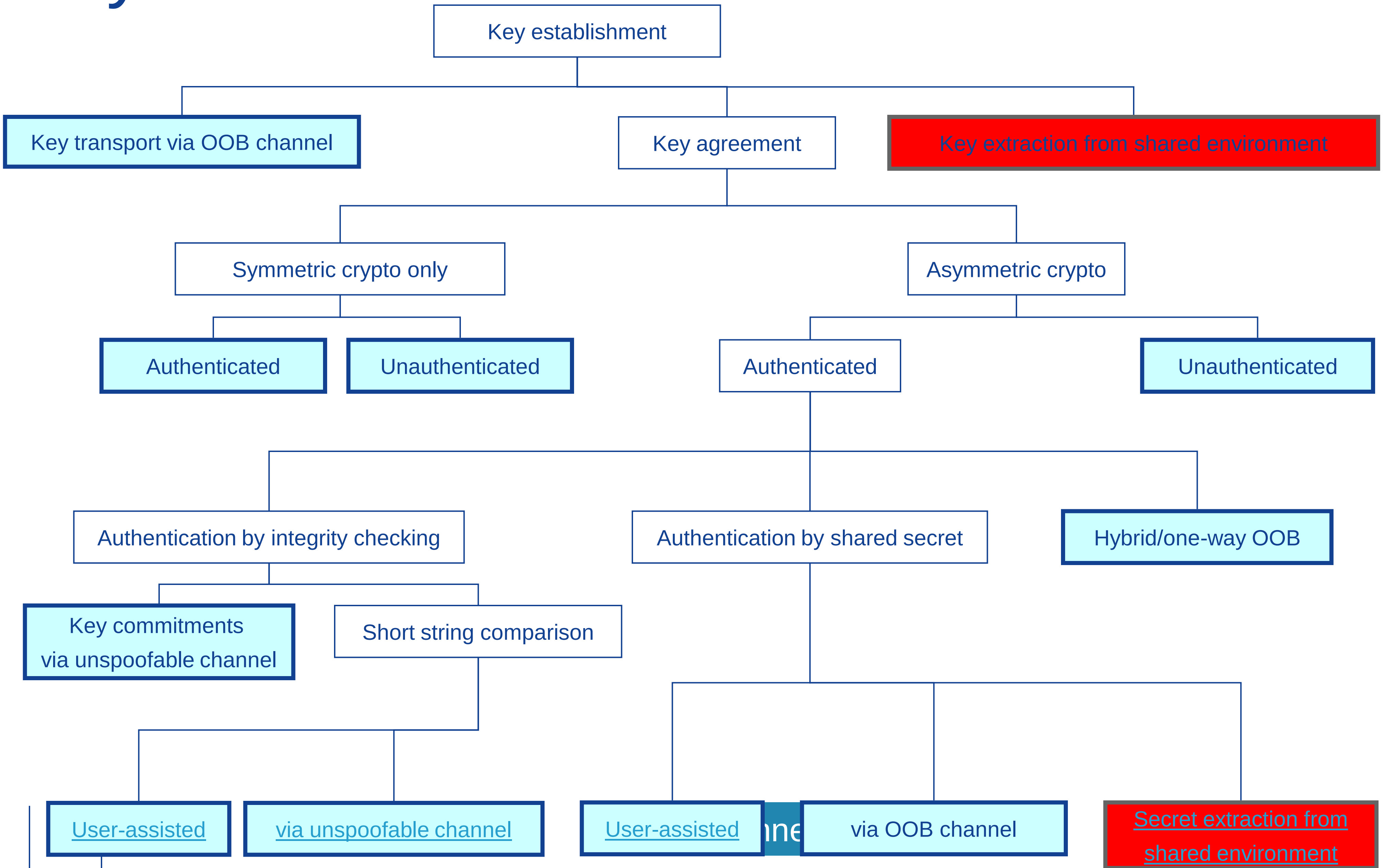
- User involvement
- Are the assumptions valid?

- If a long shared secret can be extracted, is key agreement still necessary?

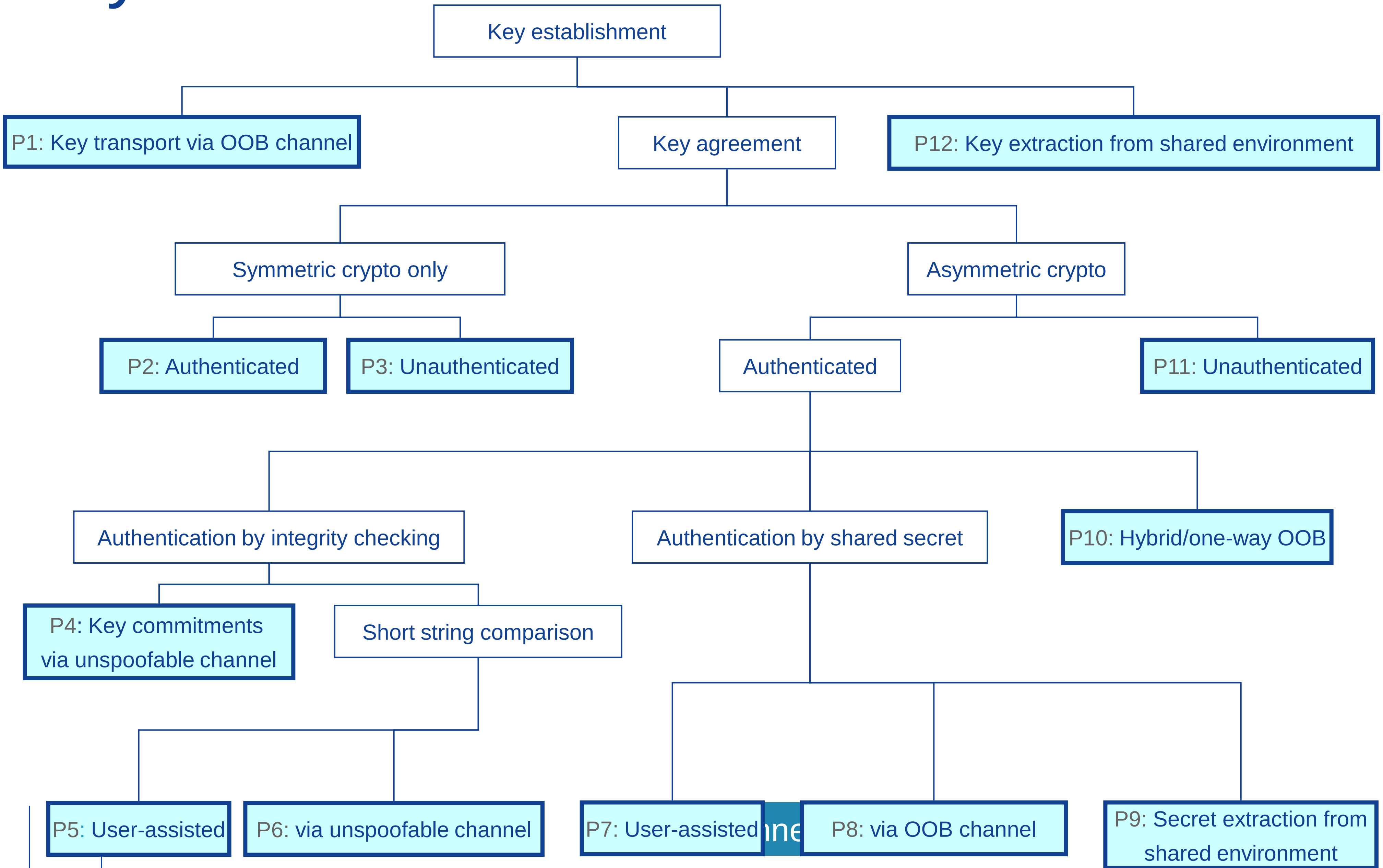
Key establishment for first connect



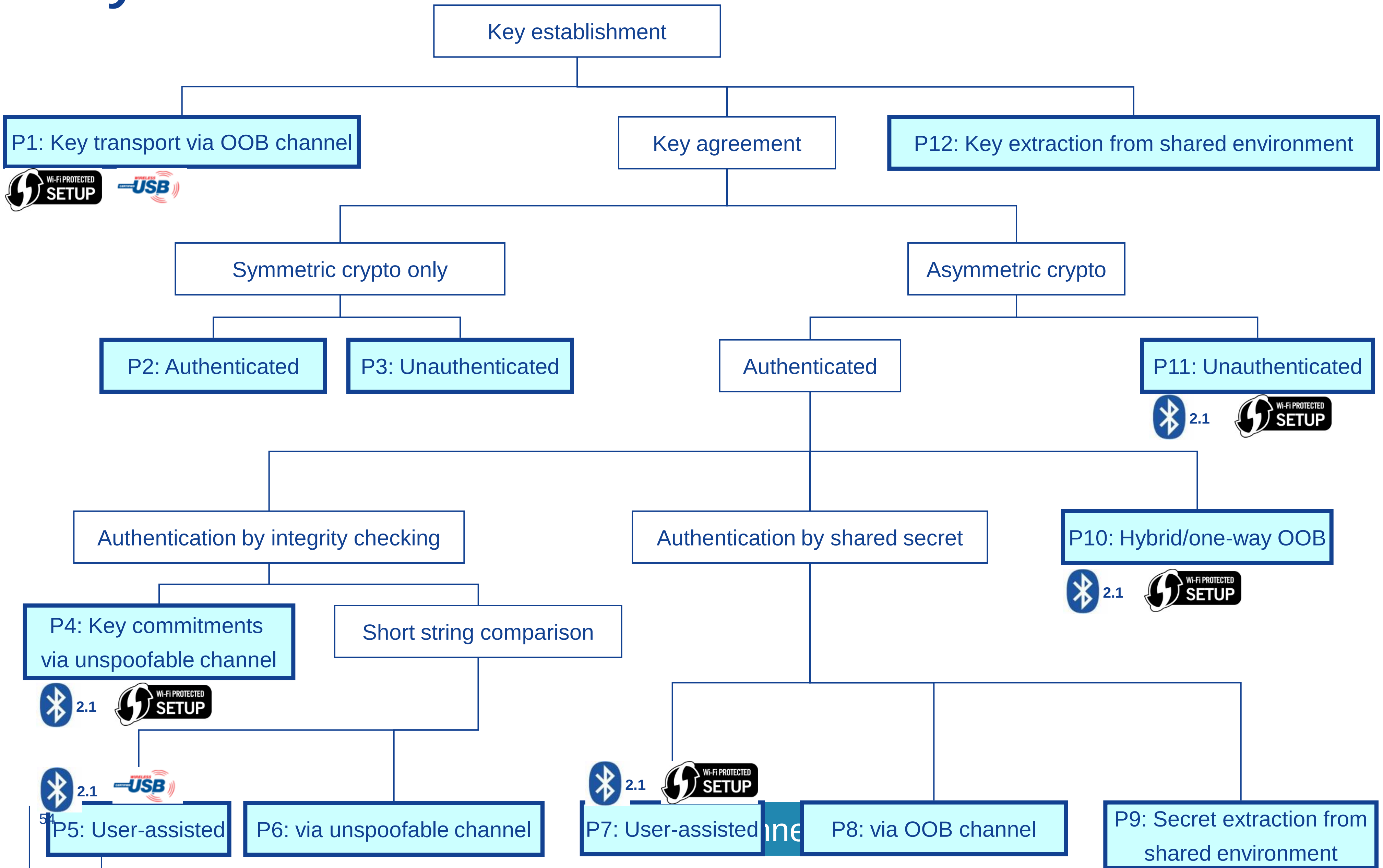
Key establishment for first connect



Key establishment for first connect



Key establishment for first connect ~2008



Key establishment for first connect ~2008

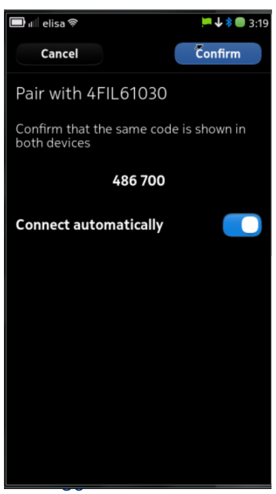
	Unauthenticated Diffie-Hellman	Authenticated Diffie-Hellman		
		short-string comparison	short PIN	Out-of-band channel
WiFi Protected Setup	“Push-button”		√	NFC
Bluetooth 2.1	“Just-works”	√	√	NFC
Wireless USB		√		USB Cable

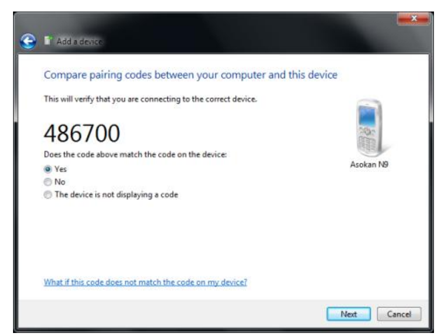
“Security associations for wireless devices” (Overview, book chapter)

“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





First Connect: status

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)

First Connect: status

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.

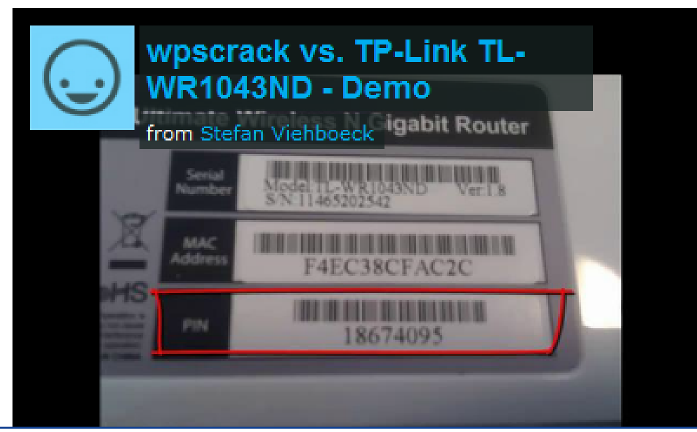
Detailed information about this vulnerability can be found in this paper: **Brute forcing Wi-Fi Protected Setup** – Please keep in mind th affected devices.

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 Solutio 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)



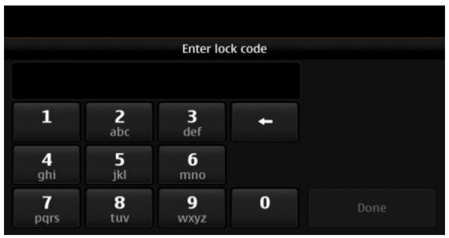
http://sviehb.wordpress.com/2011/12/27/wi-fi-protected-setup-pin-brute-force-vulnerability/
http://www.kb.cert.org/vuls/id/723755

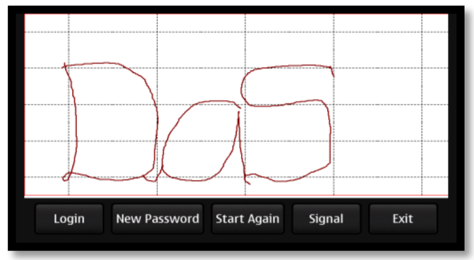
First Connect: status

Break

Local user authentication: need new methods

[SOUPS '10 paper]





Need alternatives that are:

- Faster
- More enjoyable
- Secure enough

Biometrics

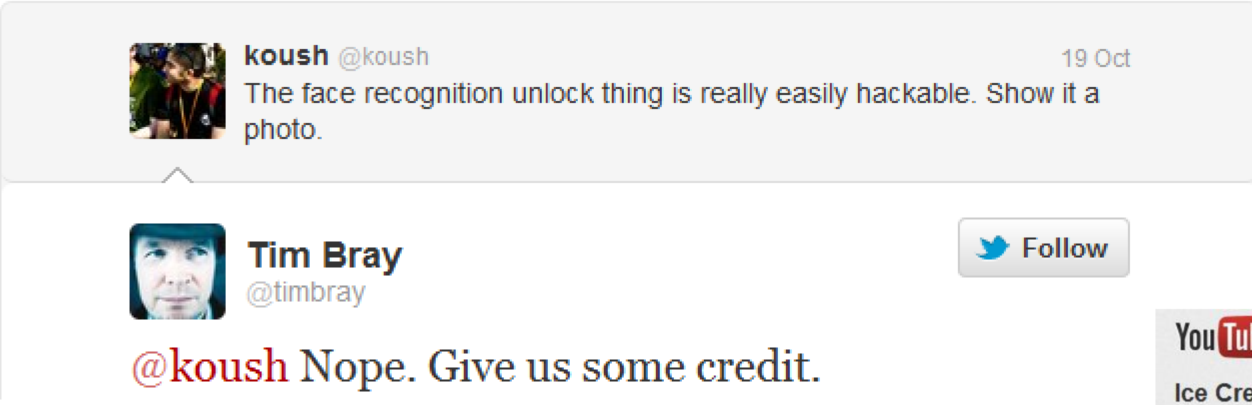
Wearables

?

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

Cost: weak PINs

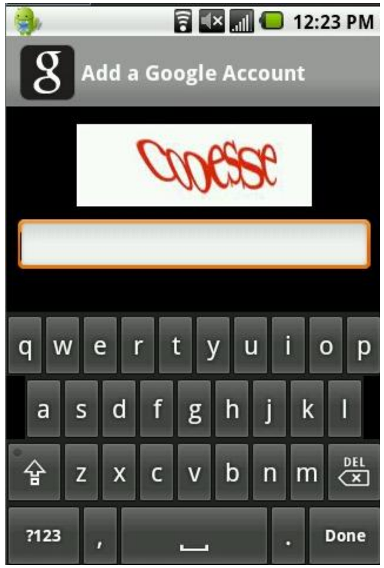
Local user authentication: a cautionary tale

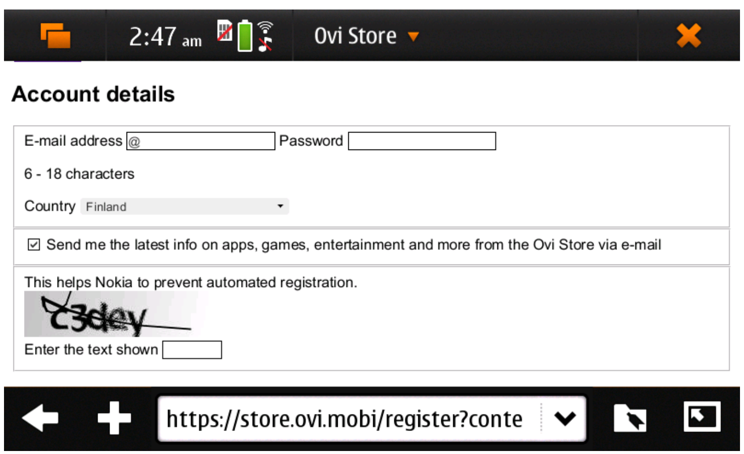


http://youtu.be/BwfYSR7HttA

Local authentication

CAPTCHA on mobile devices





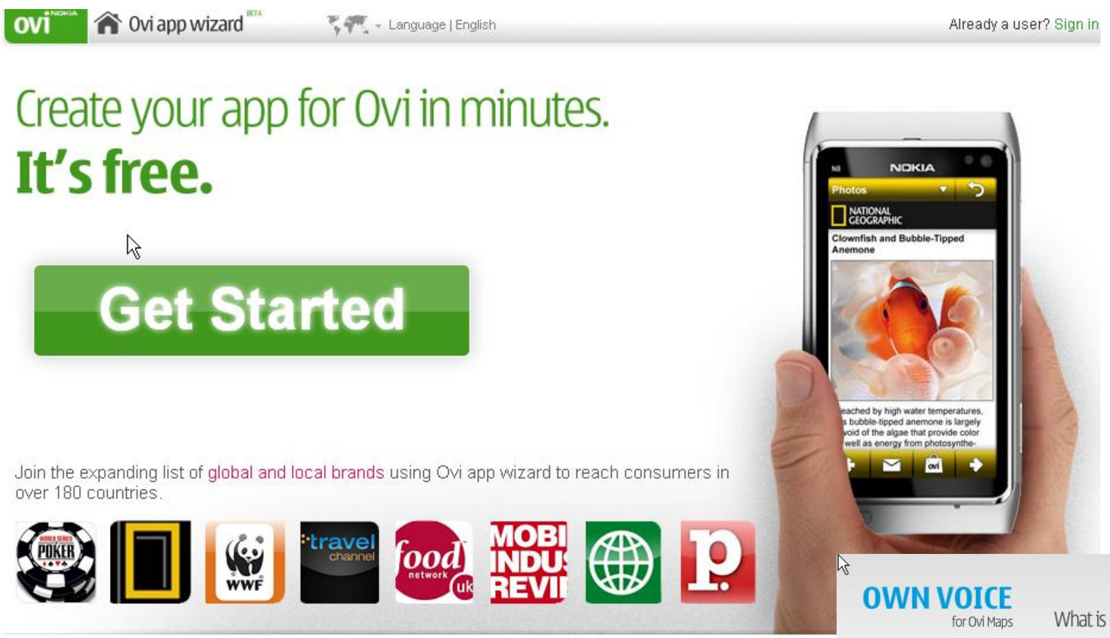
Cost:

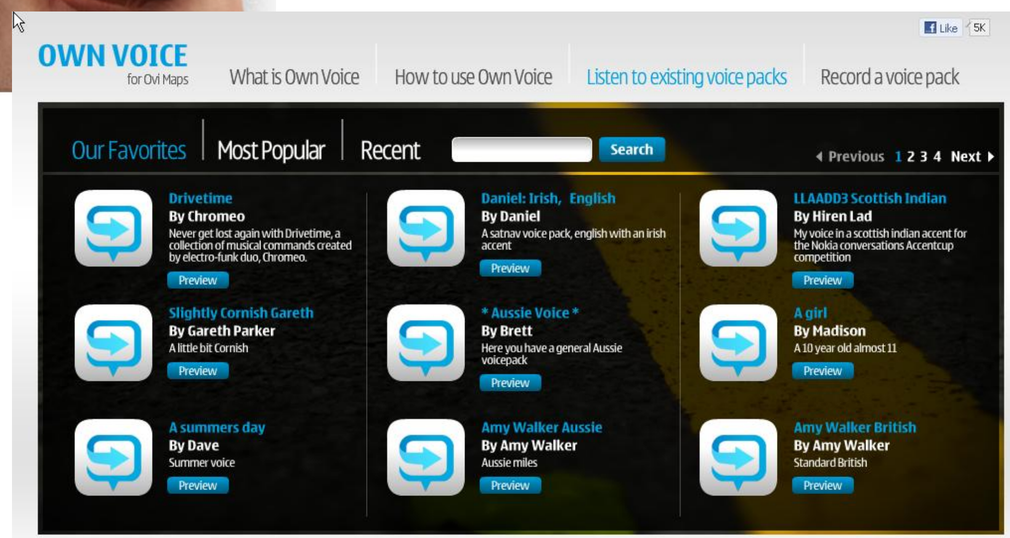
Estimated 15% drop-off rate when encountering a CAPTCHA on mobile devices

http://antigate.com

Mobile CAPTCHA

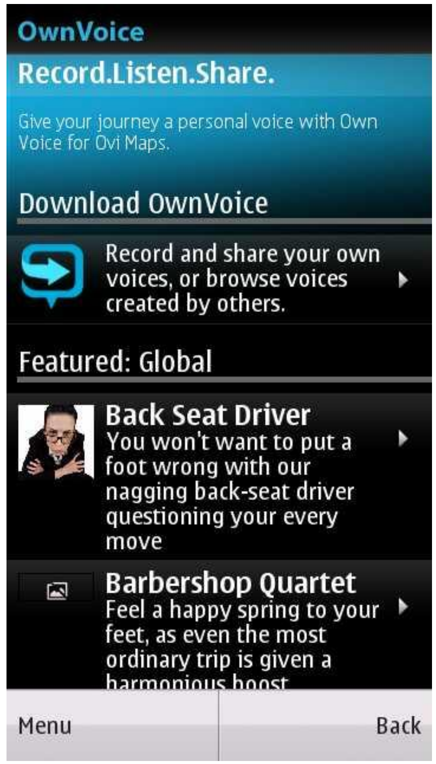
Long tail: app/content creation made easier

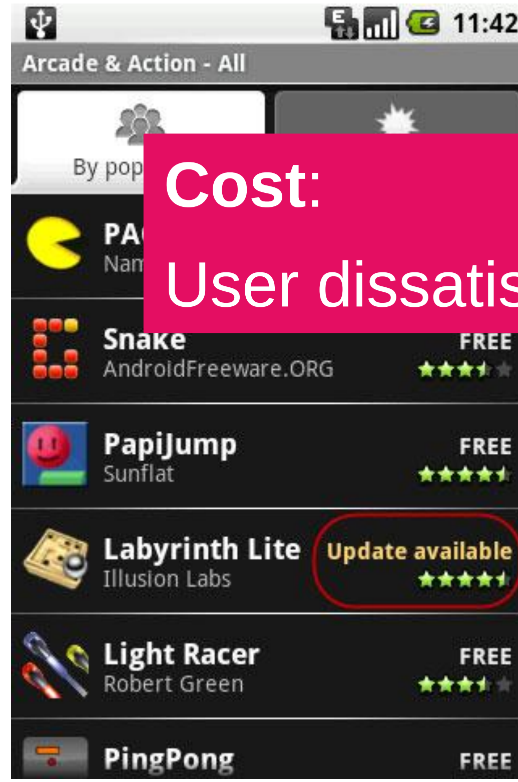




Installation

Plenty of choice for the user





Cost:

User dissatisfaction?

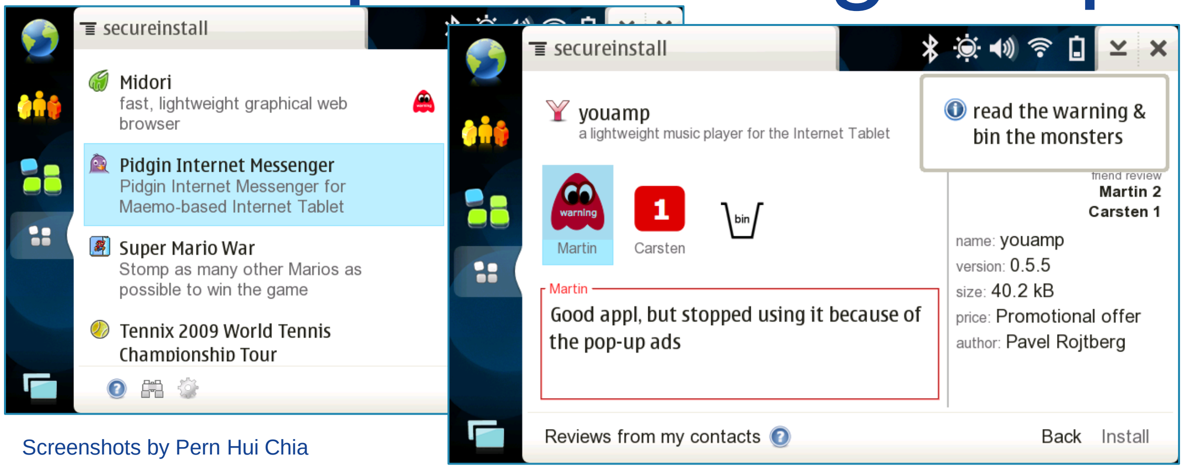
["Is this App Safe?"](#)

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

([WWW 2012](#))

Installation

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

Internet of Things

Early 2000s

From automated universal identification of “things”

AUTO-ID LABS

EPCglobal

2020?

To an interconnected network of billions of “things”

Sensors

Actuators

Autonomous Machine-to-machine communications

Characteristics of IoTs

- Resource Constraints
  - Energy, computation power, storage

  → Lightweight crypto, protocols; novel device architectures

- Scale
  - “One or two per user” to “tens or hundreds”

  → New approaches for intuitive management of IoT devices

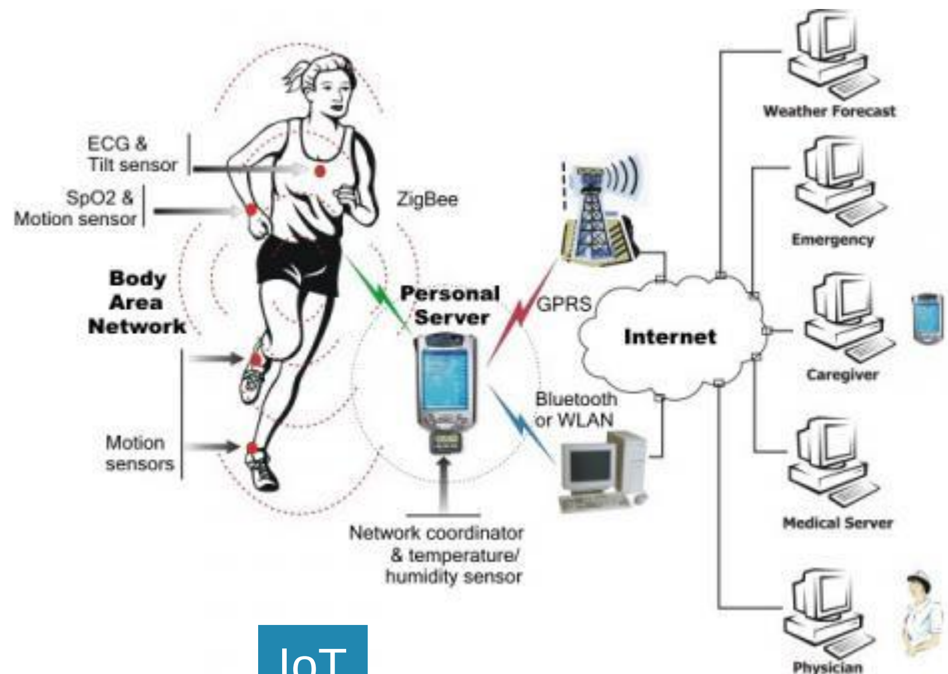
- Non-trivial access policies

Example 1: Medical body area network

- Medical devices near human body
  - Sensors: heart rate, temperature, blood pressure, steps…
  - Actuators: pace maker

- Connected to infrastructure networks
  - Via proxy device (smartphone)

Image taken from: http://si.epfl.ch/page-34870-en.html

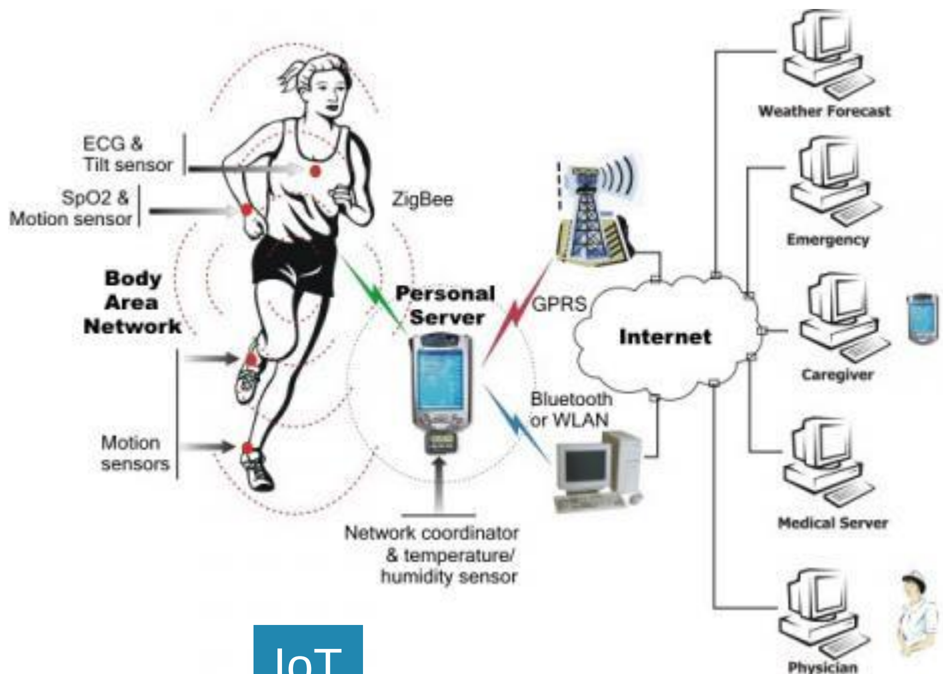


IoT

Example 1: Medical body area network

- Data gathered to an online storage
  - Private data
- By default access to data only for the user herself
  - Also planned sharing (friends, services)
- But unplanned sharing needed!
  - Medical condition
  - Accident
- Privacy vs. safety

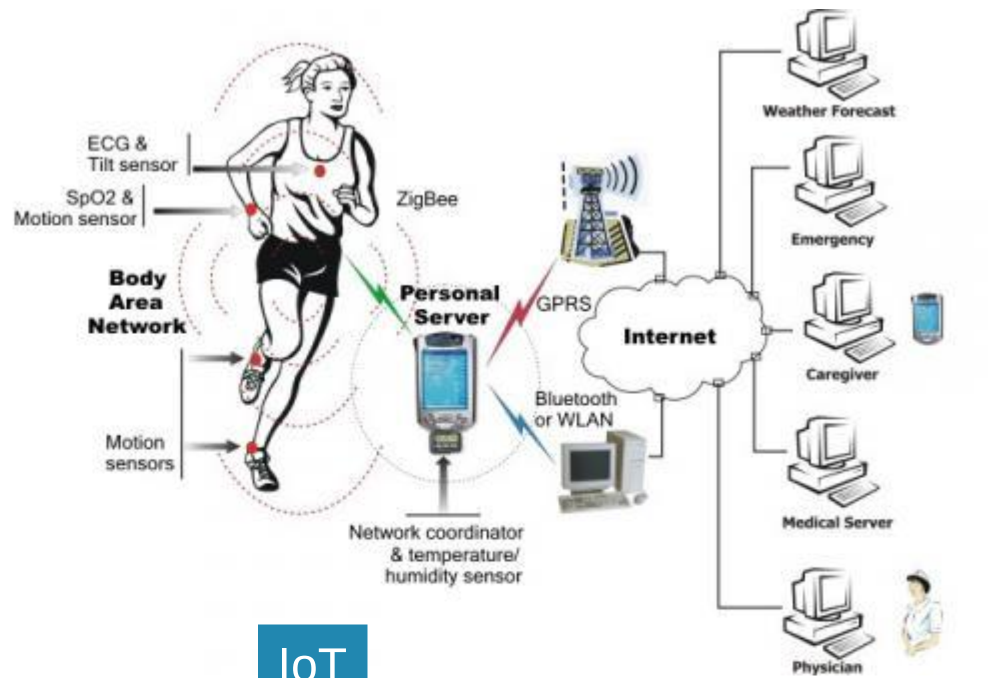
Image taken from: http://si.epfl.ch/page-34870-en.html



Example 1: Medical body area network

- Role-based access control
  - Data readable in online storage
- Attribute-based encryption
- Context-based access control

Image taken from: http://si.epfl.ch/page-34870-en.html

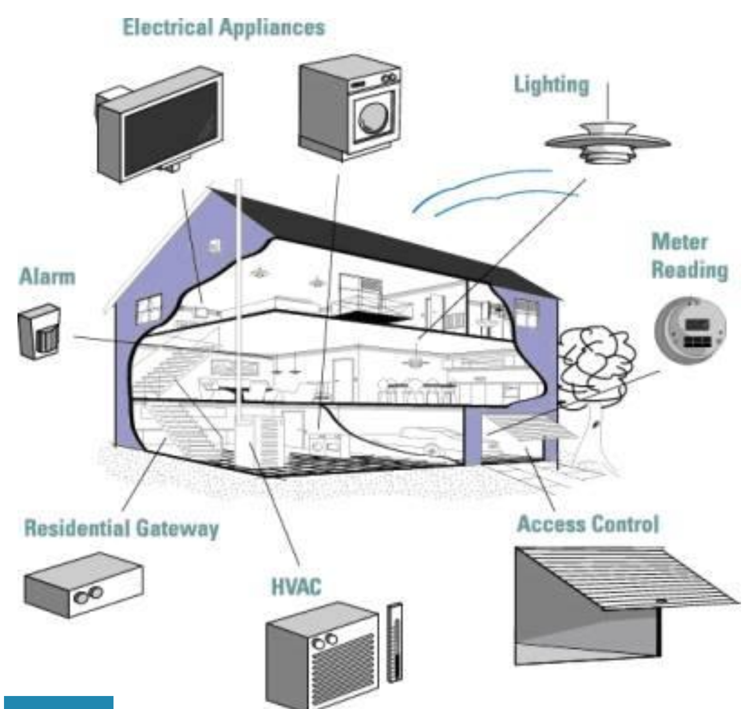


IoT

Example 2: Intelligent home

- Home equipped networked devices
  - Sensors: temperature, motion detect
  - Actuators: lighting, air conditioning, doors
- Connected to infrastructure networks
  - Remote monitoring
  - Remote control

Image taken from: http://www.eetimes.com/design/embedded-internet-design/

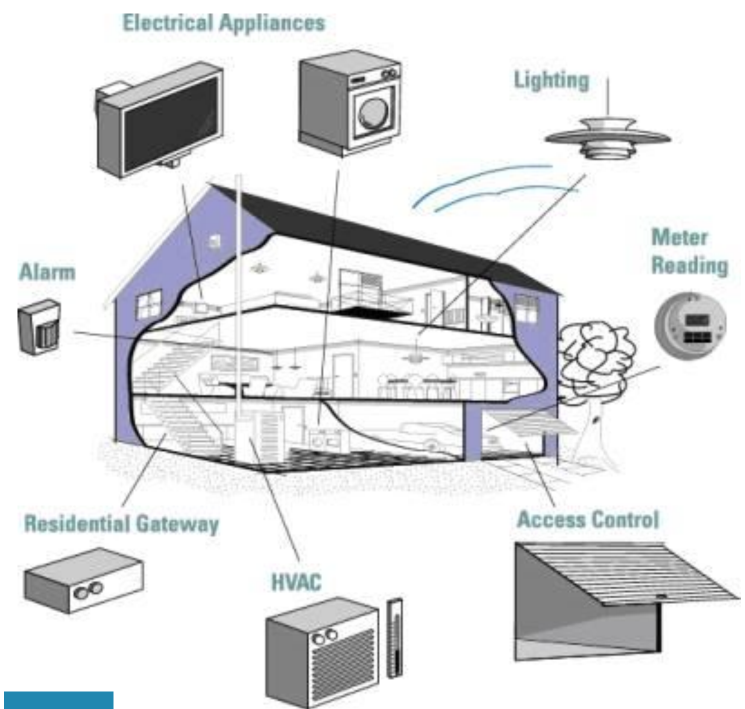


Example 2: Intelligent home

- Access control
  - Be default household owner
  - Delegated access

- Needed: intuitive ways of
  - adding/removing a device
  - specifying access control
    - “this light sensor controls that bulb”
    - “close friends can open the front door”

Image taken from: http://www.eetimes.com/design/embedded-internet-design/



IoT

Challenges in managing access control

Intuitive and secure means for

- Taking ownership of a new device
  - Possible interaction models for take ownership
    - Reading a take-ownership-code from new device
    - Based on co-location
    - …
- Granting and removing access
  - Identity- and role-based
    - “me”, “friends”, “paramedic”, “fire brigade”
  - Demonstrative
    - “this”, “that”
  - context-based
    - “heart-attack”, “fire alarm”, “unsafe neighborhood”

IoT

Some proposed solutions

- Papers from **Workshop on Smart Object Security**
  - "On Access Control in the Internet of Things"
  - A Brief Survey of Imprinting Options for Constrained Devices
  - …
- Data Security and Privacy in Wireless Body Area Networks
- Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption

Mobile devices can help security/privacy

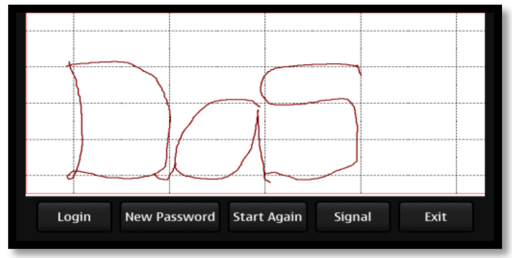
- Mobility and portability can help in surprising ways: e.g.,
  - PayPal Bump
  - "Mobility helps security in ad hoc networks", Čapkun et al, MobiHoc '03
  - ...
- Mobiles can sense location, motion, ambient light, noise level, …
  - Cues from context/history to set sharing, access control policies
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

ISAC

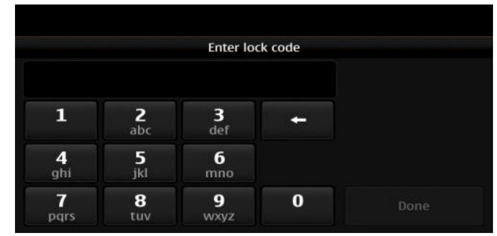
Better Dev. Lock via Context Profiling

Timeout and unlocking method adjusted based on estimated familiarity/safety of current context

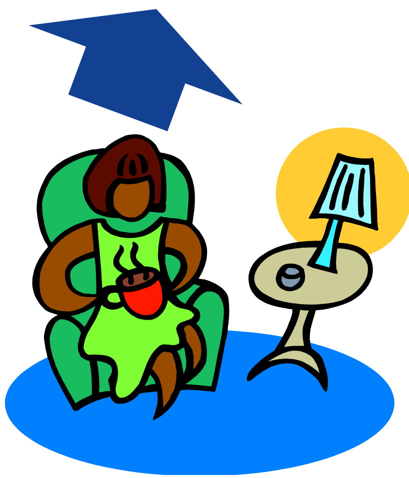
Long timeout



Medium timeout

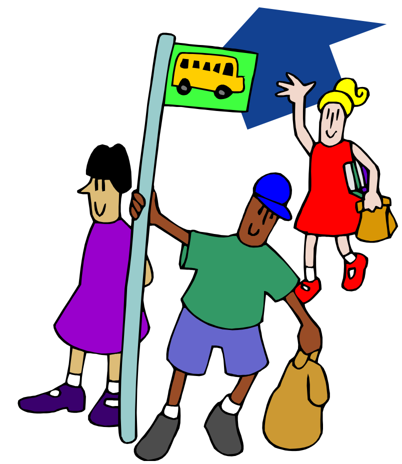


Short timeout



Home

Work Cafeteria



Unknown

ISAC

Context Profiler: estimating safety of a place?

Identify places of interest and profile them over time

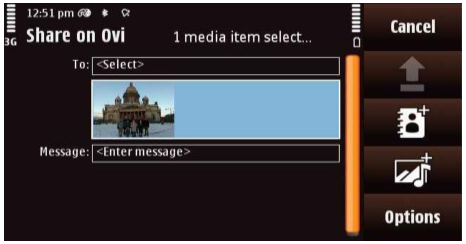
A place may not be always safe (or unsafe)

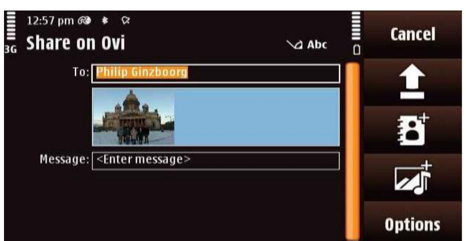
1. Identify places (generally ”contexts”) of interest: CoIs
2. Profile CoIs by keeping track of what is seen there
3. Estimate **familiarity of a device** in a CoI
4. Estimate **familiarity of CoI** based on devices present
5. Estimate **safety** based on current/historical familiarity

SocialCom ’12 Paper on context profiling

Another example: Easier photo sharing

Photo today





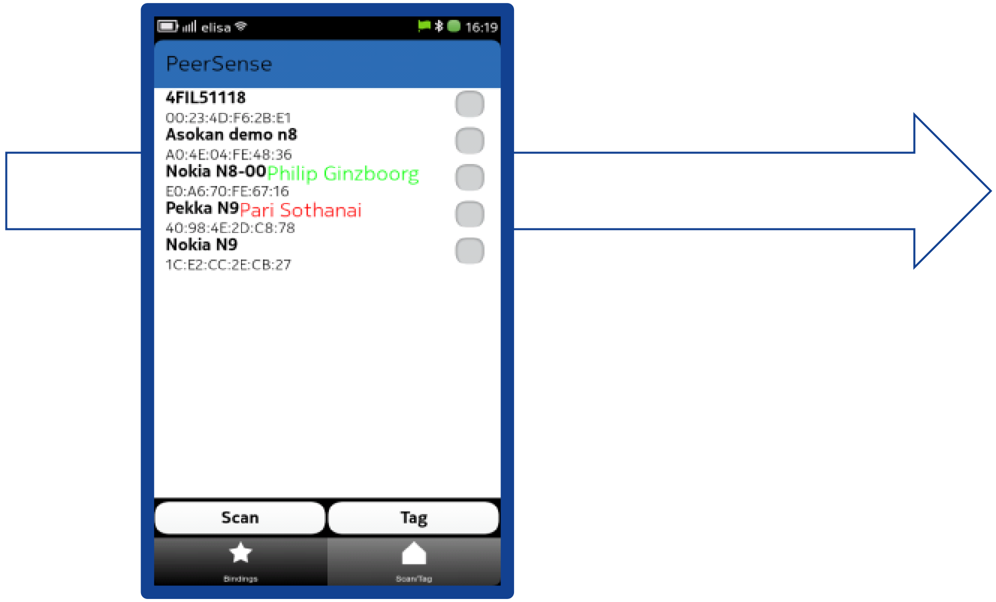
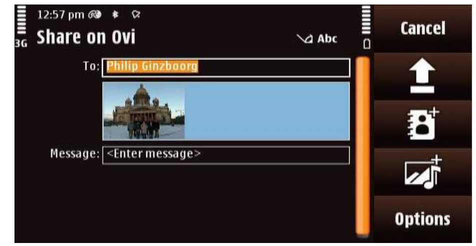


Photo sharing future



ISAC

PeerSense: recognizing nearby friends

- **How can your device recognize your friends' devices?**
  - **intuitive**: one-time simple user action to get started; user need not manually bind friends' names to device addresses
  - **private**: eavesdroppers do not learn names; servers do not learn location or co-location of devices/users
- PeerSense API allows an application to find information about nearby "friends"
  - Example: camera recording nearby friends as photo metadata(as in TagSense); use to infer likely sharing targets
- Status: Demo (shown at Percom 2012)

Summary

- Usable mobile security is a challenging but worthy 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]
    - ….
  - Usability challenges in securing IoT will be harder
- A promising avenue: intuitive security/privacy policy configuration by using context and history of 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?