# PA197 Secure Network Design

## Security aspects of wireless personal area networks (PANs)

Petr Švenda svenda@fi.muni.cz

Faculty of Informatics, Masaryk University

*Please insert any comments, hints or spotted inaccuracies here:*
*https://drive.google.com/file/d/19iHgI93Srh-QuiW1ehbK1c6Z4UYmhM17/view?usp=sharing*

# Overview

- Security considerations of wireless transmission
- Technology for Personal Area Networks (PANs)
  - Bluetooth, NFC, ZigBee
  - Design goals
  - Security vulnerabilities
  - Combination of technologies

# PERSONAL AREA NETWORKS

# Main design goals

- (Not necessary all at the same time)

1. Energy efficiency
   - Running long time only on batteries
2. Physical locality of communication (NFC)
   - Imposing restrictions on attacker
3. Quick establishment of temporary connections
   - Usable security
4. Ad-hoc networking
   - Temporary networks without pre-fixed structure

# Basic steps of communication

1. Discover other device(s)
   – Public broadcast vs. private sharing
2. Authenticate and establish initial key(s) (pairing)
   – Usually once for new devices
3. Authenticate and refresh keys for paired devices
   – If long-term persistence is maintained (known devices)
4. Exchange packets between devices
5. Terminate connection

Wireless networks

# WIRELESS MEDIUM - ATTACKS

# Attack surface is large

- Wireless signal propagates more easily
  - Eavesdropping, message injection
  - Also more difficult to localize attacker
- Processing transmissions more complicated
  - Potential for bugs in implementation, network stack
- Potential for physical device compromise
  - Device not connected => easier to be lost/stolen…

# Wireless medium – basic properties

- Eavesdropping on active transmission is easy
  - Omnidirectional vs. directional antenna
  - Active vs. passive communication mode
- Eavesdropping on passive device (RFID,ISO14443) more difficult (passive mode)
  - Tag/card does not emit signal on its own
  - Tag/card specifically distorts EM field measured by reader
- Multiple channels may require multi-channel eavesdropping
  - Frequency hopping based on secret sequence (PRNG)

# Generic attacks: Eavesdropping

- Active $\rightarrow$ active transmission
  - Directional antenna, e.g., Bluetooth $10^2 \rightarrow 10^4$ meters
- Active $\rightarrow$ passive transmission
  - Tens of meters for active signals (reader $\rightarrow$ tag), easy
  - Up to 1m for passive signals (tag $\rightarrow$ reader), difficult
- Signals must be reliable enough for normal communication => stronger than necessary minimum
- Eavesdropping cannot be generally prevented
  - Possibly only significantly limited in distance (NFC)
- Solution: use secure channels (encryption, auth)

# Attack: record and compromise later

- Eavesdropped communication is encrypted
- Used key is later recovered by other means
  - End-node compromise, side-channel attack, bruteforce…
  - => Past communication can be decrypted (later)
- How to prevent?
  - (Perfect) forward secrecy protocols (e.g., ECDH)

# (Perfect) forward secrecy protocols

- Long-term key compromise doesn’t compromise past session keys

1. Fresh keypair generated for every new session
2. Ephemeral public key used to exchange session key
3. Ephemeral private key is destroyed after key exchange
   - Captured encrypted transmission cannot be decrypted

- Long-term key is used only to authenticate ephemeral public key to prevent MitM
- Where used? TLS, OTR/Signal, ePassports…
- Where NOT used? If only symmetric crypto based

# DH based on elliptic curves used (ECDH)

Diffie-Hellman Key Exchange

| Step | Alice | Bob |
|---|---|---|
| 1 | Parameters: EC curve, G (base point) | |
| 2 | $A$ = random()<br>$a$ = A x G (scalar multiplication) | random() = $B$<br>B x G = $b$ |
| 3 | $a \longrightarrow$<br>$\longleftarrow b$ | |
| 4 | $K$ = A x B x G = A x b | B x a = A x B x G = $K$ |
| 5 | $\longleftarrow E_K(data) \longrightarrow$ | |

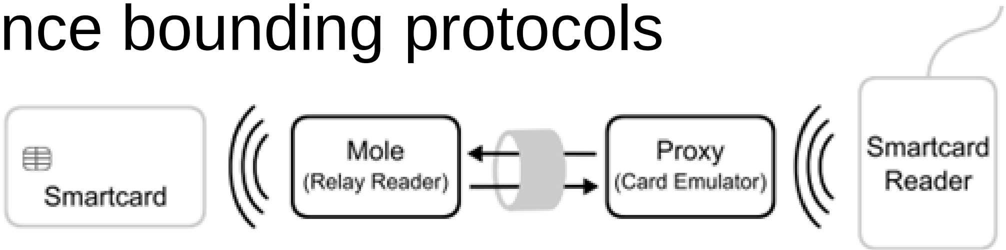
http://www.themccallums.org/nathaniel/2014/10/27/authenticated-key-exchange-with-speke-or-dh-eke/

# Generic attacks: data corruption

- Attacker tries to corrupt data during transmission
  - Channel level: additional transmission → jamming
  - Link/tunnel level: sinkhole, dropper…
    - Form of denial-of-service
- Broad vs. selective jamming
  - Broad jamming requires higher power of transmission
  - Selective jamming corrupts only few bits in header / packets
- Solution: device detects and verifies signal strength, counts transmitted/dropped packets…
  - But signal naturally fluctuates => harder to detect attack

# Generic attacks: Man-in-the-middle

- Third device acts as relay between two legitimate devices
  - Log/block/modify communication
  - Emulates perception of close presence (door lock, card payment)
- If mounted against active-active communication mode
  - Attacker can be farther away
  - Possibly needs to block legitimate traffic (to legitimate party)
- If mounted in active-passive mode
  - Attacker needs to be closer to victim (passive → active)
- May require low-latency relaying on attacker's side
- Potential defense: distance bounding protocols

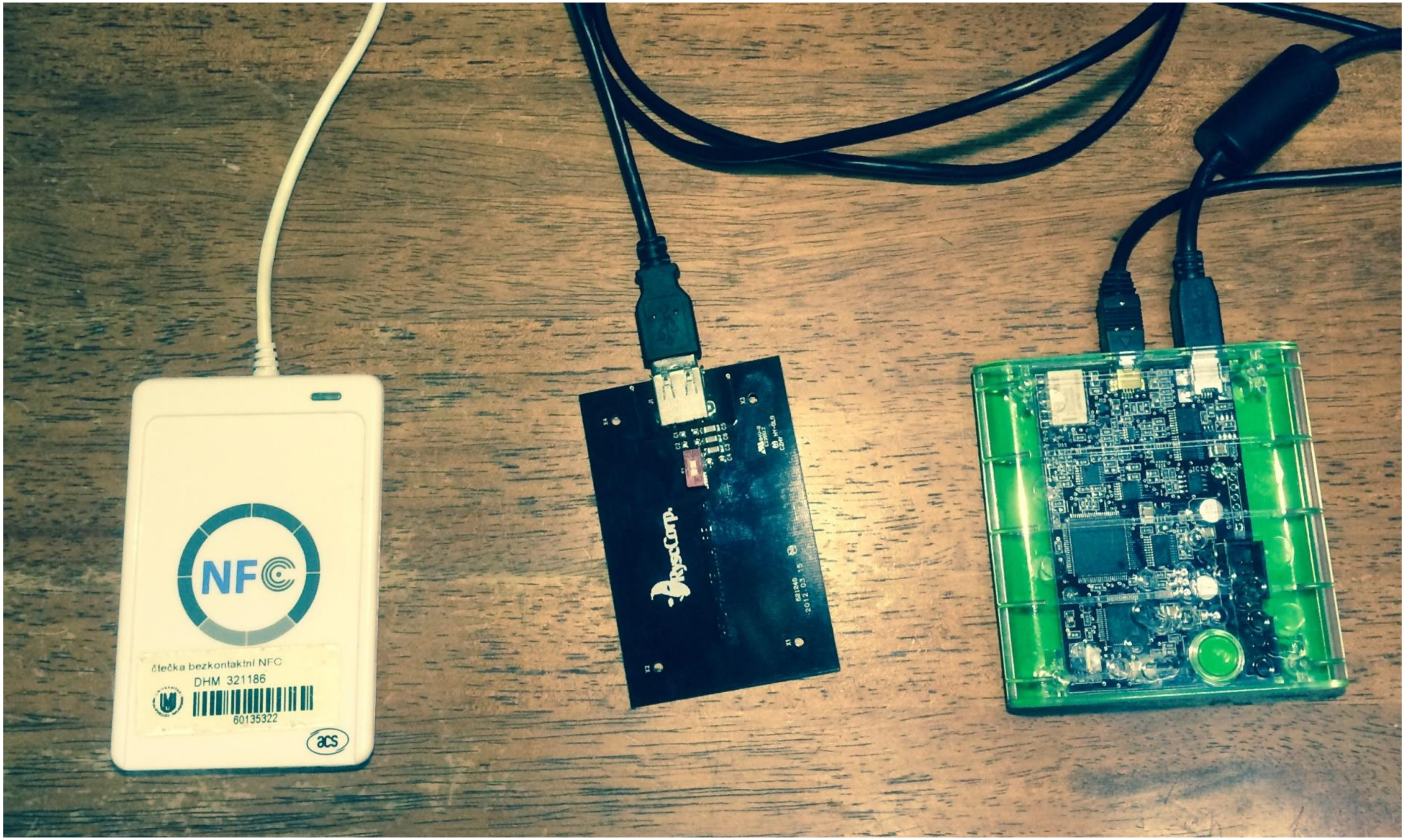


# Example: Passive wired relay

- No amplifier or other active components required
- Coaxial cable between two antennas, 20 metres or more
- Very low delay (practically not detectable)
- Low cost

*http://cdn.intechopen.com/pdfs-wm/44973.pdf*

# Example: ePassport simulator Proxmark III (M. Korec) https://is.muni.cz/auth/th/396490/fi_b/



# Distance bounding protocols

- Enable verifying device to establish upper bound on physical distance from connecting device
  – Time to receive response to challenge is measured
  – Multiplied by speed of light (~RF waves speed)
- Problem: transmission time may be significantly smaller than necessary processing time
  – Especially for high-frequency channels
  – Important to measure precisely 1 ns => 15cm error
- More likely to detect active MitM than passive relay
- http://cdn.intechopen.com/pdfs-wm/44973.pdf

Wireless networks - Bluetooth

# BLUETOOTH

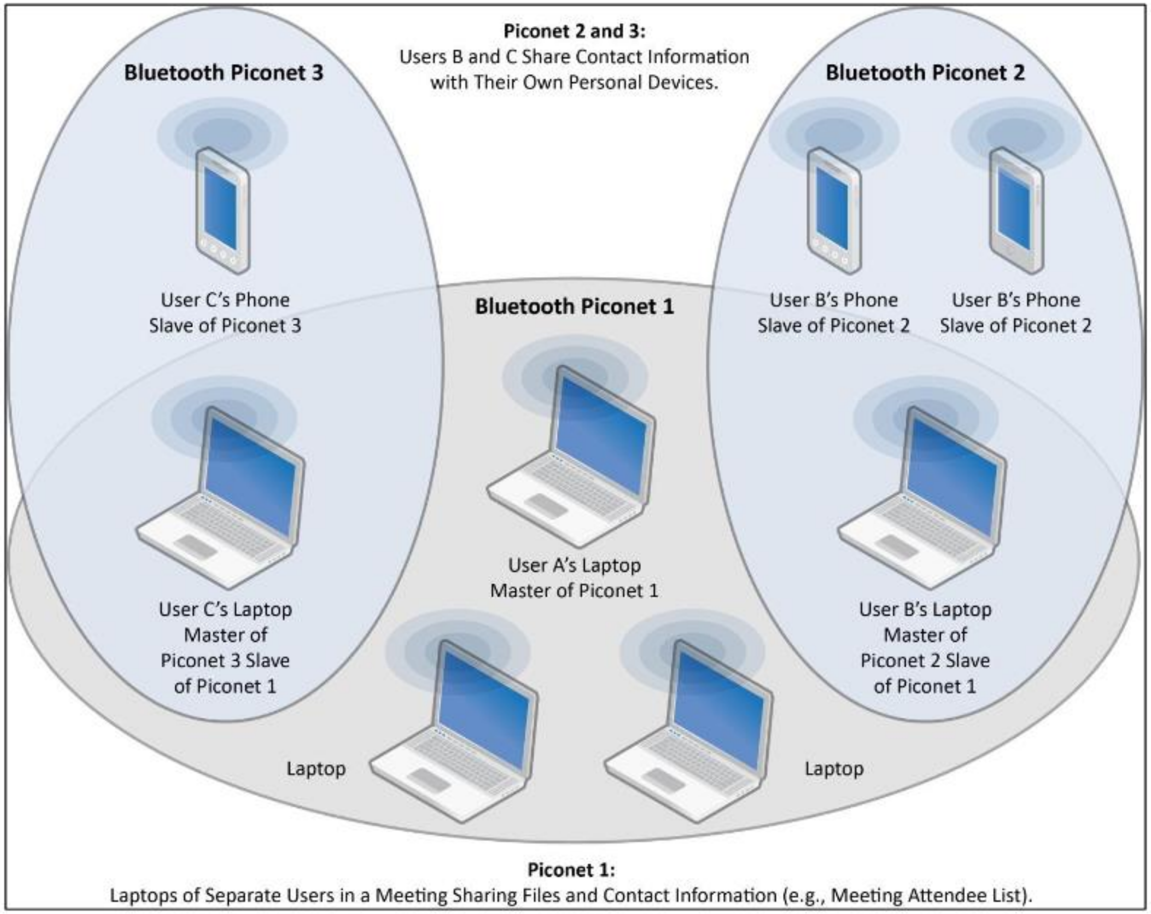
# Bluetooth – basic information

- Wireless standard for exchanging data over short distances
  - IEEE 802.15.1 standard (no longer maintained)
  - Specification maintained by Bluetooth Special Interest Group (SIG)
- UHF radio waves in the ISM band from 2.4 to 2.485 GHz (globally unlicensed band, scientific and medical)
  - Frequency-hopping spread spectrum (1600 hops/sec), Adaptive Frequency-Hopping (AFH, avoids crowded frequencies)
  - 79 designated Bluetooth 1MHz channels (40 for BT 4.x)
- Class 1/2/3 devices (max. power, distance ~100/10/1m)
- Speed 1Mbit – 24Mbit / sec
- Bluetooth usage profiles (https://en.wikipedia.org/wiki/List_of_Bluetooth_profiles)

# Bluetooth - networking

- Each BT device has unique 48-bit device address
- Discoverable vs. hidden mode
  - On demand response (device name, class, services, info)
  - If discoverable then always respond
  - If hidden then respond only if other device address is already known
- Packet-based protocol with master-slave order
  - One master $\rightarrow$ up to 7 slaves (forms piconet)
  - Even and odd medium slots for master/slave transmission
- Multiple piconets form scatternet
  - Some devices both master in piconet X and slave in piconet Y
  - Extends device range via multi-hop communication
  - (Not really used in practice so far)

# Bluetooth – piconets, scatternet




*http://csrc.nist.gov/publications/nistpubs/800-121-rev1/sp800-121_rev1.pdf*

# Bluetooth vs. WiFi

- AP-based WiFi is asymmetric (infrastructure)
  - BT is master – slave, but usually ad-hoc
- BT generally requires less configuration
- BT is more power efficient, especially BT 4.x LE
- AP-based WiFi is generally more suitable for infrastructural placement, BT for ad-hoc networking
- Cooperation of technologies
  - Initial pairing setup via BT, fast transmission via WiFi

# PA197 - PANs, Bluetooth

Wireless networks - Bluetooth

# BLUETOOTH SECURITY

# Security requirements

- What would you like to have? ☺

- NIST guidelines to Bluetooth security
  - https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf

# Bluetooth – versions, security features

- BT 1.0 [1994?] Initial version, mandatory encryption
- BT 1.1 [2002] Possibility for non-encrypted channels
- BT 2.1 [2007] Secure simple pairing (SSP)
- BT 3.0 [2009] Negotiation of high speed over 802.11 link
- BT 4.0 [2010] BT low energy (Wibree), coin cell power, Bluetooth Smart Ready, SSP not available
- BT 4.2 [2014] Introduces important features for IoT, LE Secure Connections, Link Layer Privacy, ECDH-based SPP
- BT 5 [2016] Larger range and transmission speed
- BT 5.1 [2019] Angle of Arrival/Departure (tracking devices), broadcast data without full connection (e.g., thermometer)

# Bluetooth security modes

- Mode 1 provides no security
  - Any device can connect, no encryption
  - Up to Bluetooth 2.0 + Enhanced Data Rate (EDR) and NOT beyond
- Mode 2 provides security at the service level
  - After a communication channel is established
  - Centralized security manager controls
- Mode 3 provides security at the link level
  - Before a logical channel is established
  - Authentication and encryption of all connections
  - Decreases attack surface, but requires key predistribution
- Mode 4 provides Secure Simple Pairing
  - Connects two previously unpaired devices (DH, ECDH)

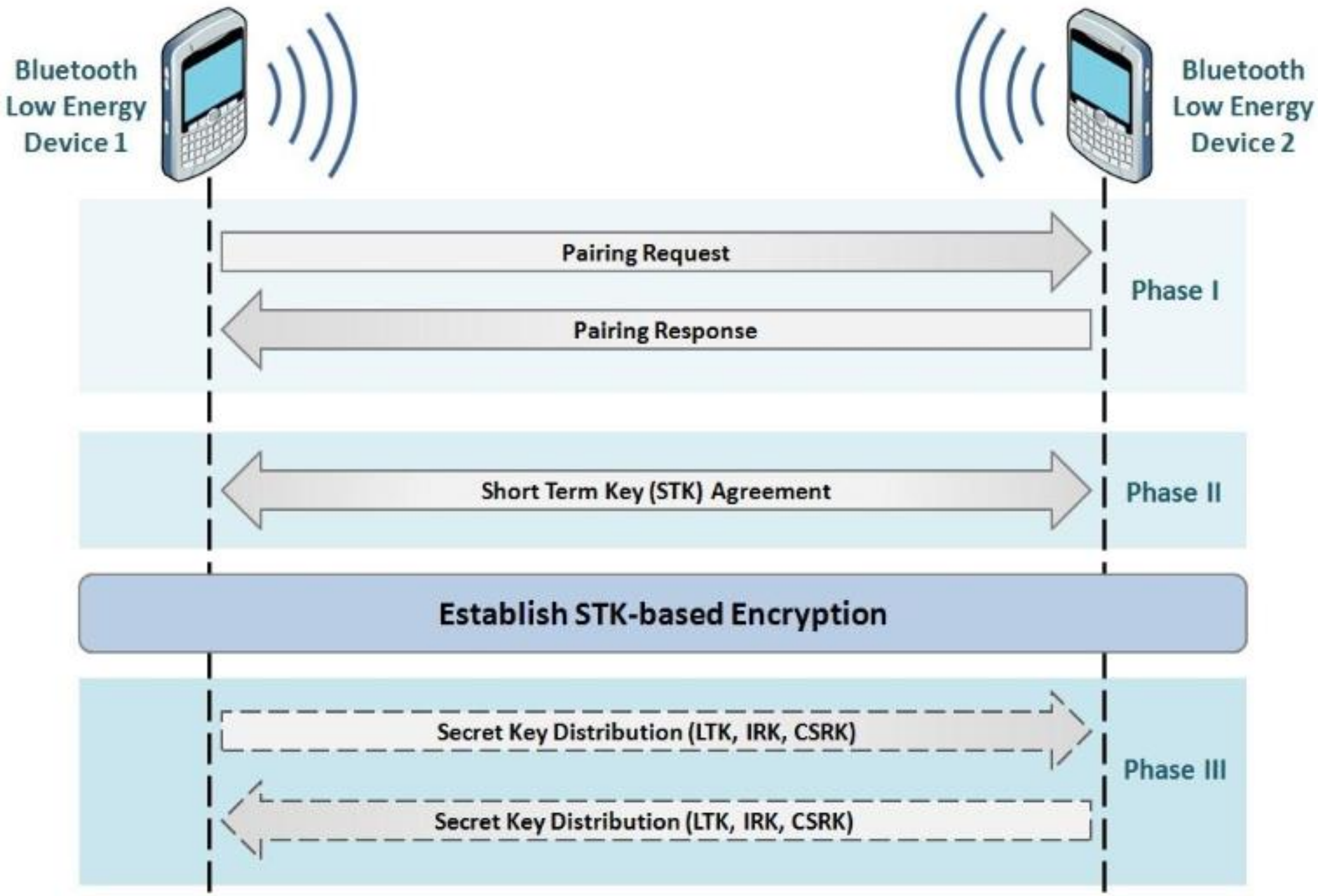
# Bluetooth – crypto algorithms used

- SAFER+ block cipher
  - used as building block for key derivation, authentication
- E0 stream cipher for encryption
  - Encryption key, master device BT address, real-time clock
- E22 key derivation algorithm
  - Derive initial key from address, rand and PIN
- E21 session key derivation algorithm
  - Link key generation from initial key
- E1 authentication algorithm
  - Authenticate devices after pairing
- AES cipher in Counter mode (AES-CCM)
  - Introduced for Bluetooth LE (BT 4.0)
- General trend: used to be custom crypto (earlier, < 4.0), move towards standard primitives (now, >= 4.0)

**BT < 4.0**

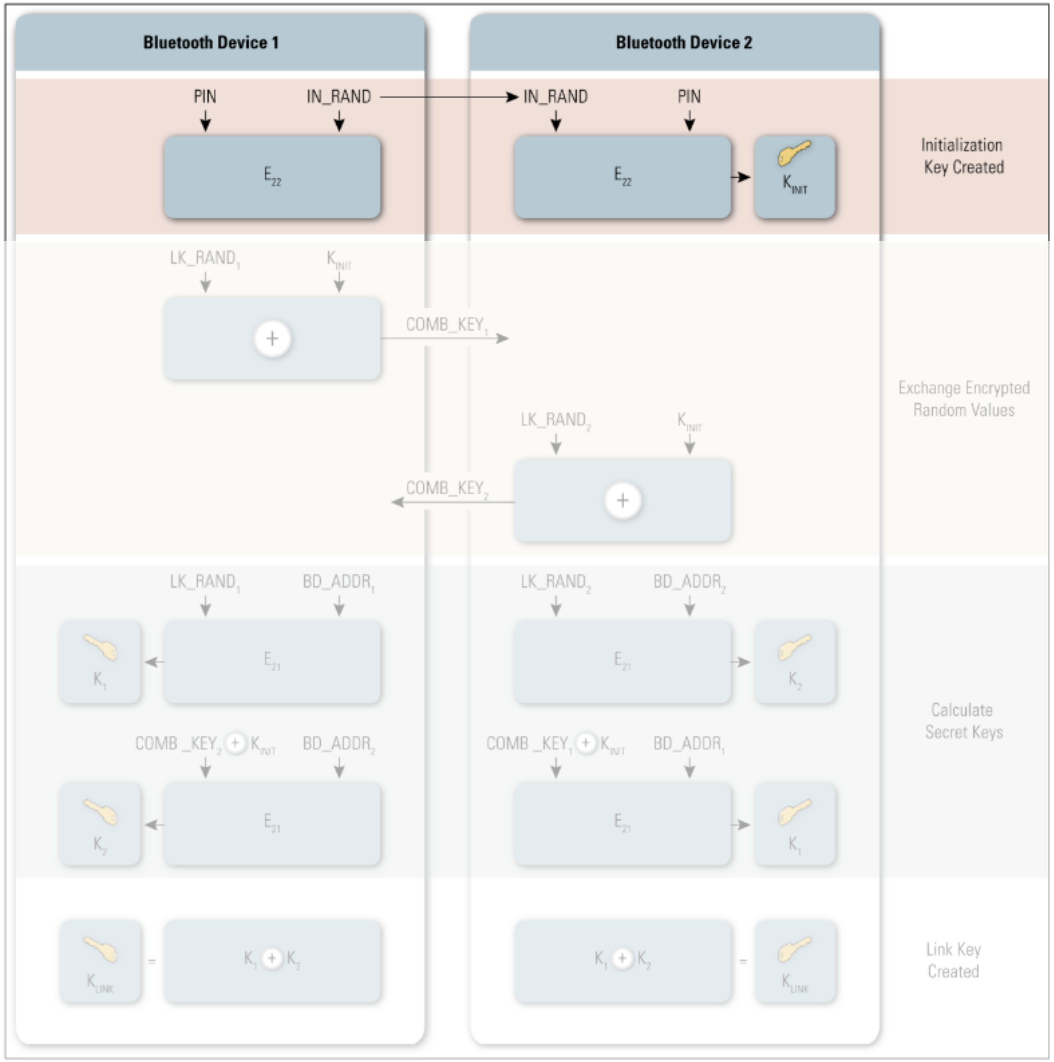
**BT >= 4.0**

# Bluetooth pairing




https://www.bluetooth.com/specifications/adopted-specifications

# BT Initial key → link key (E22 and E21)

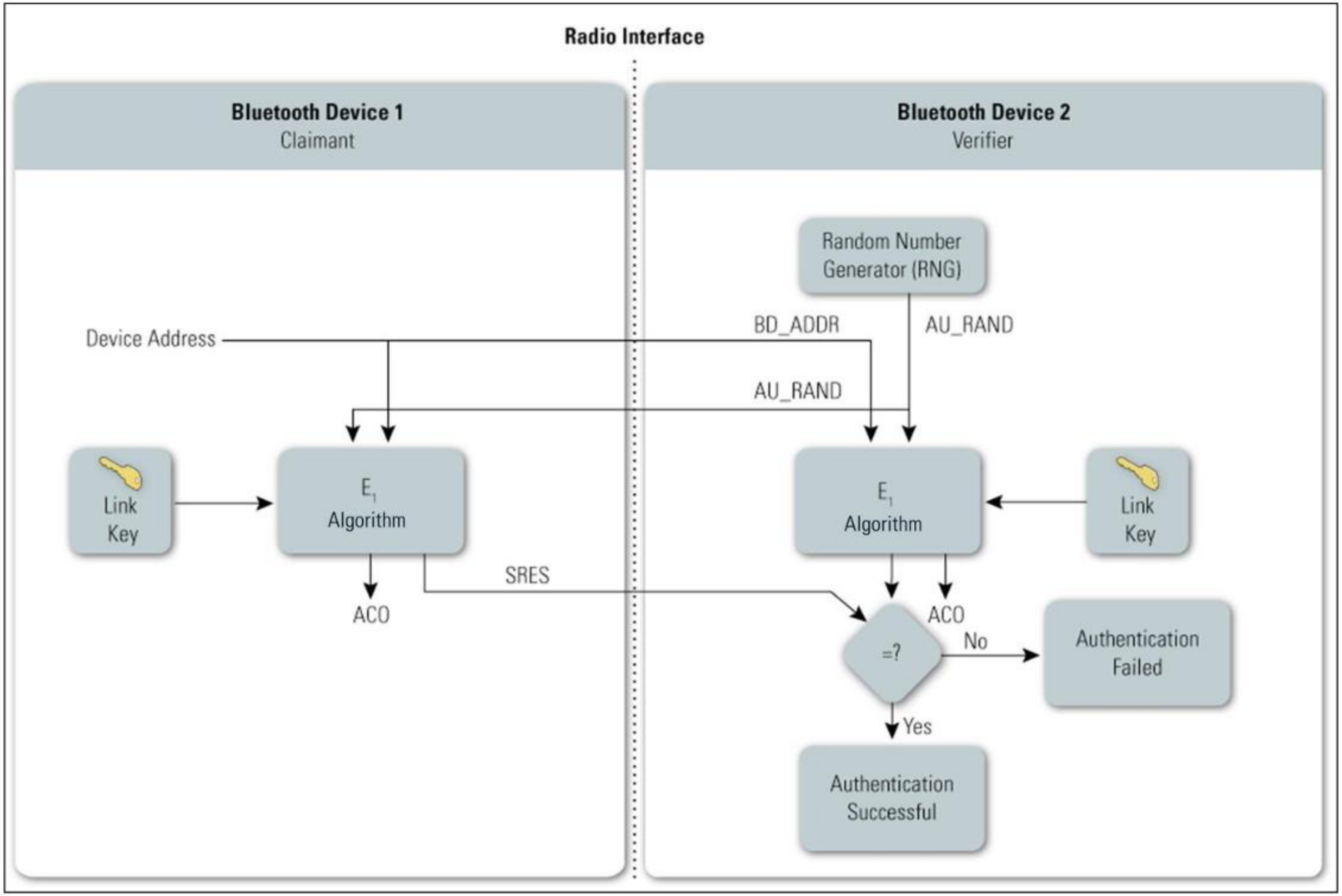



http://csrc.nist.gov/publications/nistpubs/800-121-rev1/sp800-121_rev1.pdf

# BT Pairing – Legacy pairing

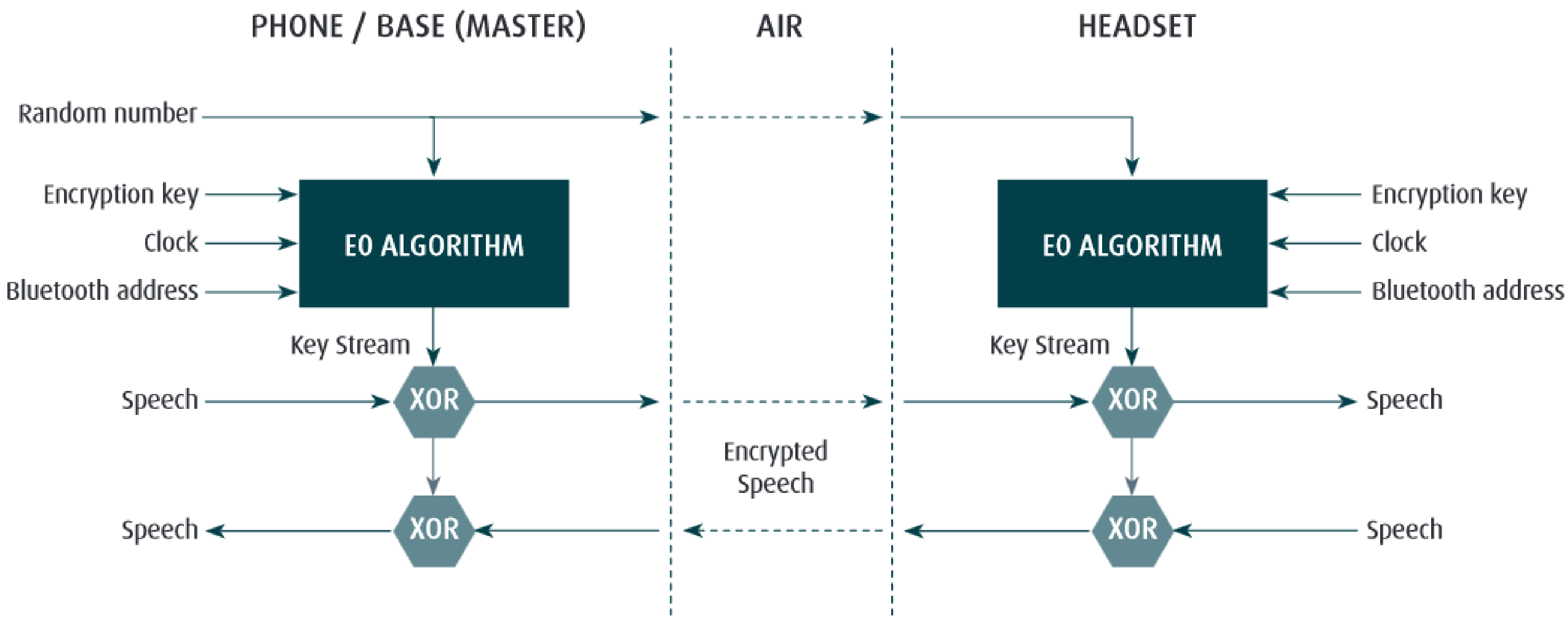
- BT 2.0 and before
- Initial key exchange ($K_{INIT}$) over unencrypted link
  - What attacks are possible?
- Passkey/PIN $\rightarrow$ initialization key $\rightarrow$ link key
  - Short passkey problem (passive attack ~ms)
  - http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/index.html

# BT – authentication (E1)




http://csrc.nist.gov/publications/nistpubs/800-121-rev1/sp800-121_rev1.pdf

# Bluetooth – E0 encryption




*http://www.jabra.com/~/media/Documentation/Whitepapers/WP_Bluetooth_50004_V01_1204.pdf*

# Bluetooth attacks

- Bluesnarfing, Bluebugging
  - Unauthorized extraction of data from device (discoverable mode)
- Guessing device address via brute-force attack
  - 48bit MAC address, but first 24 as manufacture's id
- Limited key-usage period (< BT 2.1)
  - Around 23.5 hours before simple XOR attack (E0 stream cipher)
- Encryption can be forced to be turned off (< BT 2.1)
- L2CAP level attacks
  - Parts of data packet not protected by integrity
  - Fuzzing used to find flaws in device's firmware

# BT Pairing – Legacy pairing

- BT 2.0 and before
- Initial key exchange ($K_{INIT}$) over unencrypted link
  - What attacks are possible?
- Passkey/PIN $\rightarrow$ initialization key $\rightarrow$ link key
  - Short passkey problem (passive attack ~ms)
  - http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/index.html
- Is attack prevented by perfect forward secrecy?
  - No, but force attacker to be active (MitM)
- How to use Passkey/PIN to prevent MitM?
  - Escalation protocols (fresh DH + PIN for authentication)

# DH based on elliptic curves used (ECDH)

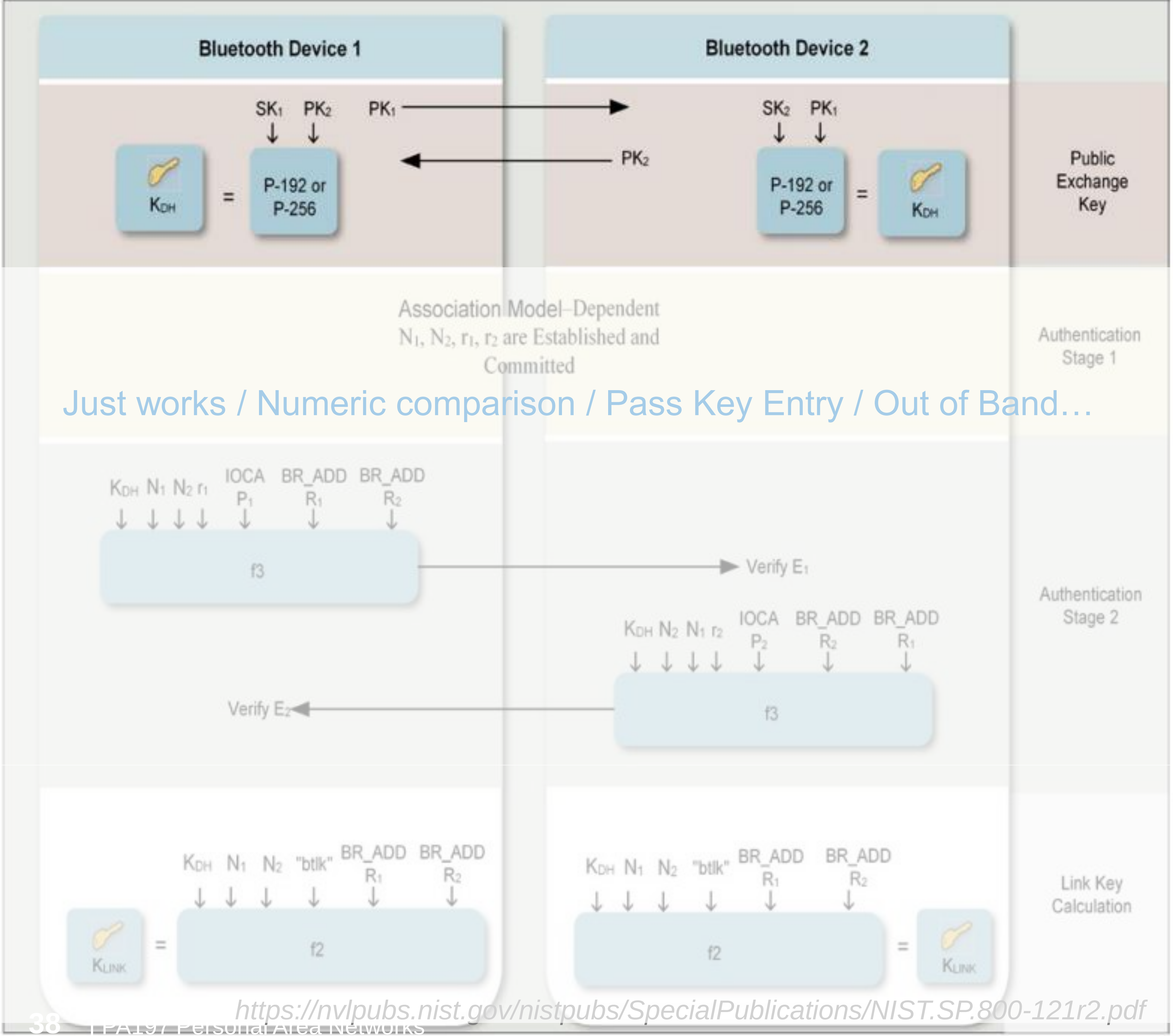
Diffie-Hellman Key Exchange

| Step | Alice | Bob |
|---|---|---|
| 1 | Parameters: EC curve, G (base point) | |
| 2 | $A = \text{random}()$<br>$a =$ A x G (scalar multiplication) | $\text{random}() = B$<br>B x G $= b$ |
| 3 | $a \longrightarrow$<br>$\longleftarrow b$ | |
| 4 | $K =$ A x B x G = A x b | B x a $=$ A x B x G $= K$ |
| 5 | $\longleftarrow E_K(data) \longrightarrow$ | |

http://www.themccallums.org/nathaniel/2014/10/27/authenticated-key-exchange-with-speke-or-dh-eke/

# BT Pairing – Secure Simple Pairing (SSP)

- Secure Simple Pairing (SSP, from BT 2.1)
  - Public-key crypto based (ECDH from BT 4.2) for key agreement
- How to authenticate ECDH public part?
  - *Just works* mode: no authentication
  - *Numeric comparison* mode: display challenge and confirm
  - *Passkey Entry* mode: insert passphrase
  - *Out Of Band* mode: use other channel to establish auth. key
- 128 bit random link key for encryption (at maximum)
  - Length negotiated by devices




Just works / Numeric comparison / Pass Key Entry / Out of Band…

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf

# Bluetooth LE/Smart (BT 4.x) (2010)

- For low-energy, storage/computation restricted devices
- Simplified protocol for link key establishment
  - LE pairing protocol establish long-term key (LTK)
  - Key transport instead of key agreement is used
  - One device generates LTK and transports during pairing
    - What are the security implications?
- Support for out-of-band for pairing
  - E.g., NFC-based exchange of Temporary Key (TK)
- AES-CCM introduced (relevant for FIPS 140-2)
- Introduction of private device address
  - Public device address from encrypted (changing) private address
  - Eavesdropper will not learn public address => no address tracking

# Bluetooth LE/Smart (BT 4.0)

- BT Secure Simple Pairing uses Diffie-Hellman
  - To prevent passive eavesdropping and forward secrecy
  - But asymmetric crypto is slow(er) + energy consuming
- Design decision for 4.0 – no SSP at the time
  - BT 4.0 LE/Smart pairing is symmetric-cryptography based
  - Passive eavesdropping + delayed key compromise possible
- BT LE pairing with ECDH keys added in BT 4.2
  - Authenticated ECDH exchange of link key

# Bluetooth – Tracking privacy

- Each BT device has unique 48-bit device address
  - BT 1.0 required mandatory transmission, later dropped
- Discoverable / non-discoverable mode
  - Once discoverable, device's address is trackable
  - Address space (48b, manufacturer) can be brute-forced
- BT 4.0 (BT LE) allows for private device address
  - Public device address (used in key establishment) broadcasted only in encrypted form
  - Eavesdropper cannot track target device based on MAC

# Bluetooth security tools

- BlueSpam - sends file via OBEX to active devices
- BlueHell http://sourceforge.net/projects/bluehell/
- Bluelog http://tools.kali.org/wireless-attacks/bluelog
  - Discover and log discoverable devices
- BlueMaho https://wiki.thc.org/BlueMaho
  - Monitor devices, test known attacks
- Bluepot https://github.com/andrewmichaelsmith/bluepot/
  - Bluetooth Honeypot

# Bluetooth – (moral) summary

- One of early protocols intended for battery-powered “limited” devices (BT 1.x)
  - Cell phones that time, wireless headsets…
  - Vulnerabilities due to insecure defaults, proprietary crypto etc.
  - Typical for the period of its introduction (recall also WiFi’s WEP…)
- More security features introduced (BT 2.x)
  - But also usability, adoption and intellectual property dispute issues
- Cooperation with other technologies, speed (BT 3.x)
  - Initial exchange and configuration, then faster WiFi transmissions
- Added focus on extra low energy devices (BT 4.x)
  - Secure by default, standardized crypto algorithms
  - Renewed interest and support, wider adoption

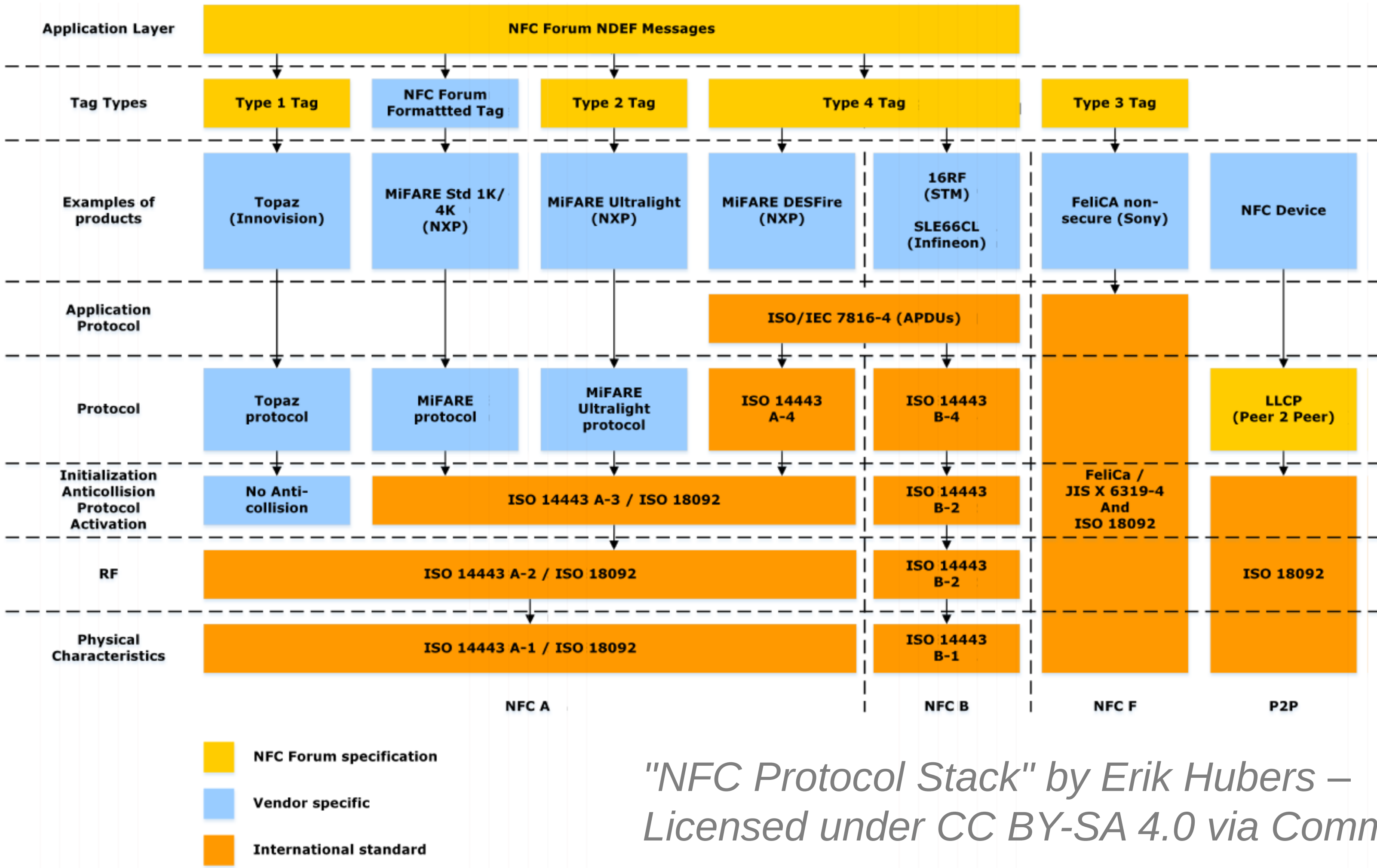

Wireless networks – Near Field Communication

# NFC

# Near Field Communication (NFC)

- Low-power, low-bandwidth communication
  - Initially for reader to tag communication
  - Possibility for tag emulation by device (=>device to device)
- Be aware of potential confusion of "NFC" term
  1. As general term (short distance communication)
  2. As NFC as specific implementation (NFC A, ISO18092)

# NFC standards




*"NFC Protocol Stack" by Erik Hubers – Licensed under CC BY-SA 4.0 via Commons*

# Security goals of NFC

1. Physical presence proof
   – Only short distance communication possible
   – Locality of eavesdropping
2. Simplify key management for other protocols (OOB)
   – Uses physical presence proof
   – NFC → initial key → BT SSP → BT/WiFi transmission
   – NFC → IP, MAC, key → WiFi-Direct
3. Utilize secure hardware via NFC reader
   – Physical tag, token, cryptographic smart card…
4. Turn mobile phone into security token
   – Card emulation

# NFC communication modes


1. Reader/writer mode
   – Read (and/or write) NFC tags and stickers
   – No security except physical presence bounding
   – Usually only tag’s/sticker’s ID transmitted
2. P2P mode
   – exchange data with other NFC peer
   – used by Android Beam between two NFC-enabled phones
3. Card emulation
   – NFC device emulates tag/cryptographic smart card

# NFC mode: Card emulation

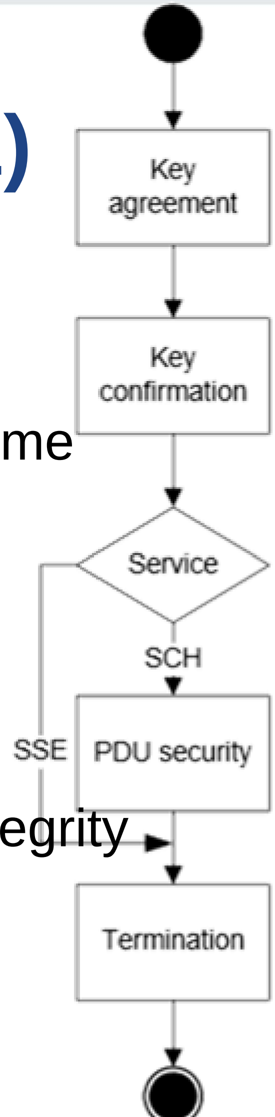
- NFC device emulates tag/cryptographic smart card

1. Card emulation mode
   - NFC device acts as NFC card
   - Emulated by separate chip in device – secure element
     - Commands are relayed to real card
2. Host-based card emulation
   - Emulation without physical secure element
   - Phone provides functionality of smart card
     - Software "smart card"
   - Apple Pay, Google Pay…

# NFC as bootstrapping technology

- Out Of Band (OOB) exchange of initial secrets
  - Utilizes “physical” presence property of NFC
  - Simplifies initial key exchange
    - dependency on difficulty of eavesdropping/MitM
- Android Beam
  - Uses NFC to exchange 6-digits passcode for Bluetooth
- Samsung S-Beam
  - IP,MAC via NFC for WiFi-Direct
- …

# NFC security (NFC-SEC, NFC-SEC-01)




- "Shared Secret Service" (SSE)
  - Results in confirmed shared key between devices
  - Based on Elliptic Curve Diffie-Hellman key exchange scheme (ECDH-192b)
  - Not authenticated (MiTM possible, but physical location)
- "Secure Channel Service" (SCH)
  - Results in link key for secure channel derived from SSE
  - Uses AES and AES-CRT for key derivation, encryption, integrity
- Application-level security possible
  - Use NFC to exchange keys for Bluetooth/WiFi
  - Implement custom protocol between devices (if needed)
- http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-386.pdf
- http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-385.pdf

# Fuzzing NFC stack

- Direct emulation of one side
  - Card emulation mode
- MITM between reader and card (proxy)
  - Data modified in transport
  - No need to implement NFC stack fully
- Compromise of NFC stack allows for BT open
  - Android, BLUETOOTH_ADMIN
- R. Miller, Exploring the NFC Attack Surface (2012)
  - http://media.blackhat.com/bh-us-12/Briefings/C_Miller/BH_US_12_Miller_NFC_attack_surface_WP.pdf

# NFC vs. Bluetooth

- NFC consumes significantly less energy
- NFC has significantly shorter maximum distance
  - Active→passive mode, advantage of physical bounding
- NFC is compatible with existing standards/devices
  - Passive RFID
- Bluetooth LE moved more towards energy-efficiency
  - But still only active-active mode

Wireless networks – Moving towards more networking

# ZIGBEE (IEEE 802.15.4)

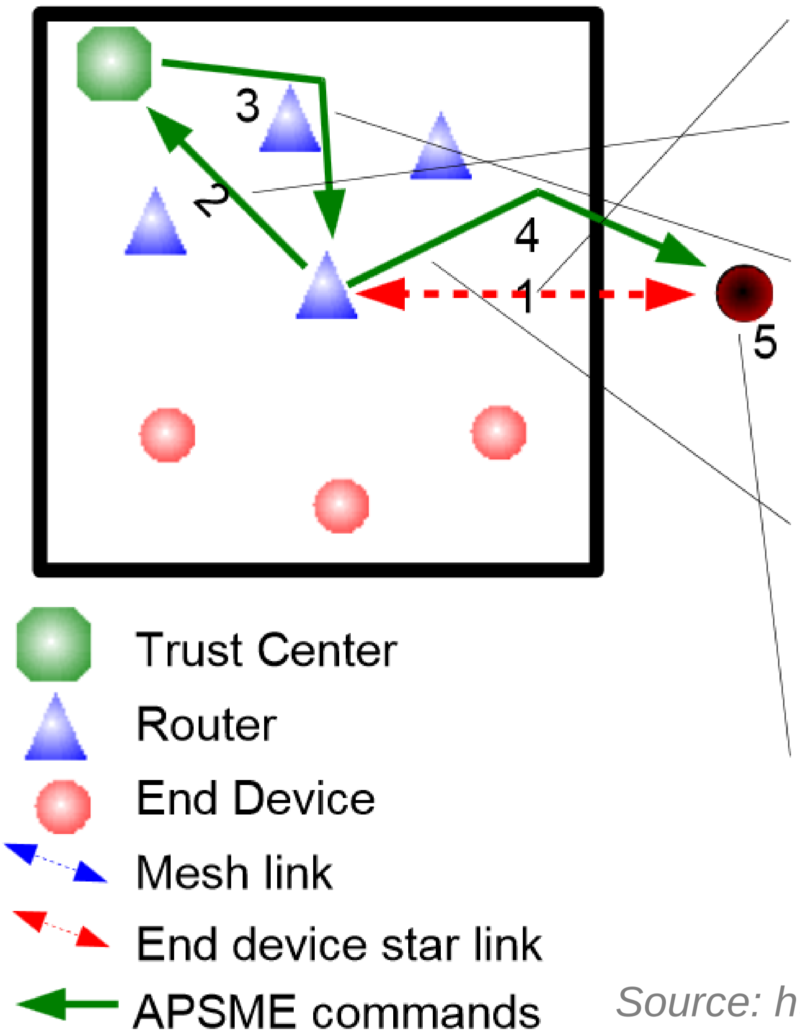
# ZigBee – characteristics


- Standardized as IEEE 802.15.4
  - ZigBee Alliance maintains current version
  - Niche between Bluetooth and WiFi
- Low cost, low power, mesh networking
  - Low power transmissions, smaller bitrate (250 kbit/s)
  - 10-100 meters (active-active communication mode)
  - Focus on sensors and control automation
- Various radio bands (2.4GHz), routing specifications
- Supports *star*, *tree* and *mesh* network topology
  - E.g., wireless sensor networks, up to 65000 nodes

# ZigBee network

- ZigBee Coordinator (ZC) / PAN coordinator
  - One coordinator per network
  - Responsible for establishment of network
  - Serve as repository for security keys
- ZigBee Router (ZR) / Coordinator
  - Pass data from one node to another (routing scheme)
  - Intermediate node in network
- ZigBee End Device (ZED) / Network device
  - Cheaper to produce, end (sensor) node
  - Cannot relay communication => can sleep => battery life

# Joining ZigBee network




1: Device does unsecured join

2: Router sends device update to TC for authorization

3: TC prepares network key transport for joining device secured with pre-configured TC link key shared between TC and joining device and tunnels it to router

4: Router unpacks tunneled network key transport and sends to device unsecured at network layer

5: Device retrieves network key from network key transport using pre-configured TC link key

*Source: https://docs.zigbee.org/zigbee-docs/dcn/09-5378.pdf*

# ZigBee keys

1. Pre-installation of master keys
   – Network key (shared by all), Link key (between 2 devices)
2. Transport of link keys
   – Trust center (ZC) sends link key to both nodes
3. Certificate-based key establishment
   – Trust center (ZC) facilitate establishment, no keys send between device and ZC
   – Elliptic Curve MQV key agreement scheme

# ZigBee cryptography

- Mostly based on symmetric cryptography
  - AES with 128b keys, master key, link key, network key(s)
  - Uses AES-CCM* mode for link layer encryption
    - encryption/integrity-only mode possible, MAC 4 bytes
- Certificate-based key establishment
  - Elliptic Curve MQV key agreement scheme
  - Requires certification authority

# SUMMARY

# Comparison: BT/NFC/ZigBee

- BT initially not for low-energy, but adapted (BT 4.x)
- NFC uses active-passive mode (locality)
- Pre-distributed keys vs. user interaction vs. locality
- ZigBee towards mesh networks
- Bluetooth LE also in direction of mesh networks

- (Next lecture will focus more on WSNs)

# Similarity between protocols (security)

- Easy eavesdropping
- Usage of proprietary (weak) ciphers (at beginning)
- Incorrect implementations of (complicated) standard
- Reuse of key stream ("never" need $2^{20}$ packets?)
- Problem of initial pairing (how to authenticate?)
- Brute-forcing usable/memorable/short PINs
- Problem of device tracking (unique device ID)
- Security generally getting better over time