Key Summary

02.04.2024 - SecPriPC - PD Dr. Zinaida Benenson

Table of Contents

1. Introduction

1.1 IoT / Pervasive Computing

Information processing (including sensing), networking and response anywhere, anytime

- Pervades everyday life
- Context-awareness
- Connected
- Sensors deliver information about physical environment
- Actors (actuators) response to physical environment

Enablers: Cheap, Small, Fast, Connected, Advanced UI

1.2 Security

To protect the right things, in a right way

Goals, Threats, Means: Protect what against whom, how

Security Goals:

1. Confidentiality: Protect data from unauthorized reading access
2. Integrity: Protect data from unauthorized changes
3. Availability: Make data always available on request by an authorized entity

→ Key is Authentication

1.3 Privacy

Past: The right to be left alone

Today: Informational self-determination; when, what, how and to whom data is passed on

1.4 S&P in IoT

- Different/new quality and quantity of data
- Profiling: habits, emotions (detected and processed via cameras and audio)
- Devices observe and interact with the physician environment
- Unprecedented data collection scale & attack surface

1.5 Security & Privacy Assessment

1. System description
2. Security & Privacy goals
3. Other goals
4. Attacker model
5. Tradeoffs

Attacker Model (ARID):

- Actors
- Resources
- Incentives (motivation)
- Damage

1.6 Design Principles for Pervasive Systems

- Default to Harmlessness: In case of system failure, degradation of services
- Be Deniable: Opt-out at any time
- Self-Disclosing: Ownership, Usage, Capabilities must be easy to find out
- Save user’s face: No harassment and embarrassment
- Save user’s time: High usability

1.7 Kerkhoff’s Principles

1. Practically
2. No security by obscurity
3. Key is communicable, retainable and changeable
4. Applicable to telegraphic correspondence
5. Portability
6. Easy to use

1.8 Abbreviations

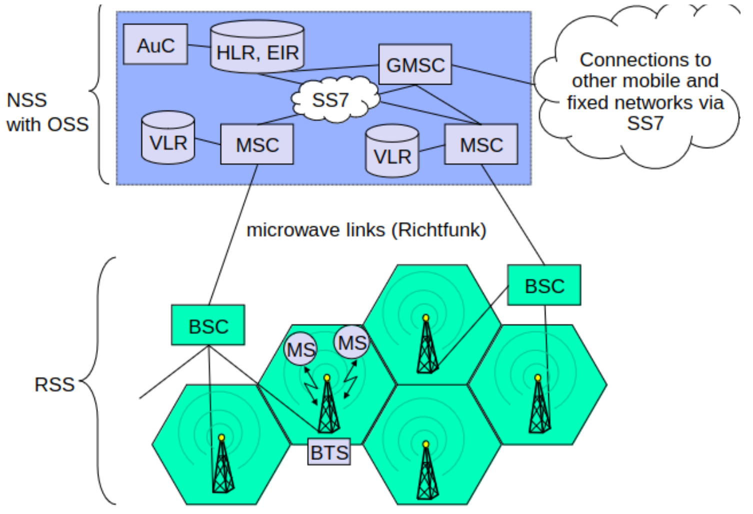
Abbreviation	Meaning
IoT	Internet of Things
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

2. Cellular

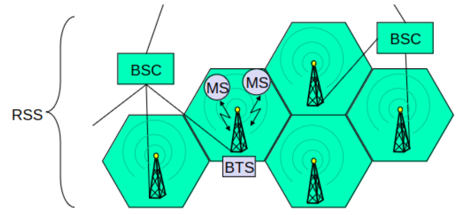
2.1 GSM Features

- Communication
- Mobility
- Worldwide connectivity
- High transmission quality

2.2 GSM Architecture

Subsystem:	**Structure:**
NSS (network and switching subsystem): Call forwarding, handover, switching	 NSS with OSS AuC HLR, EIR GMSC SS7 VLR MSC VLR MSC Connections to other mobile and fixed networks via SS7 microwave links (Richtfunk) RSS BSC BSC MS MS BTS
OSS (operation subsystem) Management of the network	
RSS (radio subsystem) Covers all radio aspects	

Radio Subsystem RSS

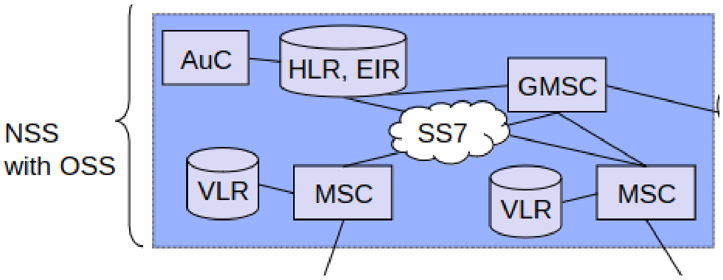
MS (mobile station): IMEI, SIM card	
BTS: base transceiver station	
BSC: base station controller – Manages several BTSs	

IMEI (international mobile equipment identifier)

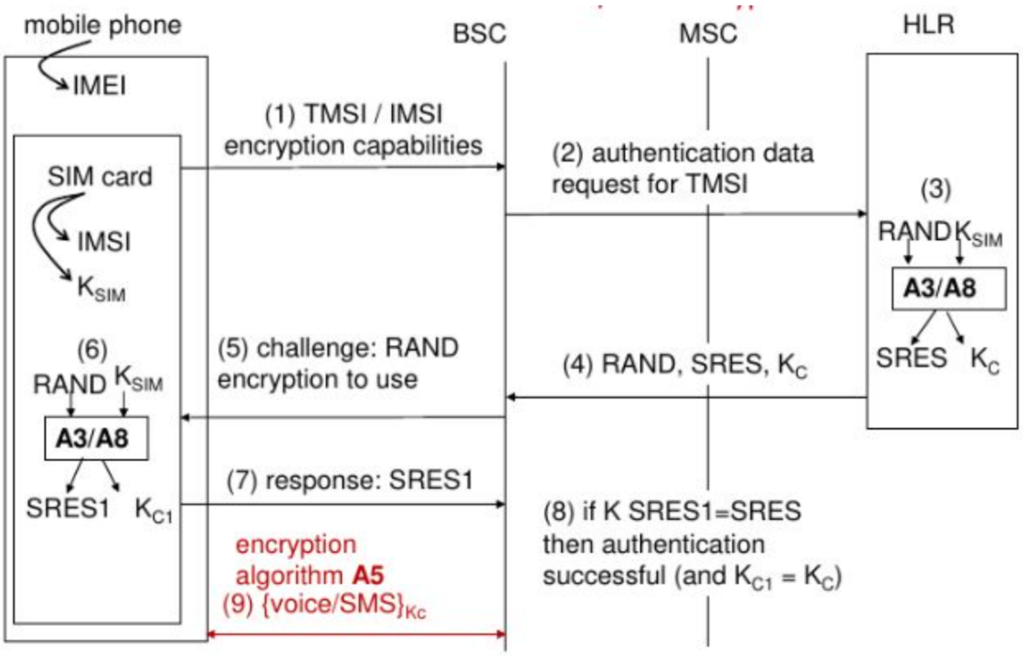
SIM card

- IMSI: International mobile subscriber identifier
- TMSI: Temporary mobile subscriber identifier

Network and Switching Subsystem NSS + Operation Subsystem OSS

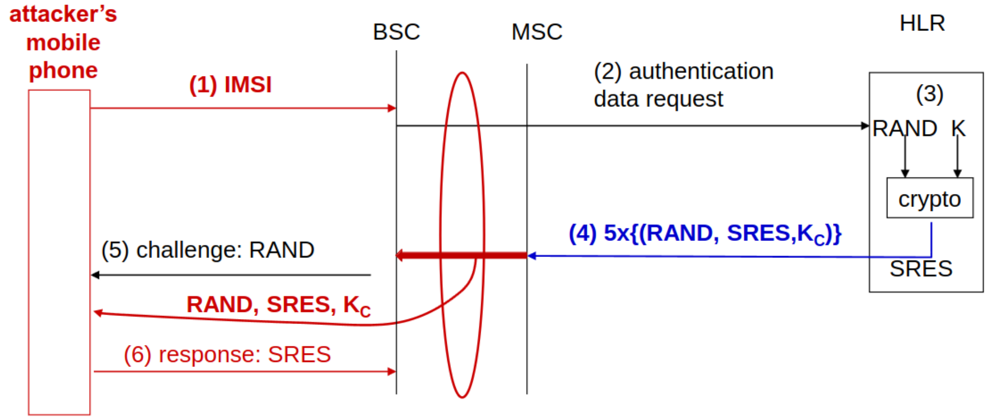
HLR: home location register – Knows the current MSC/**BSC** of the subscriber – **EIR:** equipment identity register contains IMEIs of all registered customers – **AuC:** authentication center contains symmetric cryptographic keys shared with the SIM cards	 NSS with OSS AuC HLR, EIR GMSC SS7 VLR MSC VLR MSC
MSC: mobile switching center – Manages several **BSCs**	
VLR: visitor location register (database at MSC Receives data from HLR for subscribers currently connected to its BTSs	
GMSC: gateway MSC	

2.3 GSM AUTHENTICATION



2.4 GSM Hacks

Free Call Hack by Ross Anderson

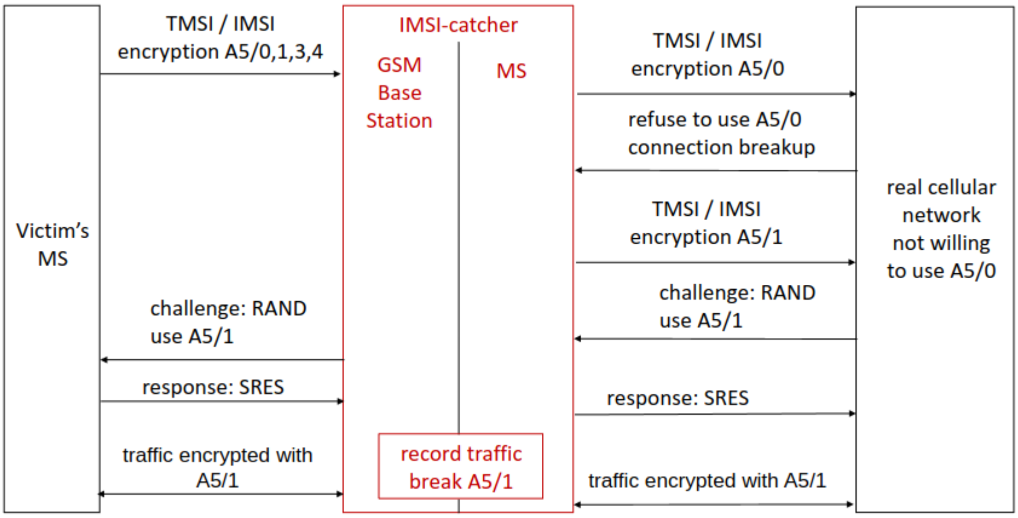


attacker eavesdrops on the unencrypted microwave link

In general Eavesdropping

By eavesdropping the unencrypted microwave link between BSC and MSC, K_C is known to the attacker.

Man in the middle attack with IMSI-Catcher

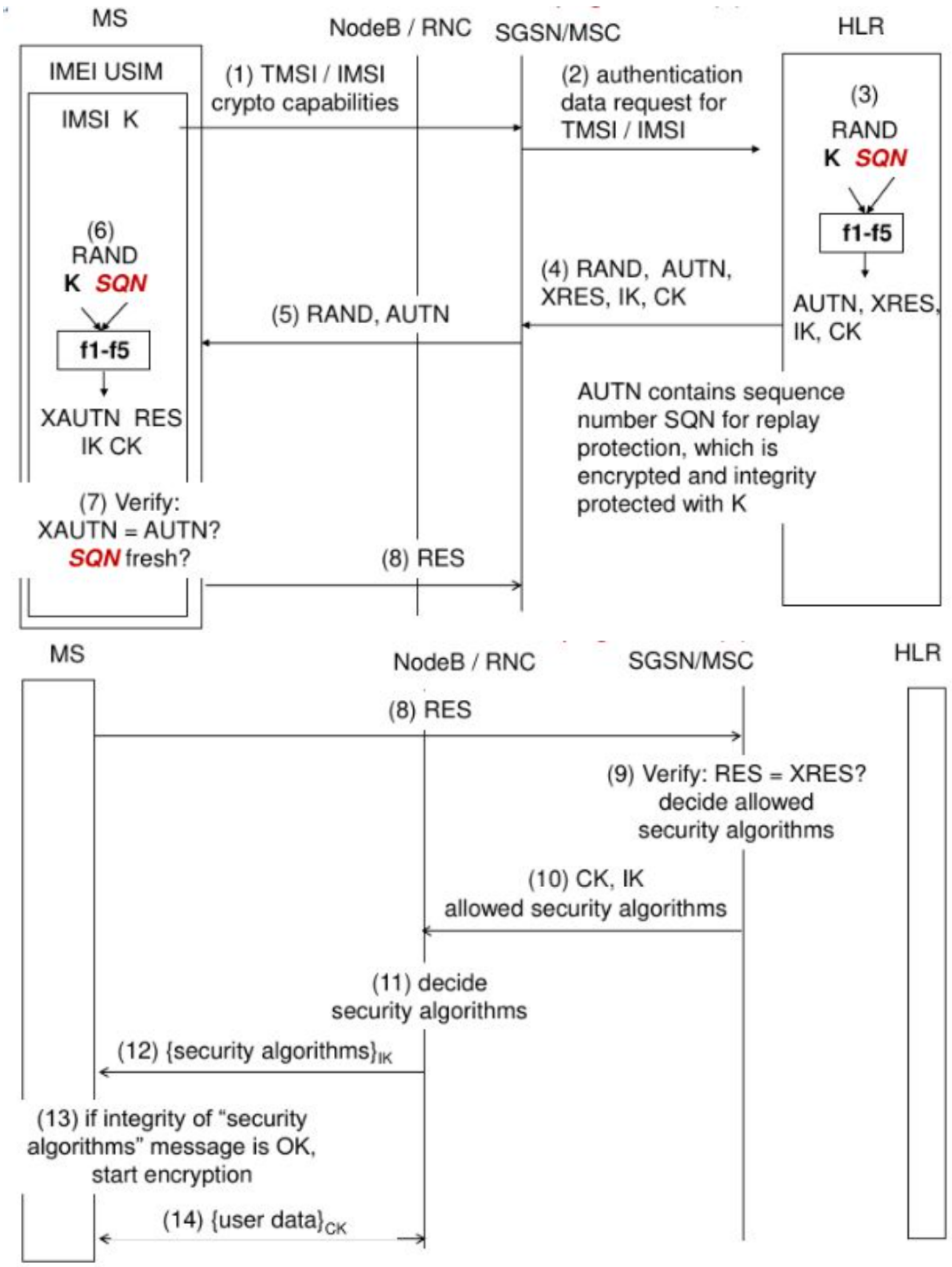

Victim's MS
TMSI / IMSI encryption A5/0,1,3,4
IMSI-catcher
GSM Base Station
MS
TMSI / IMSI encryption A5/0
refuse to use A5/0 connection breakup
TMSI / IMSI encryption A5/1
real cellular network not willing to use A5/0
challenge: RAND use A5/1
challenge: RAND use A5/1
response: SRES
response: SRES
traffic encrypted with A5/1
record traffic break A5/1
traffic encrypted with A5/1

A5/0 = Unencrypted; A5/1 can be broken in real time

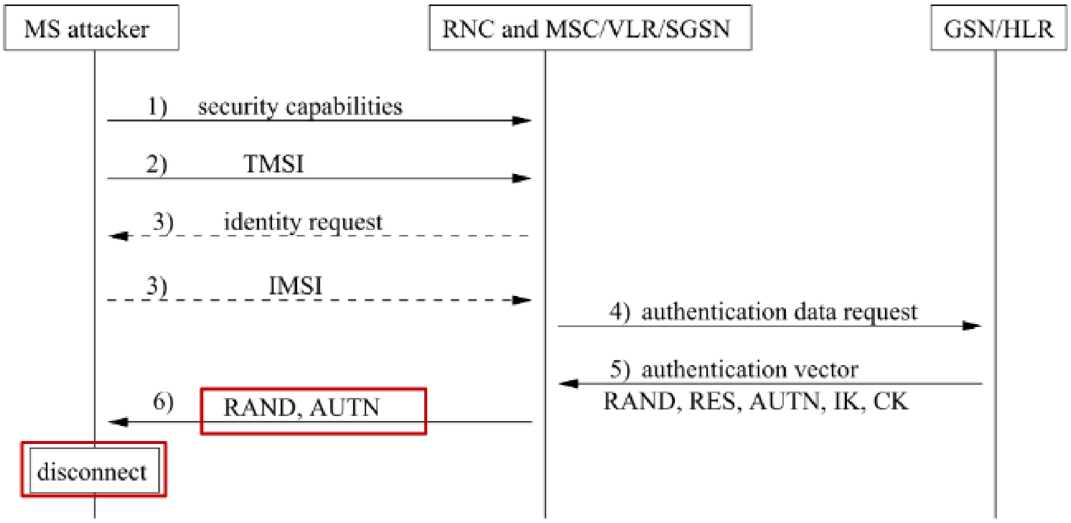
SIM Card Cloning

- Physical Extraction of private SIM-Key (very hard)
- Cryptographic Breaking of A3/A8-Encryption-Algorithms with plaintext-attack

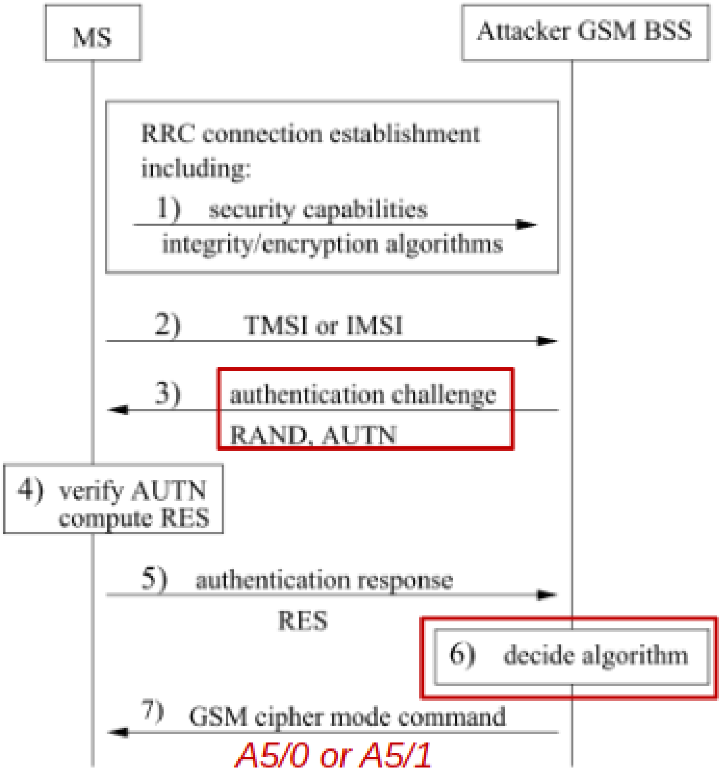
2.5 UMTS AUTHENTICATION



2.6 UMTS Degradation Attack



Now we have a valid AUTN


MS
Attacker GSM BSS
RRC connection establishment including:
1) security capabilities
integrity/encryption algorithms
2) TMSI or IMSI
3) authentication challenge
RAND, AUTN
4) verify AUTN
compute RES
5) authentication response
RES
6) decide algorithm
7) GSM cipher mode command
A5/0 or A5/1

Problems: Backward Compatibility & No Integrity check in GSM

2.7 Tracking

- Via SS7: Everyone can buy an access
  - Location Tracking: Getting IMSI and cell id of phone number
  - Eavesdropping: Getting Authentication and Encryption keys for TMSI
  - Manipulation: Forwarding IMSI traffic to specific network

    → Stealing money by redirecting SMSTan to attacker's network
- IMSI Catcher: Location targeting

2.8 LTE Impersonation Attack

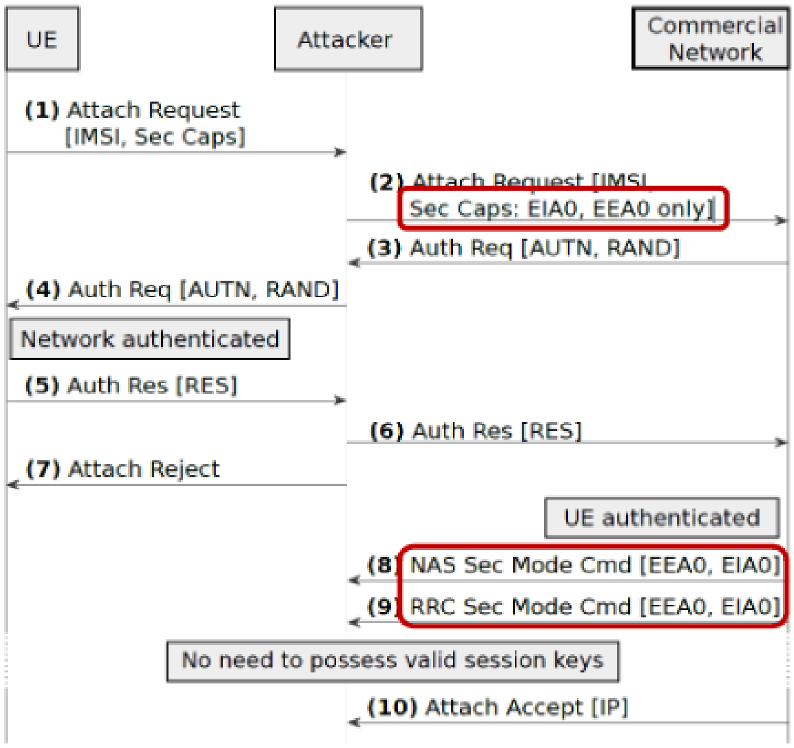

UE
Attacker
Commercial Network
(1) Attach Request [IMSI, Sec Caps]
(2) Attach Request [IMSI, Sec Caps: EIA0, EEA0 only]
(3) Auth Req [AUTN, RAND]
(4) Auth Req [AUTN, RAND]
Network authenticated
(5) Auth Res [RES]
(6) Auth Res [RES]
(7) Attach Reject
UE authenticated
(8) NAS Sec Mode Cmd [EEA0, EIA0]
(9) RRC Sec Mode Cmd [EEA0, EIA0]
No need to possess valid session keys
(10) Attach Accept [IP]

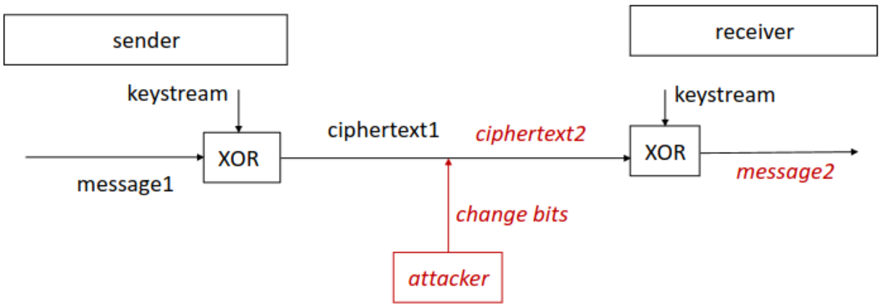
Figure 4: Impersonation attack exploiting the selection of EIA0 and EEA0 in a commercial network.

Setting the used encryption to EIA0 (= Unencrypted) as a man in the middle attacker.

2.9 LTE Attacks

- GUTI (= TMSI for LTE) rarely change → Location Tracking
- Location leaks via paging requests to eNodeBs
- Location leaks by impersonating eNodeBs: Can ask for signal strength of all surrounding cells
- Degradation attack

aLTEr attack: Voice calls use stream cipher with weak integrity check → Bitflips



ReVoLTE attack: Keystream reuse attack

Actions	Knowledge of attacker
1. Sniffing Call A: $m_1 \bigotimes keystream = c_1$	1. c_1
2. Making Call B: $m_2 \bigotimes keystream = c_2$	2. m_2, c_2
3. $c_2 \bigotimes m_2 = keystream$	3. $keystream$
4. Reversing: $c_1 \bigotimes keystream = m_1$	4. m_1

3. WiFi

3.1 WiFi Design Goals

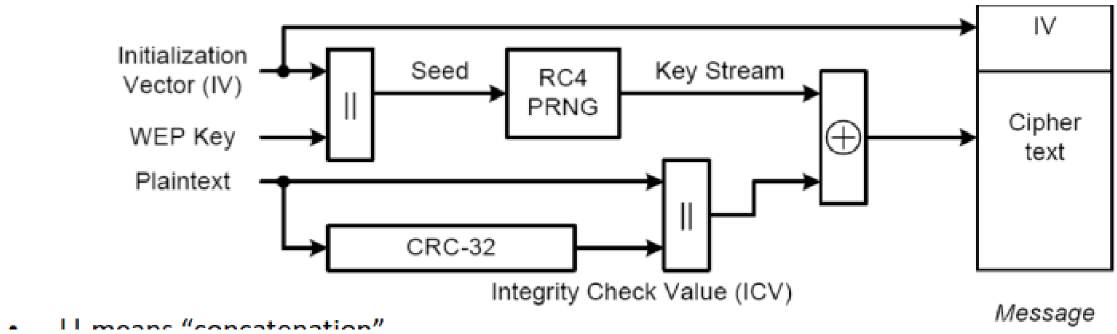
- Global, seamless operation (IEEE 802.11)
- Low power for battery use
- No special permission and license needed
- Robust
- Easy to use
- Low radiation
- Secure

3.2 Public Hotspots

- Hidden ESSID (Security by Obscurity) → Active scanning of device is detectable + Replay attack
- MAC Address Filtering → MAC-Address Spoofing
- Evil Twin attack

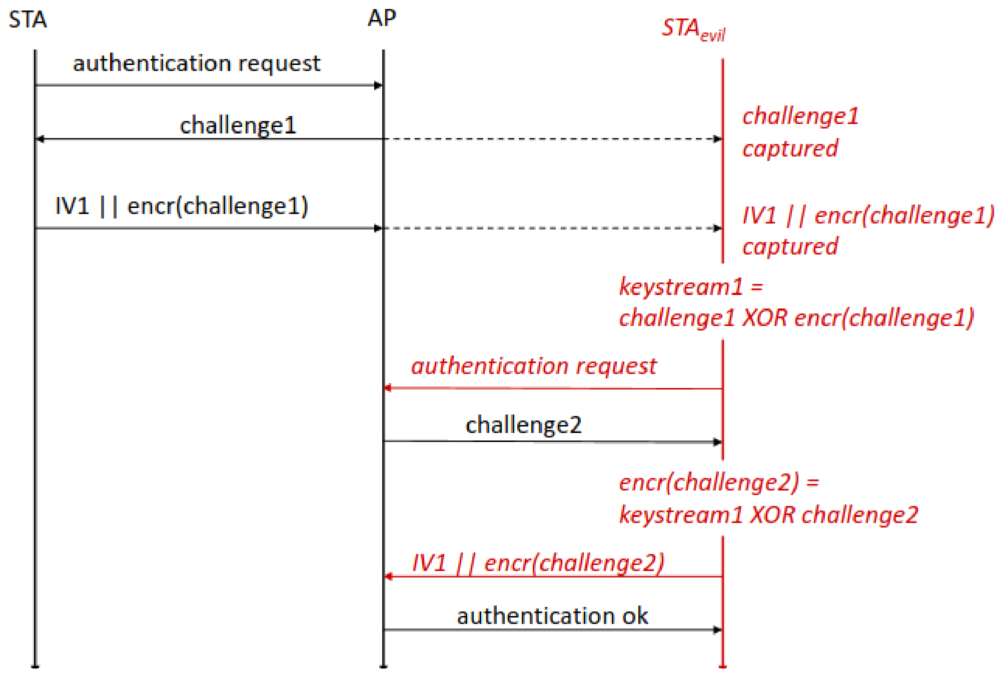
3.3 WEP (Wireless Equivalent Privacy)

Encryption (reversible stream cipher)



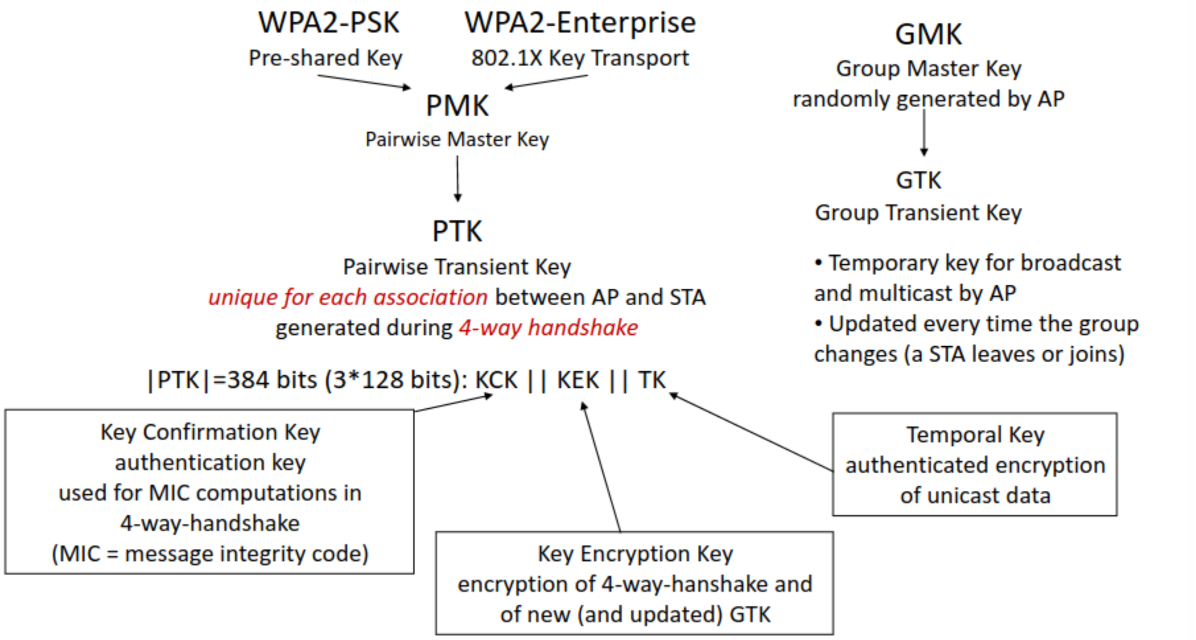
- CRC (Cyclic Redundancy Check) is very weak integrity check
- If IV is used more than once, attacker can make *two times pad* attack
  - PRNG restarts at 0 on reboot of device
  - IV restarts at 0 on reboot of device
  - IVs are too small and reused after around 7 hours
- RC4 is broken
- Master key is directly used for encryption (same for every device), no session keys

WEP replay attack



3.4 WPA2 (WiFi Protected Access 2)

Key Hierarchy

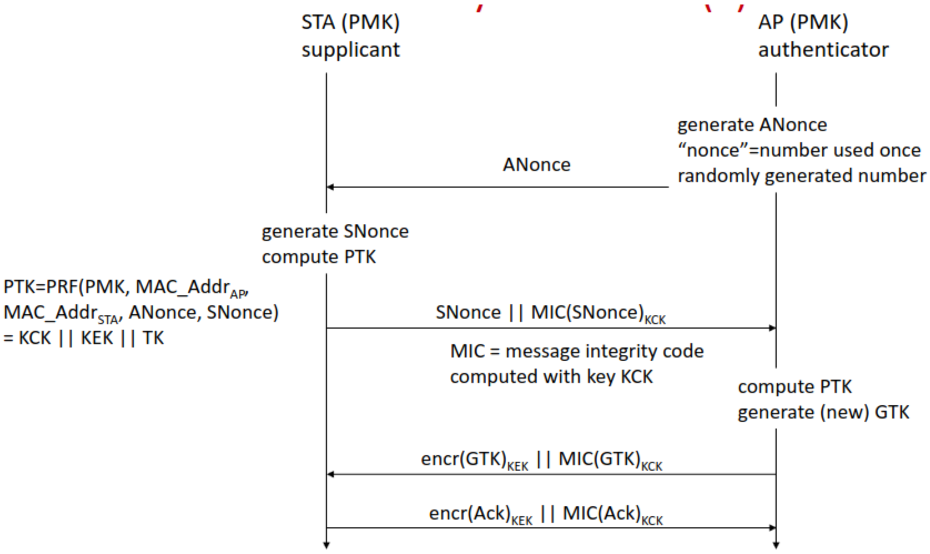

WPA2-PSK
Pre-shared Key
WPA2-Enterprise
802.1X Key Transport
PMK
Pairwise Master Key
PTK
Pairwise Transient Key
unique for each association between AP and STA
generated during 4-way handshake
|PTK|=384 bits (3*128 bits): KCK || KEK || TK
Key Confirmation Key
authentication key
used for MIC computations in
4-way-handshake
(MIC = message integrity code)
Key Encryption Key
encryption of 4-way-hanshake and
of new (and updated) GTK
Temporal Key
authenticated encryption
of unicast data
GMK
Group Master Key
randomly generated by AP
GTK
Group Transient Key
• Temporary key for broadcast and multicast by AP
• Updated every time the group changes (a STA leaves or joins)

- Master key is never used for encryption → Transient Keys
- Individual keys for each connected device

WPA2-PSK	WPA2-Enterprise

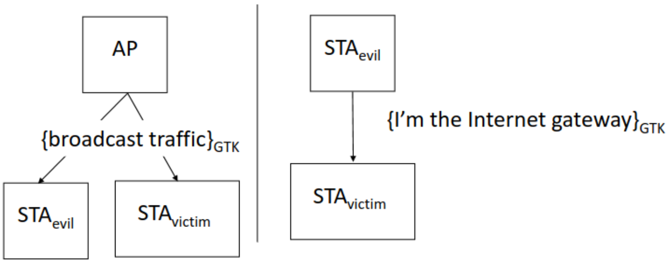
PMK = PBKDF2(WiFi-Password, ESSID)
ESSID = Salt
	PMK is sent encrypted from an authenticated Server tunneled through AP

Four-Way Handshake

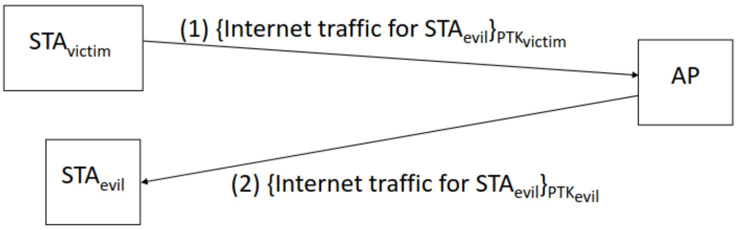

STA (PMK)
supplicant
AP (PMK)
authenticator
generate ANonce
"nonce"=number used once
randomly generated number
ANonce
generate SNonce
compute PTK
PTK=PRF(PMK, MAC_Addr$_{AP}$, MAC_Addr$_{STA}$, ANonce, SNonce) = KCK || KEK || TK
SNonce || MIC(SNonce)$_{KCK}$
MIC = message integrity code
computed with key KCK
compute PTK
generate (new) GTK
encr(GTK)$_{KEK}$ || MIC(GTK)$_{KCK}$
encr(Ack)$_{KEK}$ || MIC(Ack)$_{KCK}$

- WPA2-PSK Key Cracking: Capture handshake → Bruteforce password
- WPA2-PSK Insider attack: The same PMK is used for all devices → Capture Nonces → Decryption

Hole 196 attack in WPA2 (Insider attack also for WPA2-Enterprise)


AP
{broadcast traffic}$_{GTK}$
STA$_{evil}$
STA$_{victim}$
STA$_{evil}$
{I'm the Internet gateway}$_{GTK}$
STA$_{victim}$

ARP updates


STA$_{victim}$
(1) {Internet traffic for STA$_{evil}$}$_{PTKvictim}$
AP
STA$_{evil}$
(2) {Internet traffic for STA$_{evil}$}$_{PTKevil}$

Defenses: Static ARP tables or monitoring of ARP updates

WPA2 Key Reinstallation Attack – Man in the middle attack

- Attacker blocks 3rd message of four way handshake
- Device reinstalls the PTK in stream cipher mode (CTR)
- Attacker blocks these messages, more are sent by device with repeated IVs → two times pad attack

Dragonblood attack in WPA3: Downgrade attack to WPA2

4. ZigBee – IEEE 802.15.4

4.1 Design Goals

- Longer range than Bluetooth
- Lower power, data rate and complexity
- Multi-month to multi-year battery life
- Only Sensor data, control commands, No voice or multimedia
- Small code size, less operations to implement

4.2 Nodes & Network

Past: Application Profiles; Problem: Several profiles had to be supported

Today: ZigBee PANs (Personal Area Networks)

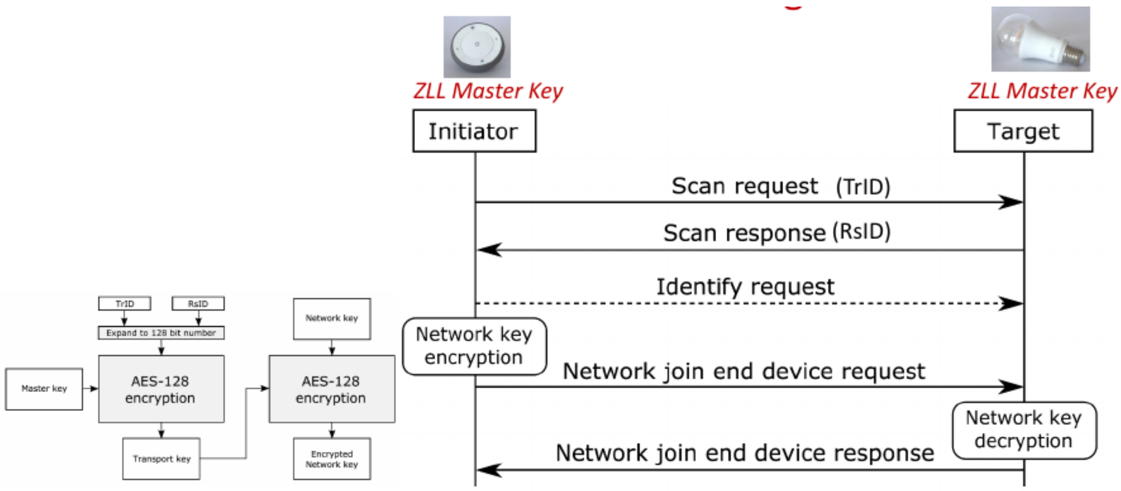
- ZigBee Coordinator (C) in centralized mode
- ZigBee Router (R)
- ZigBee End Device (E)

4.3 ZigBee Authentication

Centralized	Distributed
EZ-Mode only! Individual device key is transferred to Coordinator by scanning QR code with app. If not available both share a public *global trust center link key*	EZ-Mode vs. Touchlink Both parties share an NDA-protected key. The one for Touchlink has been leaked in 2015.

Whole Network uses one *Network Key* for symmetric data encryption.

4.4 Touchlink Commissioning



Network key is encrypted without integrity protection (AES-ECB) → device joins any network with any network key

4.5 Touchlink Inter-PAN attacks

Inter-PANs are commands that are sent after a *scan request* and *scan response*, but before the authentication handshake and mostly work even if the device is already connected to a network.

Inter-PANs are accepted if the device received a scan request with the same *TrID* before + Bug in proximity check (factory reset is missing proximity check)

These can be used to open smart door locks, DoS or ransom attacks

- Identify Action Attack: Send “Identify request”
- Reset to Factory Attack: Send “Reset to factory request”
- Permanent Disconnect Attack: Send “Network update request”
  - to change wireless channel of target
  - to join target to garbage network, as network authentication is lacking integrity protection

4.6 Touchlink Attack using the leaked key

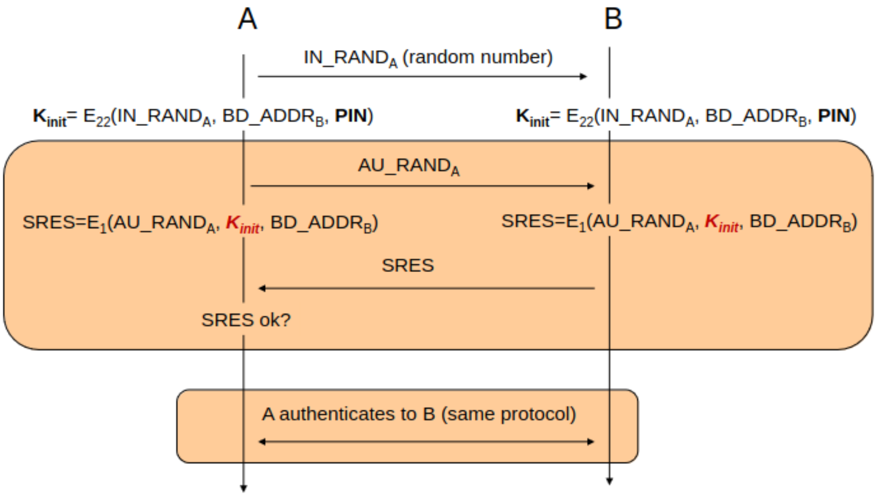
- Hijack: Send network update to connect target to attacker’s network and remote control it
- Network key extraction: Capture authentication handshake and decrypt it

5. Bluetooth & Device Pairing

5.1 Bluetooth 1.0 – 2.0 Key Hierarchy

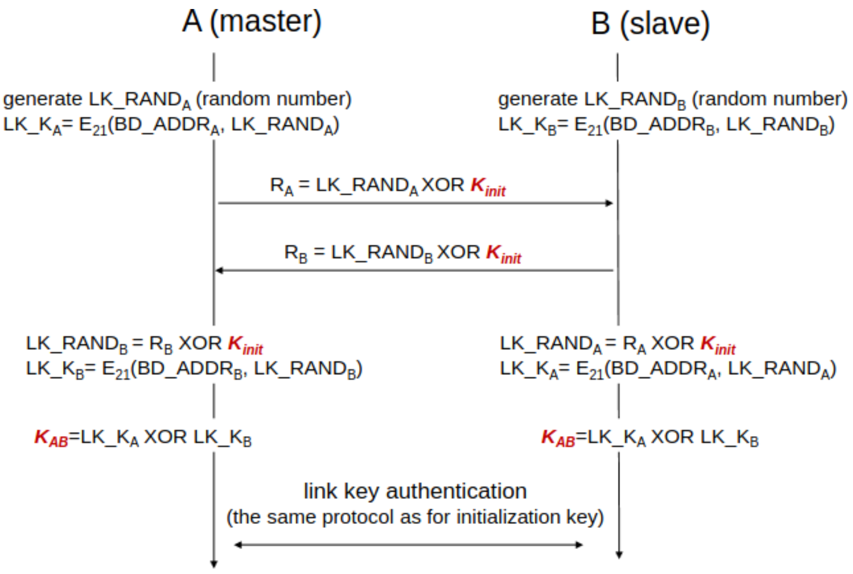
Initialization Key (PIN)	Link Key	Encryption Key
Temp key for handshake	Generated from Init-Key, Checked on Re-pairing	Generated from Linked key, Data encryption

Initialization Key Generation

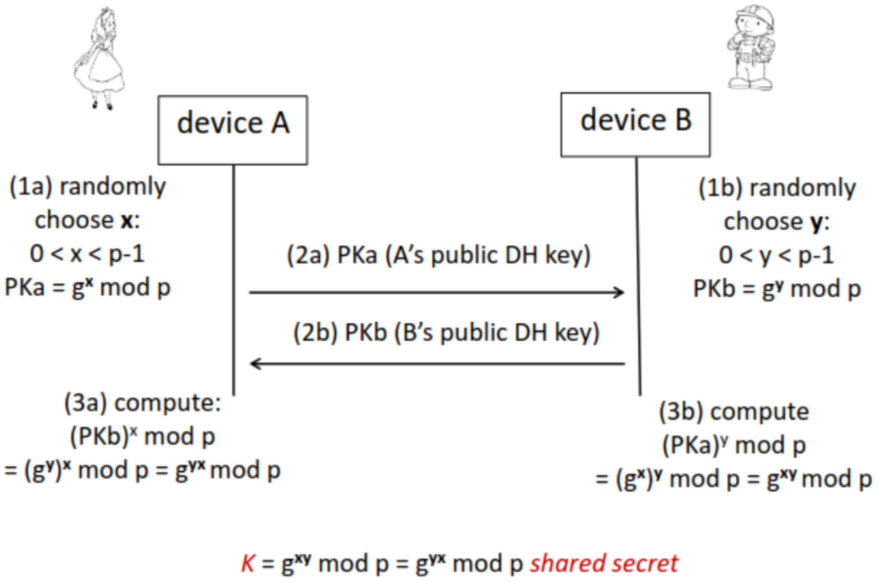

A
B
IN_RAND_A (random number)
$K_{init}= E_{22}(IN\_RAND_A, BD\_ADDR_B, \mathbf{PIN})$
$K_{init}= E_{22}(IN\_RAND_A, BD\_ADDR_B, \mathbf{PIN})$
AU_RAND_A
$SRES=E_1(AU\_RAND_A, K_{init}, BD\_ADDR_B)$
$SRES=E_1(AU\_RAND_A, K_{init}, BD\_ADDR_B)$
SRES
SRES ok?
A authenticates to B (same protocol)

PIN can be cracked by brute force → Decryption + Impersonation attack through “Forgot Key”

Link Key Generation


A (master)
B (slave)
generate LK_RAND_A (random number)
$LK\_K_A= E_{21}(BD\_ADDR_A, LK\_RAND_A)$
generate LK_RAND_B (random number)
$LK\_K_B= E_{21}(BD\_ADDR_B, LK\_RAND_B)$
$R_A = LK\_RAND_A$ XOR K_{init}
$R_B = LK\_RAND_B$ XOR K_{init}
$LK\_RAND_B = R_B$ XOR K_{init}
$LK\_K_B= E_{21}(BD\_ADDR_B, LK\_RAND_B)$
$LK\_RAND_A = R_A$ XOR K_{init}
$LK\_K_A= E_{21}(BD\_ADDR_A, LK\_RAND_A)$
$K_{AB}=LK\_K_A$ XOR LK_K_B
$K_{AB}=LK\_K_A$ XOR LK_K_B
link key authentication
(the same protocol as for initialization key)

5.2 Unauthenticated Diffie-Hellman


device A
device B
(1a) randomly choose **x**: $0 < x < p-1$ $PKa = g^x \bmod p$
(1b) randomly choose **y**: $0 < y < p-1$ $PKb = g^y \bmod p$
(2a) PKa (A's public DH key)
(2b) PKb (B's public DH key)
(3a) compute: $(PKb)^x \bmod p = (g^y)^x \bmod p = g^{yx} \bmod p$
(3b) compute $(PKa)^y \bmod p = (g^x)^y \bmod p = g^{xy} \bmod p$
$K = g^{xy} \bmod p = g^{yx} \bmod p$ *shared secret*

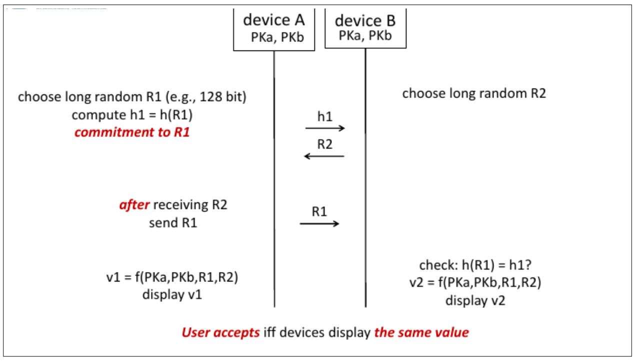
Save against: Passive Eavesdrop

Vulnerable against: Man in the middle, Evil Twin

5.3 Authenticated Diffie-Hellman (Bluetooth 2.1+)

Diffie-Hellman with authenticity check via a different OOB channel or secret

- Manual authentication (MANA): PIN, Push button, compare strings
- Numeric comparison
- Passkey Authentication (PIN input)
- Physical: NFC, USB
- QR Code scanning (Seeing is believing)
- Shake well before use
- Network in a box (Infrared)
- Touch to pair
- Wanda detect



Bluetooth allows degradation to unauthenticated Diffie-Hellman, if rest fails → Degradation attack

6. RFID

6.1 Applications

- Security and Safety (access control, verification, e-documents)
- Tracking (supply chain, hospital)
- Authenticity (medicine)
- Electronic Payment (ApplePay)

Supply chain management

EPC (Electronic Product Code): Unique identifier for every physician object

Retail

Smart shopping carts, smart shelves → Smart payment, stock accuracy, theft detection

6.2 Collision detection

Query Tree Protocol	Query Slot Protocol
Reader walks through binary tree, asking for Tag IDs that start with the current number	Reader creates 2^Q slots, Tags join them randomly. Reader asks for a random 16 bit number for each Tag in each slot. If multiple Tags in one slot, increase Q and unchecked Tags reorganize.
Deterministic Tags have no state, only send ACKs	Probabilistic, very efficient Tags only send RN16 numbers
Reader exposes/asks for all Tag IDs	Tags need random number generator hardware and storage for saving checked flag

6.3 Security & Privacy Threats

Security

- Corporate espionage: Unauthorized readers
- Competitor/Market analytics: Consumer analysis through unauthorized readers
- Vulnerable Infrastructure: Jamming
- RFID malware: SQL injection as Tag-ID response

Privacy

- Actions, Preferences, Identity and Location Tracking

6.4 Countermeasures

- Killing: Non-recoverable, killing PIN
- Covering: Physical
- Blocking: Jamming → Dozens of Collisions
- RFID Bill of Rights

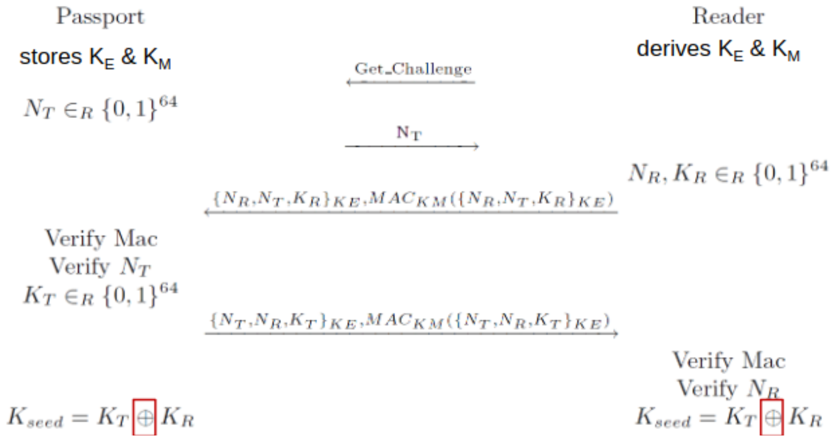
6.5 Ghost-and-Leech attack

Man in the middle attack on unauthenticated Diffie-Hellmann:

Relay/Intercept authentication by emulating RFID card and reader (missing integrity check), attacker needs to be near victim

Defense: Proximity check through Distance-Bounding-Protocols, but they are much slower

6.6 E-Passport


Passport
stores K_E & K_M
Reader
derives K_E & K_M
Get_Challenge
$N_T \in_R \{0,1\}^{64}$
N_T
$N_R, K_R \in_R \{0,1\}^{64}$
$\{N_R, N_T, K_R\}_{KE}, MAC_{KM}(\{N_R, N_T, K_R\}_{KE})$
Verify Mac
Verify N_T
$K_T \in_R \{0,1\}^{64}$
$\{N_T, N_R, K_T\}_{KE}, MAC_{KM}(\{N_T, N_R, K_T\}_{KE})$
Verify Mac
Verify N_R
$K_{seed} = K_T \boxed{\oplus} K_R$
$K_{seed} = K_T \boxed{\oplus} K_R$

- Symmetric encryption based on personal data in the passport
- Passport has to send message first to reader to protect replay attacks

Information gathering attack

Passports have different responses based on the country

Cloning Attacks

Possible to generate keys if some passport information is known.

Tracking via replay attack

1. Capture handshake with Nonces
2. Spoof MAC address → A failed MAC check is reported sooner than a failed nonce check
   (a failed Nonce means, that the MAC check before was correct → Location tracking)

7. Lessons Learned (duplicates left out)

7.1 Cellular

- No Security by Obscurity
- Mutual authentication of both parties
- Crypto algorithms easy to change (no hardware algorithms)
- Transparent development process
- Backwards compatibility leaves vulnerabilities → Downgrade attacks
- Specification must warn about non-secure modes and define security goals and threat model
- Analytical measures can impact security and privacy

7.2 WiFi

- No master keys for data encryption → Session keys + key management
- Secure integrity checks
- User passwords will be weak → Avoid user generated passwords
- Insider attacks must be considered
- Unexpected interaction with insecure 3rd party protocols (ARP in Hole 196)
- Specification clear and not ambigious

7.3 ZigBee

- No proximity verification using signal strength
- Encryption only with integrity protection

7.4 Bluetooth

- Security assumptions are important
- Do not rely on user generated passwords, they are not random
- Avoid user interaction

7.5 RFID

- Think before use
- Also secure backend
- Security & Privacy issues without cryptography
- Implementation mistakes & Hidden functionality

E-Passport

- Crypto keys must be (pseudo-)random
- Side channel attacks: Different responses based on country, timing of messages

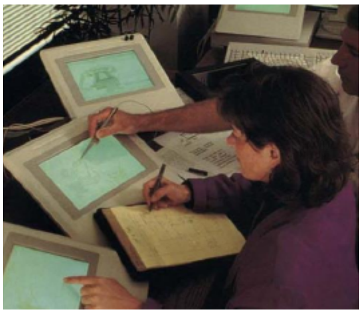
8. Socio-Tech Lectures

8.1 Ubiquitous Computing

“Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop.

Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.” — Mark Weiser

8.2 Mark Weiser’s Vision

Tabs	Pads	Active Bagdes
		
Smartphone in cheap, Only displays information	Non-individualize e-papers	Smartcard for identification
Computers vanish into the background become invisible, machines that fit the human environment → No multi-use personal computer, many single-use computers for one user instead		

8.3 Case Study: Smart Cities in India

- Smart parking did not fit city needs
- It also did not fulfil expectations
- Suffered from a lack of demand
- Application of technical solutions to problems that cannot be solved solely through technological means – “technological solutionism”
- Command and Control Centers in almost all city plans of Smart Cities Mission

8.4 Cybernetics

Cybernetics is a central historical precursor and guiding idea behind many contemporary phenomena, especially smart cities, but also other forms of ubiquitous computing.

- Derived from the Greek word „kubernetes“, or steersman
- Developed by Norbert Wiener during and after WWII
- Foundational for cybernetic contributions in all kinds of disciplines and fields
- Intellectual movement in the 1950s/60s/70s

- A specific procedure for the production of truth through feedback of information

Properties:

- Systems thinking
- Governance as steering
- Horizontal networking
- Organicist thinking
- Frictionless, apolitical understanding of process