SecPriPC Zusammenfassung

Made by *El Presidente*

Keine Garantie auf Falschheit oder Korrektheit.
Das ist meine allerletzte Zusammenfassung, nutze sie weise! (oder auch nicht).

!!!
Für alle die sagen dass der Kram hier zu lang ist: Schaut ans Ende, dort sind Klausurfragen aus der Probeklausur und den Folien inklusive Antworten drinnen.
!!!

1. Introduction

Pervasive Computing

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

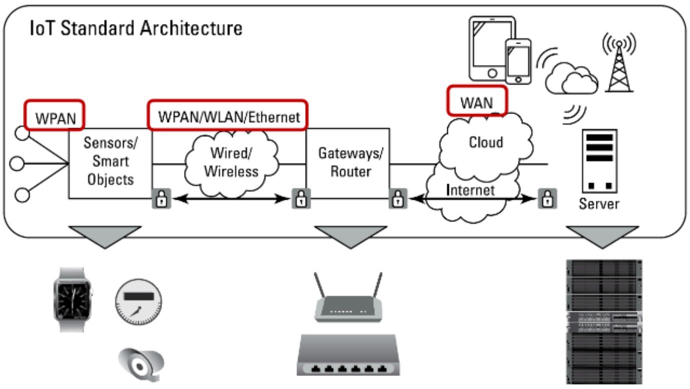
- Pervades everyday life
- Context-awareness
- Sensors deliver information about physical environment
- Actors (actuators) response to physical environment

4 Enablers of Pervasive Computing

- Miniaturization of computing & storage
- Mobility through wireless networking
- Computing & storage steadily getting cheaper and faster
- Advanced user interfaces

IoT (Internet of Things)

= Things of our daily live contain computers and are connected via e.g. the Internet or other networks to exchange data and information. Additionally, sensors of the devices deliver information about the physical environment. Actors (actuators) respond to the physical environment.



Wireless Technology

WAN = Wide Area Networks: GSM / UMTS / LTE,4G,5G / SS7

WLAN = Wireless Local Area Network: Wi-Fi

WPAN = Wireless Personal Area Network: Bluetooth, ZigBee, RFID

Definition: Security

Protect the right thing in a right way (Ross Anderson)

1. Goals: what to protect
2. Threats: against what/whom to protect

3. Means: how to protect

Security Goals (CIA) and Authentication and Authorization

Confidentiality
- Protect data from unauthorized reading access

Integrity
- Protect data from unauthorized changes

Availability
- Make data always available on request by an authorized entity

Authentication
- Distinguish between authorized and unauthorized entities

Evaluating Attackers

1. Actors: who would be interested in attacking?
2. Resources: Skills, time, money, technology, manpower
3. Incentives: why would they attack?
4. Damage: consequences of an attack

Privacy

First definition: bodily / territorial privacy (The Right to be let alone)
Information Privacy: is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.

Security & Privacy in IoT

1. Data Types: Location, video, audio, …. => Profiling/Habits => Territorial Privacy issue, as IoT devices are small and everywhere
2. Data/System Accessibility: System ownership, devices are always on, data collection & attack surface
3. Interaction with devices/systems: Invisible systems/interactions, availability + integrity more important than confidentiality

5 Assessments of S&P

1. What is the system? (Assets, Stakeholders, what informations can be extracted from the system)
2. S&P Goals (for stakeholders and assets)
3. Other goals for stakeholders? (functionality, costs, …)
4. What is the attacker model? (see “Evaluating Attackers”)
5. What are the S&P trade-offs? (all these goals, likelihood and costs)

5 Design principles for pervasive systems (by Adam Greenfield)

1. Default to harmlessness: Physical, psychic, financial safety always warranted

   - Even in case of system failure!
   - Graceful degradation of services
   - Failure of a part of a system causes as few service degradation as possible
2. Be self-disclosing
   - Ownership, usage, capabilities must be easy to find out
3. Save users’ face
   - Never embarrass (blamieren) or harass (belästigen) the user
4. Save users’ time
   - Provide high usability
5. Be deniable
   - Opt-out always possible

Need to know:

Pervasive Computing (5 parts)

IoT Enablers (4 parts)

IoT (5 parts)

WAN (+3/4 Examples)

WLAN (+ 1 Examples)

WPAN (+ 3 Examples)

Security (Definition)

Security Goals (CIA) + Authentication

4 Parts of evaluating attackers

Privacy (3 definitions)

S&P in IoT (3 parts)

5 steps of S&P Assessment

5 Design Principles of pervasive systems (A. Greenfield)

2. Cellular: GSM

Privacy in Mobile Networks

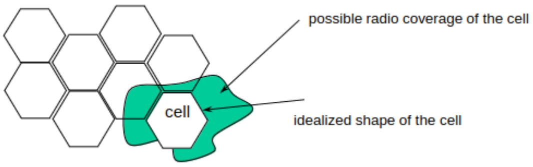
- Cellular telephony is a ubiquitous system
- Controlling telecom operators is the key to full control over citizens

GSM and its 4 features

GSM (Global System for Mobile Communications)

1. Communication: voice and data services
2. Total mobility (international, different providers)
3. Worldwide connectivity
4. High transmission quality (audio quality and reliability even when using trains/cars)

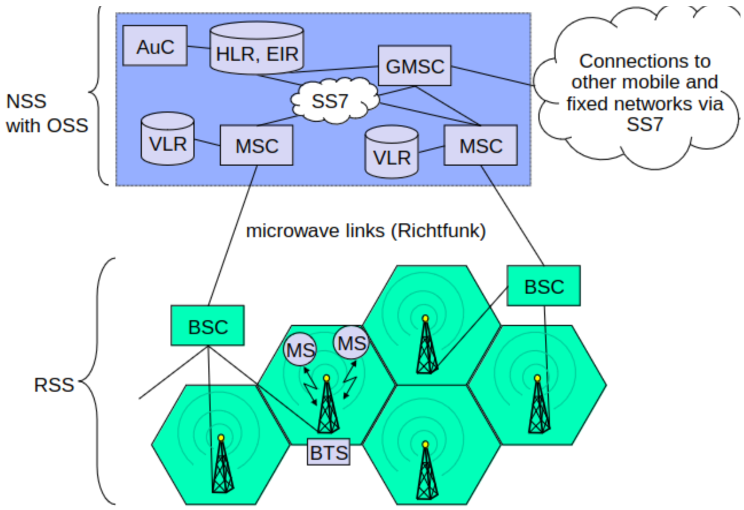
GSM cellular networks


possible radio coverage of the cell
cell
idealized shape of the cell

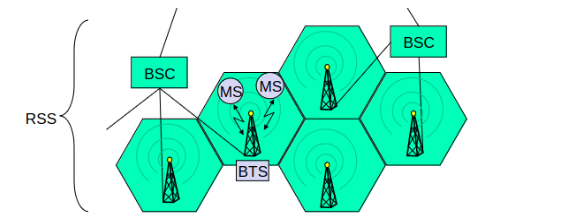
segmentation of the area into cells

- Use of several carrier frequencies
- Cell sizes: from some 100 m up to 35 km depending on user density, geography,
- transceiver power etc
- Handover: mobile user changes cell, handover of connection to the neighbor cell

GSM Architecture

Subystem:	**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

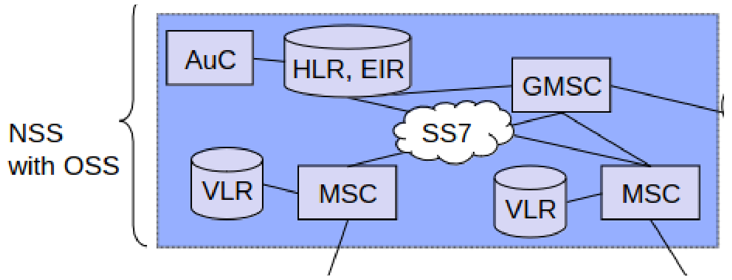
MS (mobile station): IMEI, SIM card	
BTS: base transceiver station	
BSC: base station controller – Manages several BTSs	

IMEI (international mobile equipment identifier)

SIM card

- IMSI: international mobile subscriber identifier, 64 bit
- Symmetric cryptographic key for authentication and session key agreement for voice and SMS encryption

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	

BTS = Base Transceiver Station

Often on buildings as antennas

BSC = Base Station Controller

MSC = Mobile Service Switching Center

both inside buildings or on small radio towers (boxes, medium size)

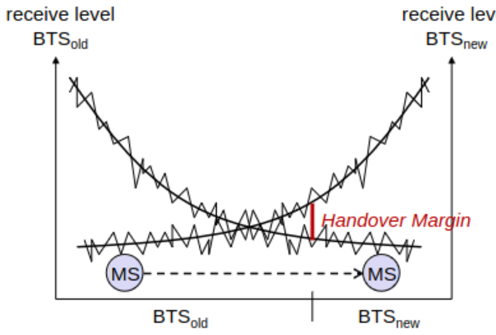
Network Subsystem + Operation Subsystem

Switching units

Management data bases

Monitoring

Handover Decision

• Mobile station (MS) regularly checks the signal strength of all base transceiver stations (BTS) in its range. • If a mobile station (MS) exits cell region, it signals a change of BTS to the mobile service switching center (MSC). • This results in a changed region id in the home file of the user in the home location register (HLR). • Mobile station (MS) always connects to the most powerful base transceiver stations (BTS).	

Call Setup

If somebody calls a mobile station ..	If the mobile station wants to set up a call ..
– the system looks into the home file of the user – the system identifies the cell region – all base stations in the cell region broadcast a call request to the mobile station – mobile station answers to request – call is set up over base station with the best reception characteristics	– it connects to the current base station and sends the phone number of the called party – the system looks up the current cell region of the called party – the system sets up the call over the corresponding switching stations

GSM S&P Assessment

Stakeholders	Telecom. companies Customers

	Device Manufacturers State
System goals	Fast connection establishment, calls to fixed telephone networks, roaming, text messages, online banking, transparent billing system
Assets and collected data	Devices and their content, communication metadata (including customer location), communication content, billing, infrastructure
Security goals	CIA of all assets + Non-repudiation of calls, privacy of subscribers

GSM Threats and Attackers

Providers	Threats	“free” calls, disruption of communication,
	Attackers	customers, criminals and saboteurs, business rivals
Customers	Threats	paying for alien or non-existing calls, loss or theft of devices, eavesdropping and tracking
	Attackers	criminals, providers, intelligence agencies, (totalitarian) states
Manufacturers	Threats	loss of availability, vandalism
	Attackers	saboteurs, terrorists, business rivals
State	Threats	untraceable / anonymous calls, no possibility for wiretapping
	Attackers	criminals, terrorists, other states

4 GSM Security Features

- Subscriber identity (IMSI) authentication
- Subscriber data confidentiality
- Subscriber location privacy
- Signaling information confidentiality (IMSI, IMEI, phone numbers)

Security features can be implemented in BSC and/or MSC

GSM Security Risks

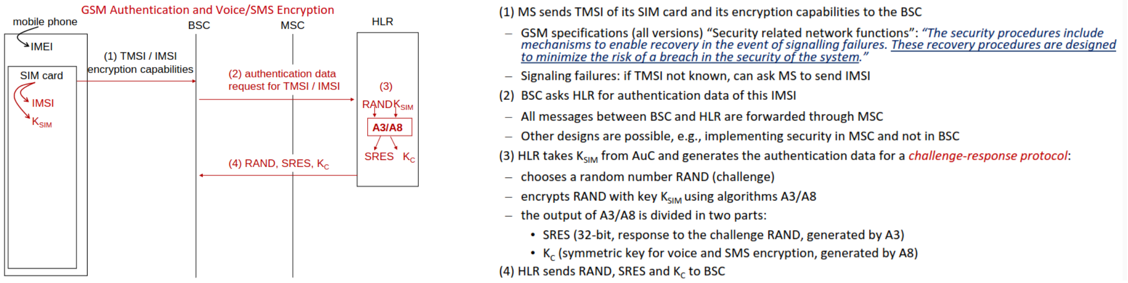
- misuse of their resources by unauthorized persons using manipulated Mobile Stations, who try to impersonate authorized subscribers; and
- eavesdropping of the various information which are exchanged on the radio path

GSM Authentication and Voice/SMS Encryption

- Identifiers: Mobile phone (IMEI), SIM card: IMSI, TMSI (temporary mobile subscription identity)
- Symmetric algorithms: In the past mobile phones were not powerful enough for asymmetric public key crypto

- Authentification: A3 and A8 Algorithm, with pre-shared key K_{SIM}
- Encryption: A5, session key K_C 64 or 54 bits (key is established for every connection)

Steps 1-4



(1) MS sends TMSI of its SIM card and its encryption capabilities to the BSC

– GSM specifications (all versions) "Security related network functions": *"The security procedures include mechanisms to enable recovery in the event of signalling failures. These recovery procedures are designed to minimize the risk of a breach in the security of the system."*

– Signaling failures: if TMSI not known, can ask MS to send IMSI

(2) BSC asks HLR for authentication data of this IMSI

– All messages between BSC and HLR are forwarded through MSC

– Other designs are possible, e.g., implementing security in MSC and not in BSC

(3) HLR takes K_{SIM} from AuC and generates the authentication data for a *challenge-response protocol*:

– chooses a random number RAND (challenge)

– encrypts RAND with key K_{SIM} using algorithms A3/A8

– the output of A3/A8 is divided in two parts:

- SRES (32-bit, response to the challenge RAND, generated by A3)
- K_C (symmetric key for voice and SMS encryption, generated by A8)

(4) HLR sends RAND, SRES and K_C to BSC

IMEI: International Mobile Equipment Identity. It is a unique identification number assigned to every mobile device

IMSI (International Mobile Subscriber Identity): This is a unique identification associated with a SIM card in a mobile device.

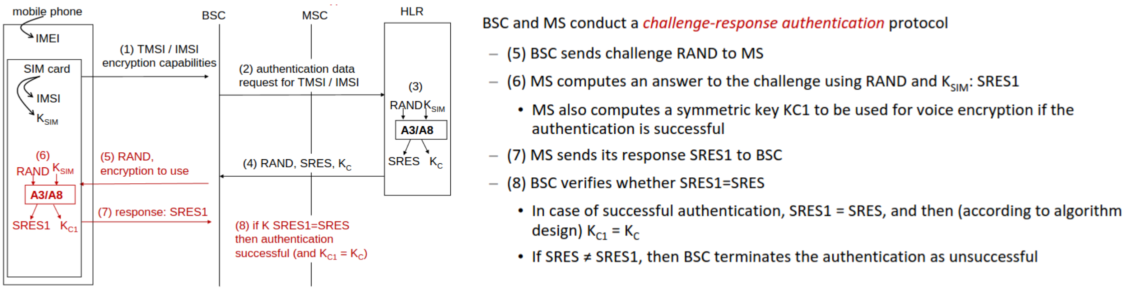
TMSI (Temporary Mobile Subscriber Identity): TMSI is a temporary identifier assigned to a mobile device by the network when it connects to the network.

BSC: Base Station Controller

MSC: Mobile Switching Center

HLR: Home Location Register

Steps 5-8



BSC and MS conduct a *challenge-response authentication* protocol

– (5) BSC sends challenge RAND to MS

– (6) MS computes an answer to the challenge using RAND and K_{SIM}: SRES1

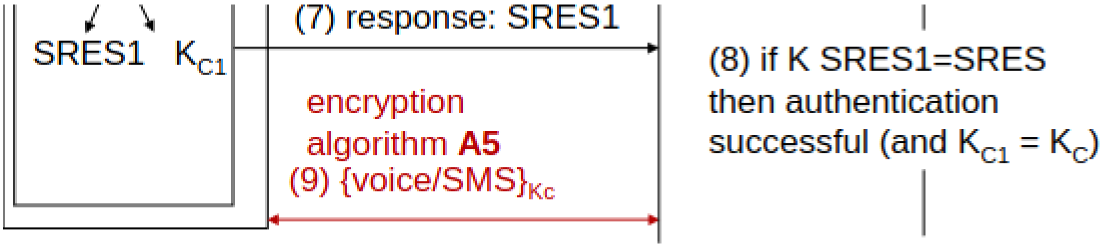
- MS also computes a symmetric key KC1 to be used for voice encryption if the authentication is successful

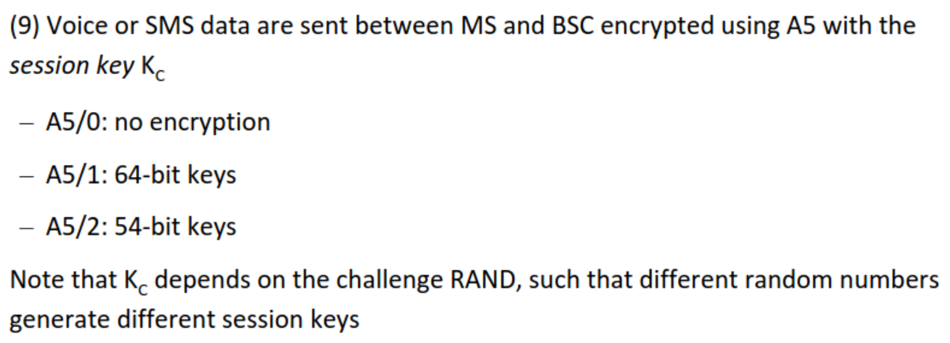
– (7) MS sends its response SRES1 to BSC

– (8) BSC verifies whether SRES1=SRES

- In case of successful authentication, SRES1 = SRES, and then (according to algorithm design) $K_{C1} = K_C$
- If SRES ≠ SRES1, then BSC terminates the authentication as unsuccessful

Step 9





(9) Voice or SMS data are sent between MS and BSC encrypted using A5 with the *session key* K_C

- A5/0: no encryption
- A5/1: 64-bit keys
- A5/2: 54-bit keys

Note that K_C depends on the challenge RAND, such that different random numbers generate different session keys

Does GSM Security Prevent Threats?

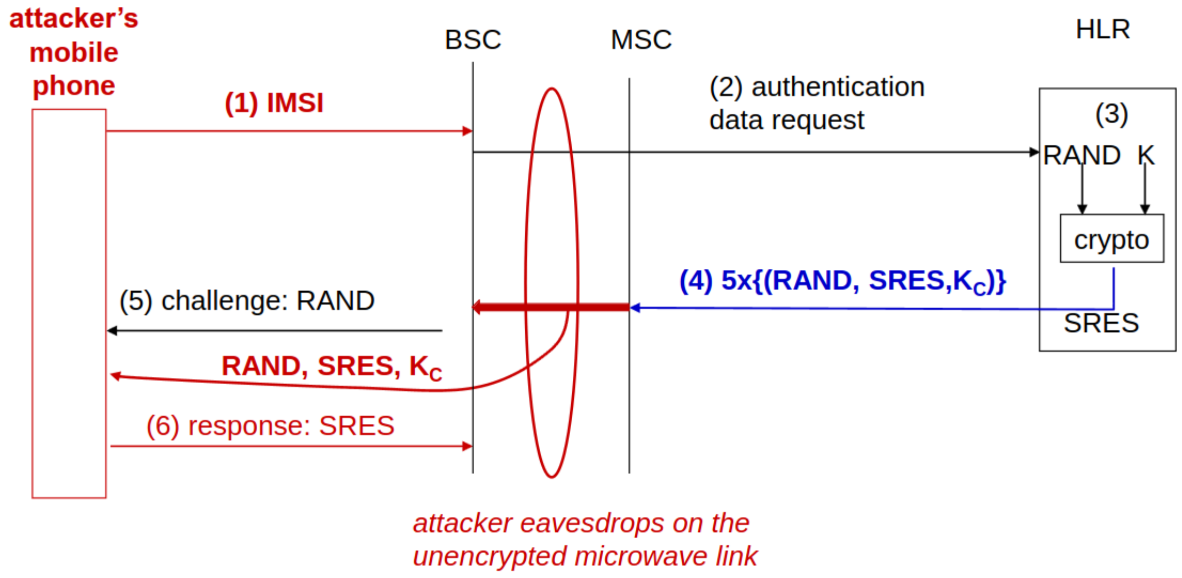
No, free calls, DoS, paying for alien/non-existing calls, eavesdropping and tracking are possible by criminals, providers, state actors.

Free call attack: Ross Anderson’s Hack

0. Eavesdrop an IMSI from elsewhere (Base Stations can ask for IMSI instead of TMSI)
1. Send IMSI to BSC
2. BSC/MSC send authentication data request to HLR
3. HLR generates SRES via a RAND and a key K
4. HLR sends 5 pairs of RAND, K_C and SRES to MSC/BSC

Between MSC and BSC, communication is unencrypted, SRES, RAND and K_C can be eavesdropped by attacker

5. BSC sends RAND as a challenge to mobile phone
6. Attackers sends the eavesdropped SRES



Free call attack: SIM Card Cloning

Option 1: Extract the key K_{SIM} from the smart card…but SIM Cards are smart cards, and smart cars are tamper proof => Secret hard to extract
Option 2: Cryptographic Attack

- Option 2.1: Get physical access to SIM card, break the crypto algorithms A3/A8
    - Submit many RAND queries, SIM card response with SRESs, analyze SRES response
    - If A3/A8 is cryptographically secure against chosen plaintext attacks, this attack should be infeasible
    - A3/A8 is NOT cryptographically secure
- Option 2.2: Over-the-air cloning (OTA): find out K_{SIM} from communication
    - Same as above, but with a more restricted number of RAND; SRES pairs
    - Eavesdrop on (RAND, SRES) paris, break the crypto algorithms A3/A8

A3/A8 Security by Obscurity

The authentication algorithms A3/A8 were kept secret, and they were based on COMP128

Design partially leaked in 1997, the rest was reverse engineered

Attacked using well-known standard techniques

=> A3/A8 was cracked and very insecure because it relied on obscurity, and GSM operators had to replace the algorithm

A3/A8/COMP128 Replacement

All SIM cards & back end had to be replaced over several months/years, as COMP128 was implemented in hardware.

Kerckhoffs' Six Principles

1. The system must be practically, if not mathematically, indecipherable;
2. **It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience; only the cryptographic key(s) should be kept secret**
3. Its key must be communicable and retainable without the help of written notes, and changeable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. t must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Principle 2: foundation of modern cryptography

Security by Obscurity

= keeping secret which algorithm is used for encryption. If the algorithm is known though, the systems can be compromised via cryptographic attacks, as the algorithm might be "bad"/broken.

=> SbO is bad, cryptographic algorithms should instead be subject to open analysis

AES (Advanced Encryption Standard)

AES algorithm was determined in the AES competition, it uses symmetric cipher with 128-bit key, (cipher = method to encrypt and decrypt data)

Attack: Stolen or Lost MS

Attack: Attackers steals MS (=steals mobile phone)

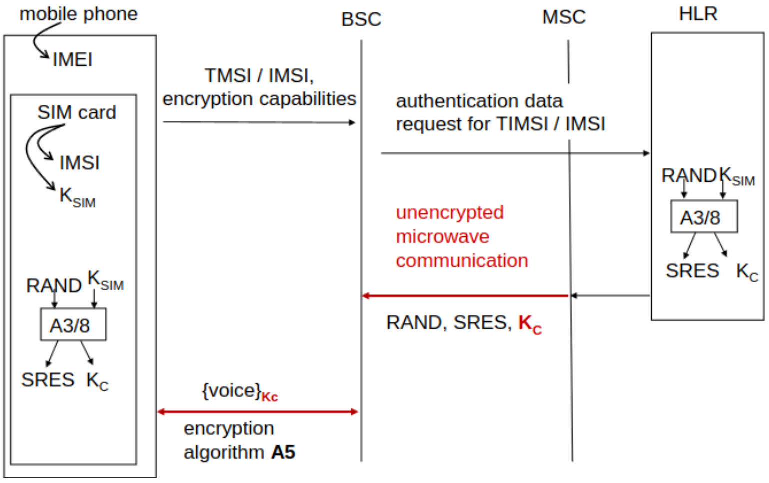
Countermeasures:

- Use of authentication for mobile phone and SIM card (PINs, patterns, biometrics)
- IMEI blacklisting (this can also be used to track criminals)
  - Requires to extract and write down the IMEI first

Attack: Backend Eavesdropping

Eavesdrop on the microwave link between base station and mobile switching center.
This link is was usually left unencrypted, Eavesdrop directly or find out key K_C

Eavesdropping on the A5 Key



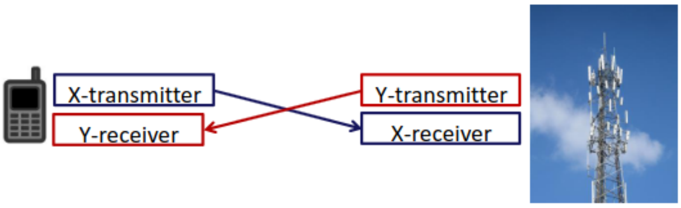
A5 Security by Obscurity

A5/1: Encryption algorithm A5/1 kept secret, secret was leaked, A5/1 could then be cracked with a powerful server. (key size: 64 bit)
A5/2: Intentionally lower security (key 54 bit). Can be decrypted in real time on a PC, and was prohibited/deprecated in 2007

How to Eavesdrop? => IMSI Catcher

Background

Mobile phone can't eavesdrop other mobile phone, as both transmit on one frequency range (X), but receive on other frequency range (Y)
X transmitter and Y receiver both built directly into mobile phone hardware => no X receiver => no eavesdropping

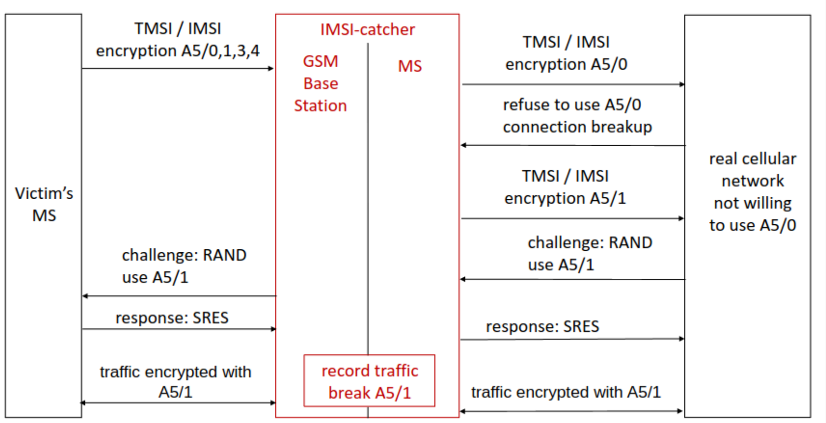


IMSI Catcher

IMSI catcher: = portable base station.

- Mobile Phone always connects to BS with strongest signal!
- Base station can require MS to send its real IMSI instead of TMSI
- Tracking using IMSI always possible!
- Sometimes it allows eavesdropping
- Works on GSM, UMTS and LTE
- Used by police and intelligence agencies (fake phone towers in Washington DC)
- GSM network tells MS which encryption algorithms to use => unciphered connection can be enforced (=A5/0), or one that can be cracked (A5/1)

Eavesdropping 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

GSM: Lessons Learned

- No “security by obscurity” in cryptography => Employ real cryptographers for system design and analysis, and use well-established ciphers
- Provide mutual authentication of MS and GMS network parts to avoid man in the middle attacks
- Provide security in every part of the system (CIA everywhere!)
- Crypto algorithm should be easy to change in a system (in case a cipher breaks one day)
- Consider future tech developments and new more powerful attackers in threat analysis
- Provide transparent technology development processes

Need to know

GSM (+4 of its features/properties)

GSM cellular network

GSM Architecture:

* NSS with OSS
  * HLR
  * EIR
  * AuC
  * MSC
  * VLR
  * GMSC
* RSS
  * MS
  * BTS
  * BSC

Handover Decision

Call setup

GSM S&P Assesement (4 parts)

GSM Threats, Victims (4 examples) and Attackers

4 GSM Security features

2 GSM Security Risks

9 steps of GSM athentification and voice/SMS encryption

* Grafik!
* IMEI
* IMSI
* TMSI
* BSC
* MSC

* HLR

* Encryption Algorithms

Free Call Attack: Ross Anderson's Hack (7 steps)

Free Call Attack: SIM Card Cloning (2 Options)

A3/A8 Algorithms (secure?)

Kerckhoff's Six Principles (name the mos timportant principle)

Security by Obscurity (+how to do it right)

AES

Attack: Stolen or Lost MS and Countermearues

Attack: Backend Eavesdropping

IMSI Catcher + Countermeasure

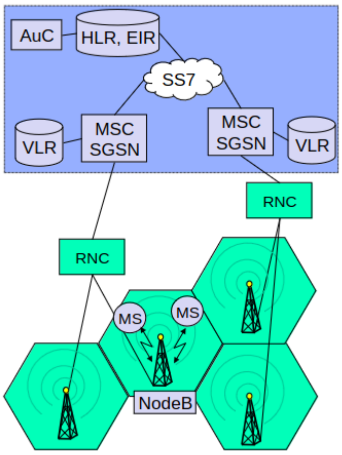
3. Cellular: UMTS Architecture and Security

UMTS (Universal Mobile Telecommunications System)

More secure successor of GSM:

- Mutual authentication between MS and network
- Confidentiality and integrity of signaling data
- Confidentiality of user data
- Better algorithms: A5/3 and A5/4
  - Academically broken, but practical attacks in UMTS networks is doubtful
- Backwards compatible to GSM
  - Allows man-in-the-middle attack

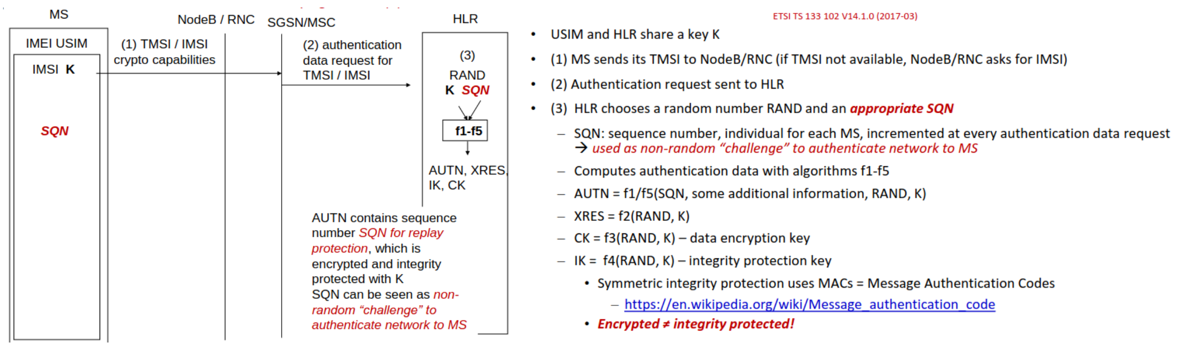
UMTS Architecture



- MS: Mobile Station
- USIM: UMTS SIM
- *NodeB*: Base Transceiver Station
  - = BTS from GSM
- *RNC*: Radio Network Controller
  - = BSC from GSM
- MSC: Mobile Switching Center
  - For voice and SMS
- *SGSN*: Serving GPRS Support Node
  - *For data transfer*
- VLR: Visitor Location Register
- HLR: Home Location Register
- EIR: Equipment Identity Register
- AUC: Authentication Center
- SS7: internal network protocols

UMTS Authentication and Key Agreement (14 steps)

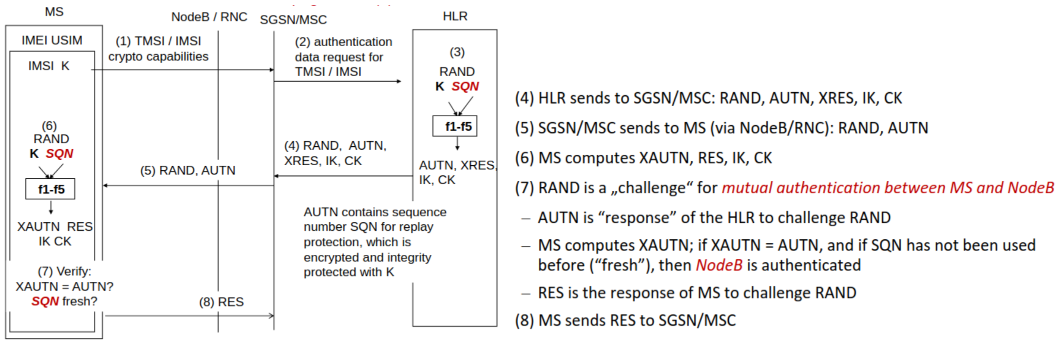
Steps 1-3


MS
NodeB / RNC
SGSN/MSC
HLR
IMEI USIM
IMSI K
SQN
(1) TMSI / IMSI
crypto capabilities
(2) authentication
data request for
TMSI / IMSI
(3)
RAND
K SQN
f1-f5
AUTN, XRES,
IK, CK
AUTN contains sequence number SQN for replay protection, which is encrypted and integrity protected with K
SQN can be seen as non-random "challenge" to authenticate network to MS

ETSI TS 133 102 V14.1.0 (2017-03)

- USIM and HLR share a key K
- (1) MS sends its TMSI to NodeB/RNC (if TMSI not available, NodeB/RNC asks for IMSI)
- (2) Authentication request sent to HLR
- (3) HLR chooses a random number RAND and an ***appropriate SQN***
  - SQN: sequence number, individual for each MS, incremented at every authentication data request
    → *used as non-random "challenge" to authenticate network to MS*
  - Computes authentication data with algorithms f1-f5
  - AUTN = f1/f5(SQN, some additional information, RAND, K)
  - XRES = f2(RAND, K)
  - CK = f3(RAND, K) – data encryption key
  - IK = f4(RAND, K) – integrity protection key
    - Symmetric integrity protection uses MACs = Message Authentication Codes
      - https://en.wikipedia.org/wiki/Message_authentication_code
    - ***Encrypted ≠ integrity protected!***

Steps 4-8



(4) HLR sends to SGSN/MSC: RAND, AUTN, XRES, IK, CK

(5) SGSN/MSC sends to MS (via NodeB/RNC): RAND, AUTN

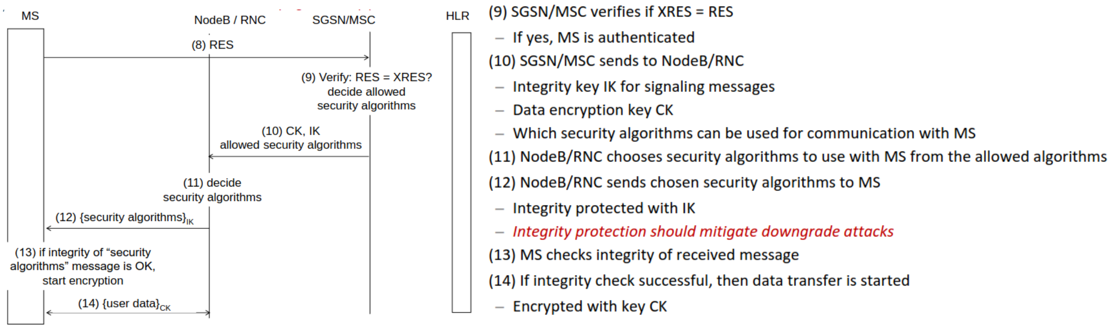
(6) MS computes XAUTN, RES, IK, CK

(7) RAND is a „challenge" for *mutual authentication between MS and NodeB*

- AUTN is "response" of the HLR to challenge RAND
- MS computes XAUTN; if XAUTN = AUTN, and if SQN has not been used before ("fresh"), then *NodeB* is authenticated
- RES is the response of MS to challenge RAND

(8) MS sends RES to SGSN/MSC

Steps 9-14



(9) SGSN/MSC verifies if XRES = RES

- If yes, MS is authenticated

(10) SGSN/MSC sends to NodeB/RNC

- Integrity key IK for signaling messages
- Data encryption key CK
- Which security algorithms can be used for communication with MS

(11) NodeB/RNC chooses security algorithms to use with MS from the allowed algorithms

(12) NodeB/RNC sends chosen security algorithms to MS

- Integrity protected with IK
- *Integrity protection should mitigate downgrade attacks*

(13) MS checks integrity of received message

(14) If integrity check successful, then data transfer is started

- Encrypted with key CK

Attack: Man-in-the-middle GSM-UMTS Degredation Attack

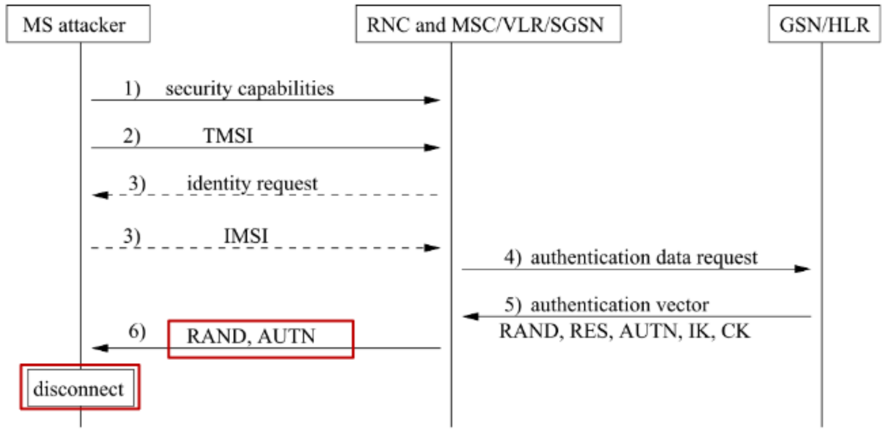
• Assumption: MS implements both GSM and UMTS

• IMSI-catcher impersonates MS in UMTS mode to RNC

• RNC sends to MS: "fresh" authentication token AUTN

– IMSI-catcher breaks up the connection, saves AUTN

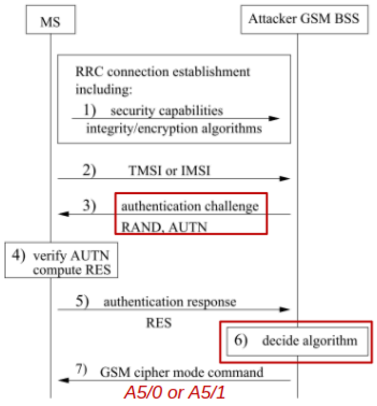
Step 1: use TIMSI / IMSI to get valid AUTN



Now we have a valid AUTN…

Step 2:

- Authenticate to target MS using AUTN
- *What next?*


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

IMSI-catcher *immediately* initiates a GSM connection to the MS

- This connection also has to be mutually authenticated
  - Use AUTN (because it is *"fresh"*)

IMSI-catcher tells the MS to use A5/0 or A5/1

IMSI-catcher sets up a "normal" call to the UMTS network

Result: A5/1 or A5/0 traffic between MS and IMSI-catcher, normal traffic between IMSI-catcher and network

Drawbacks

- IMSI-catcher has to pay for the call
- Victim calls *from a different phone number*
  - Would communication partner notice this?

Why is this possible?

Integrity protection of the "security algorithms" command is not possible

– GSM does not support integrity protection of signaling messages

Lessons Learned

(see previous chapter) +
Implementing backward compatibility can leave old vulnerabilities exploitable

Need to know

UMTS
UMTS Architecture

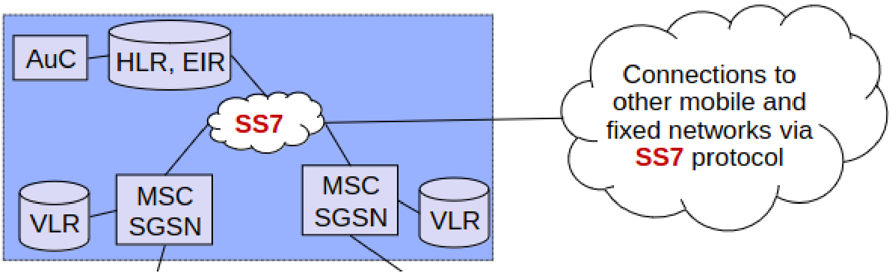
- MS
- USIM
- NodeB
- RNC
- MSC
- SGSN
- VLR
- HLR
- EIUR
- AUC
- SS7

UMTS Authentication and Key Agreement (14 steps)
Man-in-the-middle GSM-UMTS Degradation Attack (7 steps)
Why does GSM-UMTS Degradation Attack work?
Problem of backward compatibility

4. Cellular: SS7, LTE, Location privacy

SS7 Protocol

- Developed in 1980 for communication between telecom operators
  - Trust Assumption: everybody is trusted, nos security needed
  - No authentication, no plausibility states
- Current state: used for GSM/UMTS, everybody can buy access for SS7


AuC
HLR, EIR
SS7
VLR
MSC
SGSN
MSC
SGSN
VLR
Connections to other mobile and fixed networks via SS7 protocol

Attack: Rogue SS7 Operator

- Locate & track:
  - Ask HLR for IMSI of phone number
  - Ask HLR which MSC is this IMSI
  - Ask MSC: At which BTS is this IMSI
- Eavesdrop: “Please send authentication and encryption keys for this TMSI”
- Manipulate: “This IMSI wants its calls/SMS forwarded to my network”

Attack: Stealing Money via SS7

1. Attacker gains control over victim’s online banking account (via phising or malware)
2. Attacker looks up victim’s phone number (Online banking with OTP(one-time-password) via SMS)
3. Sets up SMS redirect
4. Logs into online banking
5. Start translation, use SMS OTP to verify transaction => Money!

Location Privacy

= the ability to prevent other parties from learning one’s current or past location.

Location Triangle: Who, Where, When

- Location + Time known: Where have you been at which time
  => Identification of person possible (Nights => At home, Workdays => in the office, …)

- Who + At Which Time => Predict where a person is or might be
- Who + where: predict which time (Max is at the office => it's a workday between 8 and 16)
- Mobile Profiling and Surveillance possible, and crime investigation (Capitol Attack 2021)

Data Retention EU (Vorratsdatenspeicherung)

2006: All traffic data (not content!) must be stored for 6 months, and accessible for law enforcement, no additional data may be generated
2007: Transfer of the Data Retention law to national law for telecommunication
2007: Constitutional complaint was filed at the Bundesverfassungsgericht (basic privacy violation, cost in no relation to use)
2009: Transfer for internet
2010: Bundesverfassungsgericht decides that data retention is against German Constitution
2012: EU takes legal action against Germany
2014: EuGH rejects EU-wide data retention law as its being against the EU Charta of human rights
2015: New Data retention law in Germany, next round of complaints
2016: EuGH rejects data retention
2017: data retention stopped in Germany
2022: EuGH conforms rejection

6 Data Things Stored by Operators (telephony, mobile telephony, internet)

1. Phone number of caller and callee
2. Start and end of connection
3. IMSI & IMEI of caller and callee
4. Cell id of caller and callee at the beginning of the connection
5. IP Address of caller and callee (Internet Telephony)
6. Begin and end of access to Internet and IP address of the user

GSM / UMTS Tracking Possibilities

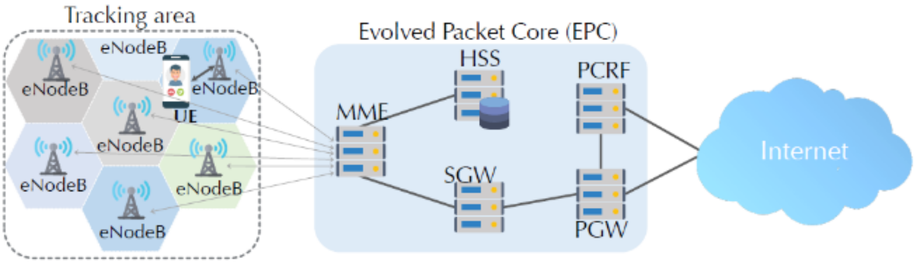
IMSI Catcher: Pinpoint targeted person with precision up to several meters
TMSI: TMSI can be switched off by BS, TMSI is rarely changed
IMEI: BS can ask mobile phone to transmit their IMEI (feature against phone theft)

LTE/4G

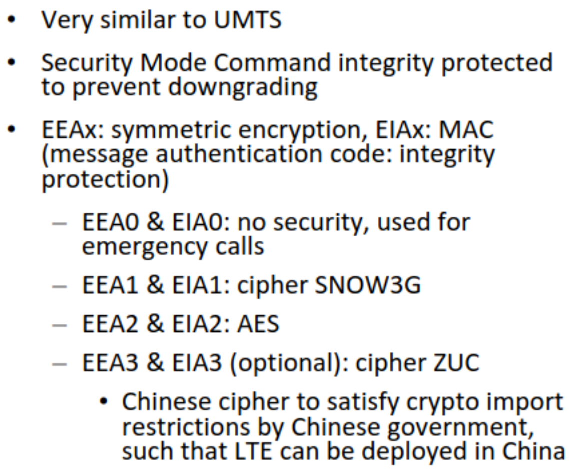
- LTE = 4G
- MS → UE: User Equipment
- TIMSI → GUTI: Globally Unique Temporary Identifier
- NodeB → eNodeB (e for evolved)
- MSC → MME: Mobility Management Entity
- …

LTE Architecture

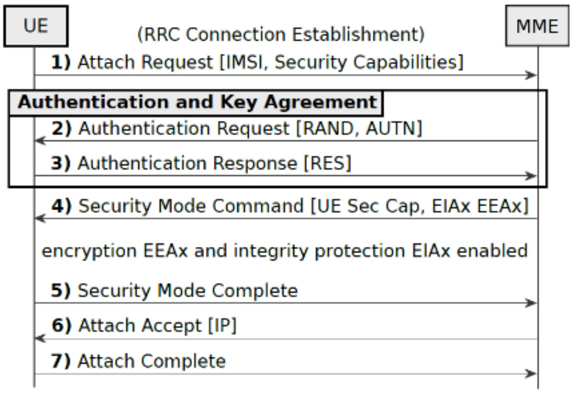
- Mutual authentication, integrity, confidentiality, location privacy
- MME selects crypto algorithm, runs authentication with UE, keeps track of locations
- AES can be used for encryption and integrity protection (other algorithm are also possible)
- Communication between eNodeB and EPC, and inside EPC protected with IPSec



LTE Connection Setup



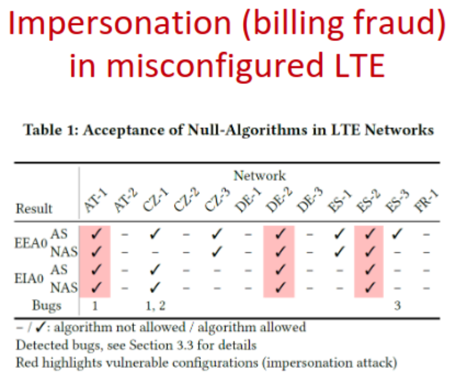
- Very similar to UMTS
- Security Mode Command integrity protected to prevent downgrading
- EEAx: symmetric encryption, EIAx: MAC (message authentication code: integrity protection)
  - EEA0 & EIA0: no security, used for emergency calls
  - EEA1 & EIA1: cipher SNOW3G
  - EEA2 & EIA2: AES
  - EEA3 & EIA3 (optional): cipher ZUC
    - Chinese cipher to satisfy crypto import restrictions by Chinese government, such that LTE can be deployed in China



UE
MME
(RRC Connection Establishment)
1) Attach Request [IMSI, Security Capabilities]
Authentication and Key Agreement
2) Authentication Request [RAND, AUTN]
3) Authentication Response [RES]
4) Security Mode Command [UE Sec Cap, EIAx EEAx]
encryption EEAx and integrity protection EIAx enabled
5) Security Mode Complete
6) Attach Accept [IP]
7) Attach Complete

Figure 2: LTE Attach Procedure including the AKA.

Attack: Impersonation (blling fraud) in misconfigured LTE

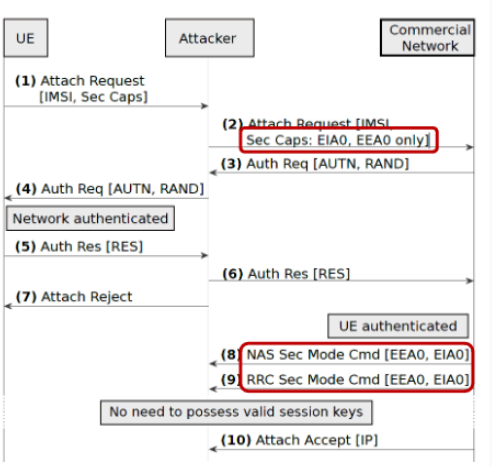


Impersonation (billing fraud) in misconfigured LTE

Table 1: Acceptance of Null-Algorithms in LTE Networks

Result		Network AT-1	AT-2	CZ-1	CZ-2	CZ-3	DE-1	DE-2	DE-3	ES-1	ES-2	ES-3	FR-1
EEA0	AS	✓	–	✓	–	✓	–	✓	–	✓	✓	✓	–
	NAS	✓	–	–	–	✓	–	✓	–	✓	✓	–	–
EIA0	AS	✓	–	✓	–	–	–	✓	–	–	✓	–	–
	NAS	✓	–	✓	–	–	–	✓	–	–	✓	–	–
Bugs		1		1, 2								3	

– / ✓: algorithm not allowed / algorithm allowed
Detected bugs, see Section 3.3 for details
Red highlights vulnerable configurations (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.

LTE Privacy: 3 Tradeoffs

- GUTI management similar to TMSI management in UMTS/GSM
  - Globally Unique Temporary Identity
  - Rarely changed: remains the same for several days, even when moving
  - Trade-off performance ↔ privacy
- Location leaks via paging requests to eNodeBs
  - Is UE with this GUTI somewhere in the vicinity?")
  - Unexpected effects of new functionality: smartphone apps
  - Facebook messages (even from non-friends)
  - WhatsApp: just typing (but not sending) a message
- Location leaks by impersonating eNodeBs
  - UE can be asked to send signal strengths of all eNodeBs it can hear without authentication
  - Reason: troubleshooting, but can be used for trilateration
  - Trade-off availability/reliability ↔ privacy
  - Messages can also include GPS coordinates of UEs

LTE Availability: 1 Tradeoff

- eNodeB impersonation
  - eNodeB messages of the type "service X not allowed" do not require authentication
  - "LTE services not allowed" can be used for
    - Degradation attacks to 2G/3G
    - Permanent DoS (till UE reboot)
- Trade-off availability ↔ security

Cryptography Basics

Differences between symmetric and asymmetric?

- Symmetric cryptography:
  - Uses a single key for both encryption and decryption.
  - Faster compared to asymmetric cryptography.
  - Typically used for encrypting large amounts of data.
  - Requires secure key exchange between parties.
  - Examples include AES (Advanced Encryption Standard) and DES (Data Encryption Standard).
  - Symmetric: MACs = Message Authentication Codes
- Asymmetric cryptography:
  - Uses a pair of keys: public and private.
  - Public key is used for encryption, private key for decryption.
  - Slower than symmetric cryptography due to complex mathematical operations.
  - Eliminates the need for secure key exchange.

  - Enables digital signatures and key distribution.
  - Examples include RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography).
  - Asymmetric: digital signature

Differences between stream ciphers and block ciphers? (=> Confidentiality Protection)

- Stream ciphers encrypt one bit or byte of plaintext at a time, often in real-time, generating a continuous stream of ciphertext.
- Block ciphers encrypt fixed-size blocks of plaintext (e.g., 64 or 128 bits) at a time, padding may be required for uneven blocks.
- Stream ciphers are often faster and more suitable for real-time communication, while block ciphers are more versatile and commonly used for data encryption where data can be processed in blocks.

Usage of asymmetric encryption (=> Confidentiality Protection)

- Secure key exchange: Used to establish shared secret keys for symmetric encryption.
- Digital signatures: Ensures authenticity and integrity of messages or data.
- Key management: Facilitates secure distribution and management of encryption keys.
- Secure communication channels: Helps establish secure channels for communication over untrusted networks

Usage of Hybrid Protocols (=> Confidentiality Protection)

- Hybrid protocols combine the strengths of both symmetric and asymmetric encryption to achieve efficiency and security.
- Typically, a hybrid protocol involves using asymmetric encryption for key exchange and symmetric encryption for actual data transmission.
- For example, in SSL/TLS protocols, asymmetric encryption (such as RSA) is used for initial key exchange and authentication, while symmetric encryption (such as AES) is used for bulk data encryption, ensuring both security and performance.

Difference between MACs and digital signatures? (Integrity protection of messages)

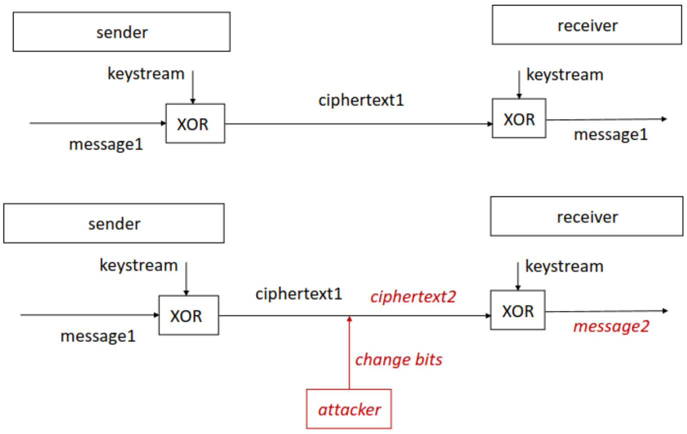
- MACs (Message Authentication Codes):
  - Uses a secret key and a message to generate a fixed-size authentication tag.
  - Provides integrity protection and authentication.
  - Typically symmetric-key based.
- Digital Signatures:
  - Generated using the sender's private key and the message.
  - Verifiable using the sender's public key.
  - Provides integrity protection, authentication, and non-repudiation.
  - Typically asymmetric-key based.

LTE Integrity

Messages/Signaling Information:	**Voice Calls:**
confidentiality + integrity protection	only confidentiality protection
block cipher	stream cipher
AES in CBC-MAC mode	AES-CTR
Messages processed in blocks, each block undergoes several separate transformations	AES-CTR key is used to generate keystream Encryption and decryption happens bit-by-bit: keystream XOR message
	performance ↔ security, block cipher is more secure but not well suited for real-time voice calls
	aLTEr Attack possible due to missing integrity protection

Attack: aLTEr Stream Cipher attack

- Attacks Integrity of data
- Change encrypted traffic without knowing plain text or keystream
- Attack is possible due to XOR and missing integrity check
- Can be used for DNS redirection to malicious websites
- Receiver and sender use same keystream to encrypt/decrypt stream
- Attacker can flip/change bits
- Can often only be used to create a gibberish message, but it has been proven that it is possible to manipulate the message properly



Need to know:

SS7 protocol
Attack: Rogue SS7 Operator (3 parts attacks, 3 steps)
Attack: Stealing Money via SS7 (5 steps)
Location Privacy definition
Location Triangle (3 parts + predictions)
GSM / UMTS Tracking Possibilities (3 parts)
LTE Architecture (Features, encryption + protection)
LTE Connection Setup (7 steps + encryption capabilities)
LTE Impersonation/Billing Fraud (10 steps)
LTE Privacy Tradeoffs (3 parts)
LTE Availability Security Tradeoff

Difference between Symmetric and Asymmetric Cryptography + Example Algorithms/standard

Difference Between Block and Stream Cipher

Usage of asymmetric Encryption + Which part of CIA?

Difference between MAC and digital signature

LTE Integrity: Messages vs Vouce Calls

Attack aLTEr Attack

5. Cellular: Attacks on LTE and 5G

Attack: Eavesdropping Encrypted LTE Calls (ReVoLTE)

LTE voice and media calls use VoLTE service

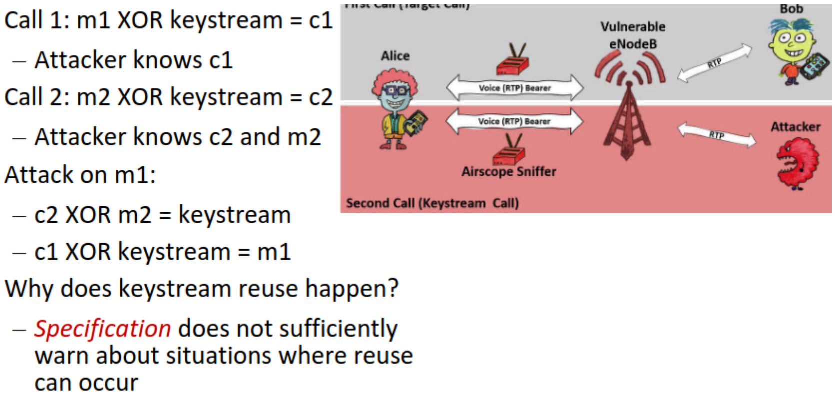
- bit by bit stream cipher encryption via keystream XOR message
- **All stream ciphers are vulnerable to keystream reuse attack!**

ReVoLTE Attack:

(1) Target Call: Alice calls Bob, attacker Eve records the call

(2) Keystream Call: Eve calls Alice immediately after the first call

- Theoretically, new keys should be negotiated
- In practice, base stations often use the same keystream
- Flawed implementation due to unclear specification



Call 1: m1 XOR keystream = c1

– Attacker knows c1

Call 2: m2 XOR keystream = c2

– Attacker knows c2 and m2

Attack on m1:

– c2 XOR m2 = keystream

– c1 XOR keystream = m1

Why does keystream reuse happen?

– *Specification* does not sufficiently warn about situations where reuse can occur

5G Security Issues (in a nutshell)

- Equipment Manufacturers (Huawei) could have modified their hardware to spy on other countries (Smartphones, 5G)
- Huge Attack Surfaces due to interconnected IoT devices and high data rates (cars, smartphones, …)
- 5G authentication doesn't have explicit threat model and security goals in documentation
  - IMSI Catcher doesn't work anymore, as SUPI (=5G IMSI) isn't sent in clear text anymore
  - But: Other tracking and impersonating attacks still possible

5G Authentification

Serving Network is a Base Station

SQN = Sequence Number (SQN from Subscriber and Homenetwork should be the same)

UMTS hat auch SQN!

pubK and privK are used to encrypt/decrypt SUPI, by computing SUCI which is the encrypted SUPI

This Authentication makes IMSI catchers useless due to the encryption used

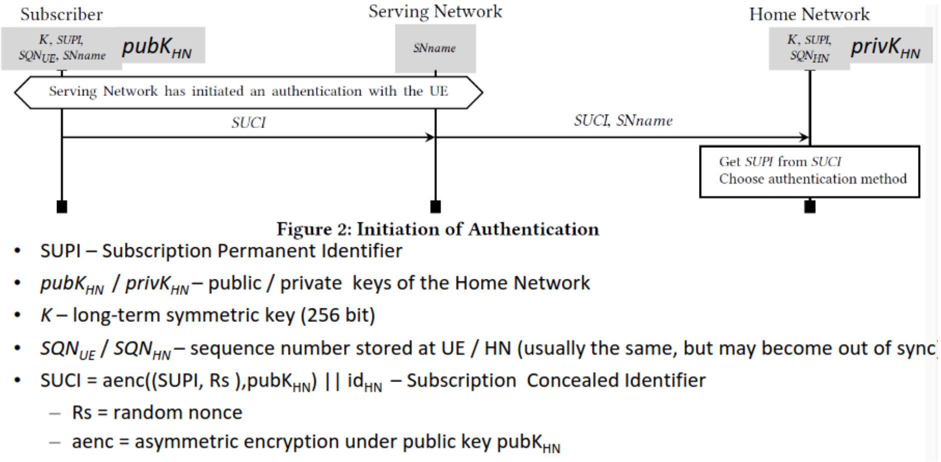


Figure 2: Initiation of Authentication

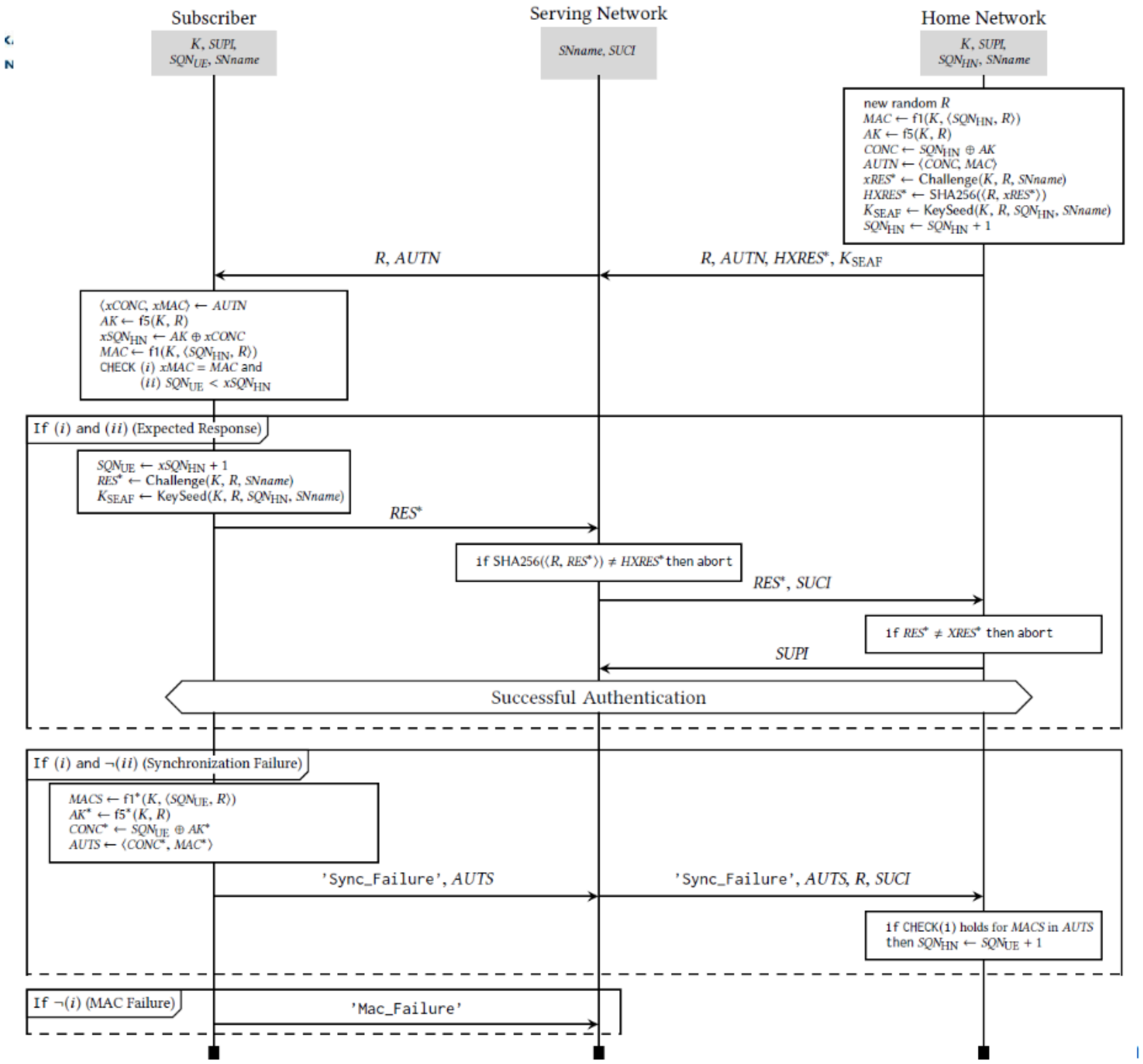
- SUPI – Subscription Permanent Identifier
- $pubK_{HN}$ / $privK_{HN}$ – public / private keys of the Home Network
- K – long-term symmetric key (256 bit)
- SQN_{UE} / SQN_{HN} – sequence number stored at UE / HN (usually the same, but may become out of sync
- SUCI = aenc((SUPI, Rs),$pubK_{HN}$) | | id_{HN} – Subscription Concealed Identifier
  - Rs = random nonce
  - aenc = asymmetric encryption under public key $pubK_{HN}$

5G AKA Protocol

AKA: Authentication and Key Agreement protocol
Das hier ist wie oben, bloß mit resynchronisierung (SQN von Subrsiber und Home Network können out of sync sein)
Grafik müssen wir nicht können, aber wir müssen wissen dass die procedure eigentlich genau so wie ist bei UMTS. Es gibt:

- AUTN (Authentifizierung des Home Networks gegenüber des Subscribers)
- AUTN und Challenge R wird von Home Network über Serving Network zu Subsriber durchreroutet und serving Network bekommt einen key K_{SEAF}
- Subrsibe rüberprüft on AUTN okay ist
- Subrisber sendet RES (Response) to Serving Network, Sertving Network leitete weiter an Home Network
- Home Network überprüft ob Response passt, wenn ja sendet Home Network SUPI and Serving Network
- Authentifizierung erfolgreich

- Failure 1 - Sync Failure: Authentifikation erfolgreich, außer in der kommunikation ist die SQN out-of sync (? Nachrichten gingen verloren), dann wird Sync failure gesendet, udn ein resync procress gestartet
- Failure 2 - MAC Failure: Authentifikation erfolgreich, außer MAC Failure passiert (wird in RFID nochmal genauer gesprochend)
  - Failures erlauben Attacken, siehe RFID


Subscriber
K, $SUPI$, SQN_{UE}, $SNname$
Serving Network
$SNname$, $SUCI$
Home Network
K, $SUPI$, SQN_{HN}, $SNname$
new random R
$MAC \leftarrow \mathrm{f1}(K, \langle SQN_{HN}, R \rangle)$
$AK \leftarrow \mathrm{f5}(K, R)$
$CONC \leftarrow SQN_{HN} \oplus AK$
$AUTN \leftarrow \langle CONC, MAC \rangle$
$xRES^* \leftarrow \mathrm{Challenge}(K, R, SNname)$
$HXRES^* \leftarrow \mathrm{SHA256}(\langle R, xRES^* \rangle)$
$K_{SEAF} \leftarrow \mathrm{KeySeed}(K, R, SQN_{HN}, SNname)$
$SQN_{HN} \leftarrow SQN_{HN} + 1$
R, $AUTN$
R, $AUTN$, $HXRES^*$, K_{SEAF}
$\langle xCONC, xMAC \rangle \leftarrow AUTN$
$AK \leftarrow \mathrm{f5}(K, R)$
$xSQN_{HN} \leftarrow AK \oplus xCONC$
$MAC \leftarrow \mathrm{f1}(K, \langle SQN_{HN}, R \rangle)$
CHECK (i) $xMAC = MAC$ and
(ii) $SQN_{UE} < xSQN_{HN}$
If (i) and (ii) (Expected Response)
$SQN_{UE} \leftarrow xSQN_{HN} + 1$
$RES^* \leftarrow \mathrm{Challenge}(K, R, SNname)$
$K_{SEAF} \leftarrow \mathrm{KeySeed}(K, R, SQN_{HN}, SNname)$
RES^*
if $\mathrm{SHA256}(\langle R, RES^* \rangle) \neq HXRES^*$ then abort
RES^*, $SUCI$
if $RES^* \neq XRES^*$ then abort
$SUPI$
Successful Authentication
If (i) and $\neg(ii)$ (Synchronization Failure)
$MACS \leftarrow \mathrm{f1}^*(K, \langle SQN_{UE}, R \rangle)$
$AK^* \leftarrow \mathrm{f5}^*(K, R)$
$CONC^* \leftarrow SQN_{UE} \oplus AK^*$
$AUTS \leftarrow \langle CONC^*, MAC^* \rangle$
'Sync_Failure', $AUTS$
'Sync_Failure', $AUTS$, R, $SUCI$
if CHECK(1) holds for $MACS$ in $AUTS$ then $SQN_{HN} \leftarrow SQN_{UE} + 1$
If $\neg(i)$ (MAC Failure)
'Mac_Failure'

Cellular Security & Privacy: 12 Lessons Learned

- No “security by obscurity” in cryptography
- Provide mutual authentication
- Provide security (confidentiality + integrity) in every part of the system
- Crypto algorithms should be easy to change
- Consider future technology developments and adequately powerful attackers in threat analysis
- Provide transparent technology development processes
- Implementing backward compatibility can leave old vulnerabilities exploitable
- Management of pseudonyms should be specified & implemented very carefully
- Management of non-secure modes should be specified & implemented very carefully
- Specifications should warn clearly about possibilities of insecure implementations
- Availability, reliability, performance measures as well as new applications can have unforeseen security & privacy consequences
- Specifications should precisely define security goals and threat model

Need to know:

Attack: Eavesdropping Encrypted LTE Calls (ReVoLTE)
5G Security Issues (in a nutshell)
5G Authentification
5G AKA Protocol
Cellular Security & Privacy: 12 Lessons Learned

5. Wi-Fi: Wi-Fi, OAN, WEP

Wi-Fi (Wireless Fidelity, IEEE 802.11 since 1997)

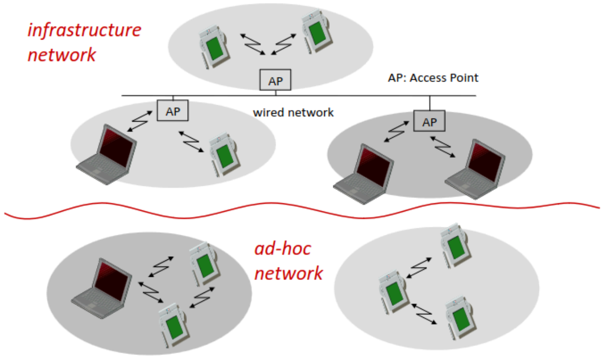
8 Goals:

- Global + Seamless Operation
- Low Power (for battery use)
- No special license or permissions needed
- Robust transmission
- Simplified spontaneous cooperation of users
- Easy to use
- Safety (low radiation)
- Security

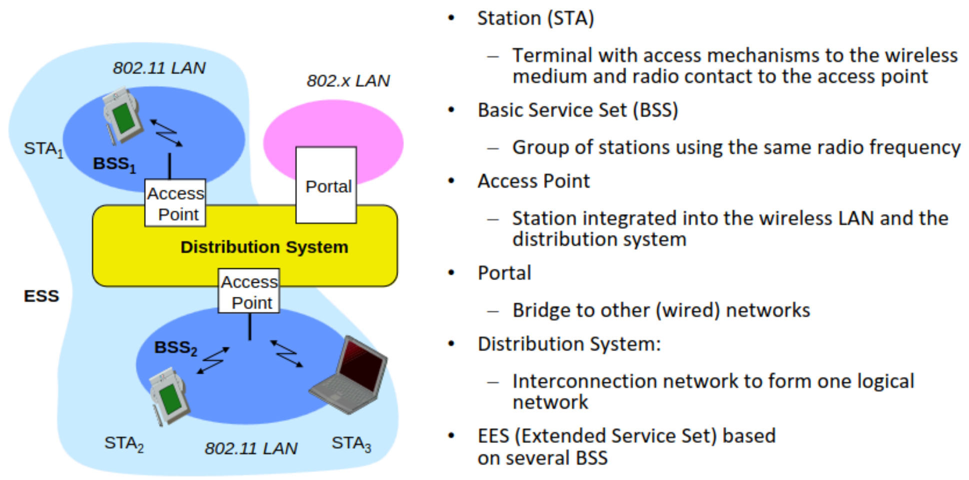
Different Versions over the years:

- WEP
- WPA
- WPA2
- WPS
- WPA3

Network Types: Infrastructure and Ad-hoc



Wi-Fi Network Infrastructure



- Station (STA)
  - Terminal with access mechanisms to the wireless medium and radio contact to the access point
- Basic Service Set (BSS)
  - Group of stations using the same radio frequency
- Access Point
  - Station integrated into the wireless LAN and the distribution system
- Portal
  - Bridge to other (wired) networks
- Distribution System:
  - Interconnection network to form one logical network
- EES (Extended Service Set) based on several BSS

Connecting to Wi-Fi Network

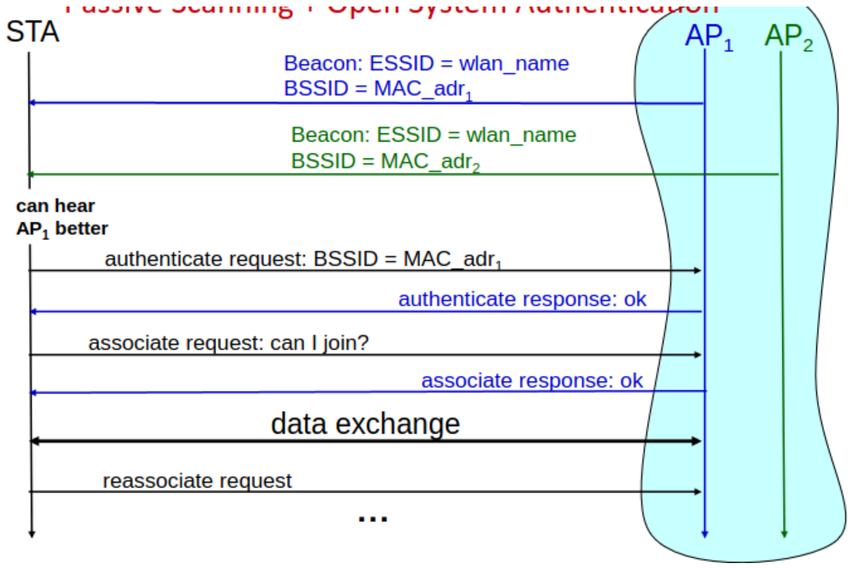
Scanning

- Active: STA sends probes into the medium and waits for an answer
- Passive: STA listens into the medium for beacon signals
- Beacons: signals sent by AP for announcing its presence

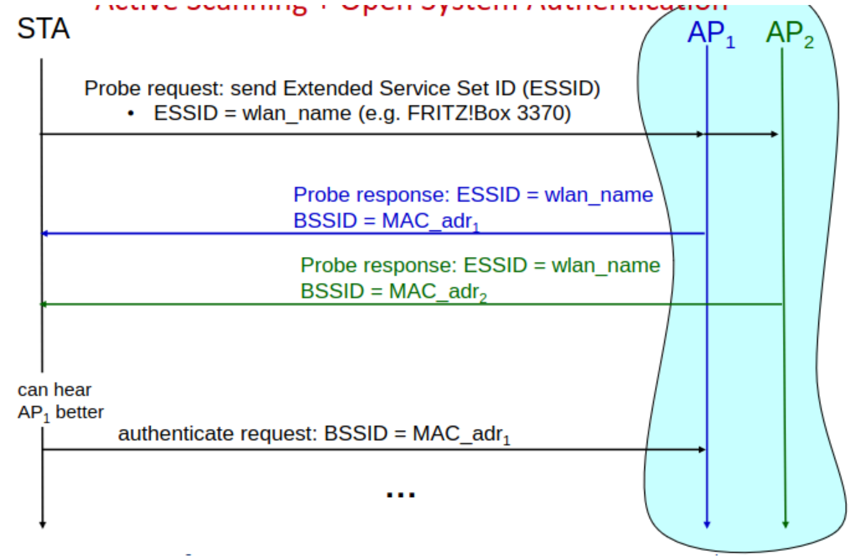
Authentication

- Open system (no cryptographic protocols)
  - STA sends “authenticate request”
  - AP sends “authenticate response”
- Shared key (using WEP/WPA/WPA2/WPA3 cryptographic algorithm)
  - Challenge-response protocol
  - WEP: broken
  - WPA: partially broken
  - WPA2: more or less okay
    - Password guessing attacks, insider attacks: possible “by design”
  - WPA3: public key crypto

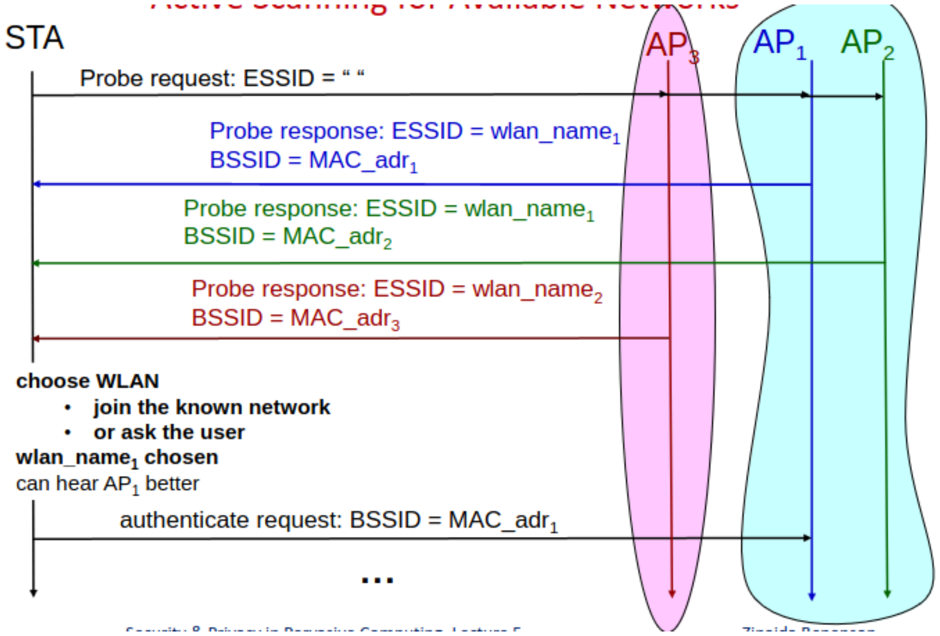
Passive Scanning + Open System Authentication


STA
AP$_1$
AP$_2$
Beacon: ESSID = wlan_name
BSSID = MAC_adr$_1$
Beacon: ESSID = wlan_name
BSSID = MAC_adr$_2$
can hear
AP$_1$ better
authenticate request: BSSID = MAC_adr$_1$
authenticate response: ok
associate request: can I join?
associate response: ok
data exchange
reassociate request
...

Active Scanning + Open System Authentication


STA
AP$_1$
AP$_2$
Probe request: send Extended Service Set ID (ESSID)
• ESSID = wlan_name (e.g. FRITZ!Box 3370)
Probe response: ESSID = wlan_name
BSSID = MAC_adr$_1$
Probe response: ESSID = wlan_name
BSSID = MAC_adr$_2$
can hear
AP$_1$ better
authenticate request: BSSID = MAC_adr$_1$
...

Active Scanning for Available Networks



Probe request: ESSID = " "

Probe response: ESSID = wlan_name$_1$
BSSID = MAC_adr$_1$

Probe response: ESSID = wlan_name$_1$
BSSID = MAC_adr$_2$

Probe response: ESSID = wlan_name$_2$
BSSID = MAC_adr$_3$

choose WLAN

- **join the known network**
- **or ask the user**

wlan_name$_1$ chosen

can hear AP$_1$ better

authenticate request: BSSID = MAC_adr$_1$

...

Attacks: Open Network Protection -How to prevent alien STAs from joining the WLAN - and how an attacker can join anyway

Solution 1: Hidden ESSID

How it works:

- APs do not send beacons, wait for STAs to ask for a particular ESSID
- = Security by Obscurity! => Bad

How to attack it: Joining Attack

- Sniff till some STA sends a probe request to the hidden WLAN
- Replay Attack: Send probe request for the same WLAN name

Privacy: STAs actively looking for hidden WLANs at every location

Solution 2: MAC Address Filtering

How it works:

- APs only answer to the probes or authentication requests of STAs with known MAC addresses
- Security by Obscurity => Bad

How to attack:

- Sniff allowed MAC addresses
- Schange (spoof) your MAC addressed to sniffed allowed one

Attack: Rogue Access Point (Evil Twin Attack)

Setup the Attack:

- Attacker sets up AP, this AP impersonates the legitimate AP
- Broadcasts beacon signal with the ESSID of the legitimate AP
- => Evil Twin

Attack:

- Device connect to the AP with the strongest signal
- Evil Twin beacon can be made the strongest (by going near the victim's device)

What can be done with an Evil Twin:

- Send fake login pages to user device, steal passwords etc.
- Forward Internet traffic (=sniff all clear text information, including login info)
- Phishing: Use DNS spoofing to redirec to evil servers (redirect traffic from My Bank to evil fake version of the Bank)

Defense:

- VPN
- SSL/TLS
- but: no general defense

WEP (Wired Equivalent Privacy, 1999)

Goal

Province same security as a wired connection

How it works

One permanent shared master key for all network members, via RC4 stream cipher

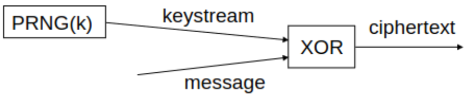
- RC4 used to implement PRNG for generating the keystream
  - PRNG = Pseudo Random Number Generator
  - RC4 is considered insecure nowadays
- IVs (initialization vectors)
  - Produces different keystream for each wireless frame
  - Sent in the clear together with the frame

Secure?

No, real time traffic decryption and key recovers possible, RC4 is insecure

Cryptography Basics: Stream Ciphers

- Pseudo random number generator PRNG()
  - Generates bit sequences: 0010111010101…
- Cryptographically secure
- Secret key k = seed
- PRNG(k) = pseudo-random bit stream
- Bitwise XOR with the message
  - Similar to one time pad, but no perfect secrecy



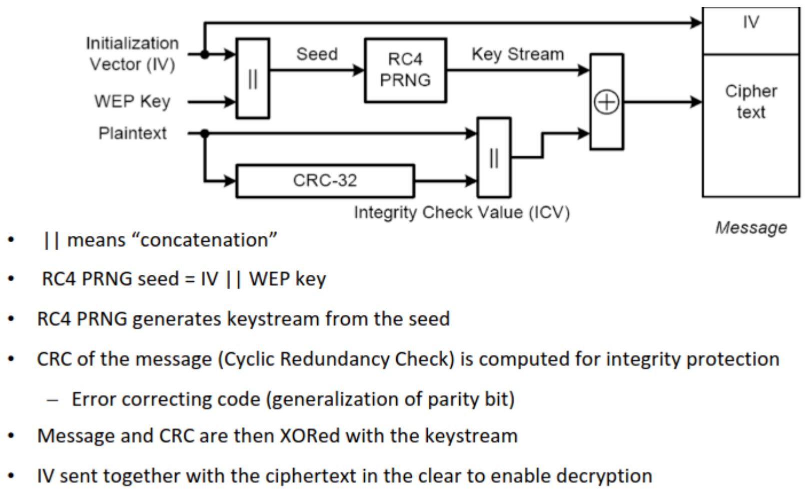
Cryptography Basics: Pseudo Random Number Generator (PRNG)

- PRNG algorithm should be public (Kerckhoff's Principle 2)
- Seed is secret (=key)
- Cryptographically secure PRNG:
    - 1) Infeasible to predict next output bit of the output with probability
    - significantly better than ½ without knowing the seed
    - Even if all previous bits of output are known
    - 2) Infeasible to predict previous bits of output if the current state becomes known

No markov chain for you, sorry!

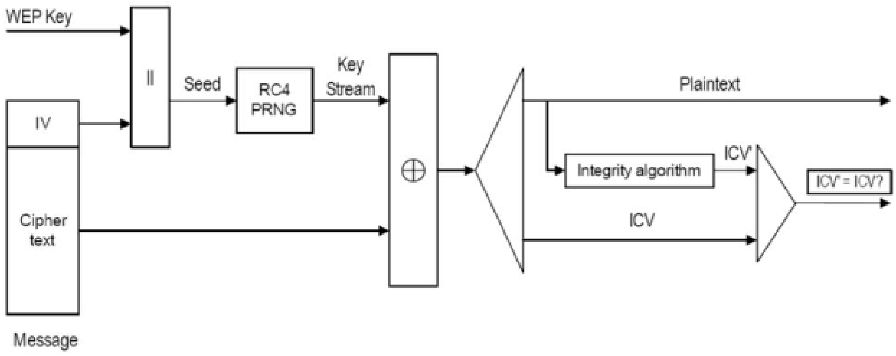
WEP Encryption and Decryption

Encrypt:



- || means "concatenation"
- RC4 PRNG seed = IV || WEP key
- RC4 PRNG generates keystream from the seed
- CRC of the message (Cyclic Redundancy Check) is computed for integrity protection
    - – Error correcting code (generalization of parity bit)
- Message and CRC are then XORed with the keystream
- IV sent together with the ciphertext in the clear to enable decryption

Decrypt



Attack on WEP: - Cryptographic Attacks

Real-Time Key recovery possible
Weaknesses in RC4 and WEP IV management led to more efficient cryptographic attacks

- Problem/Security Issue: PRNG seed = IV || WEP key
- Key recovery possible after capturing some traffic

Attack on WEP: Non-crypto Attacks

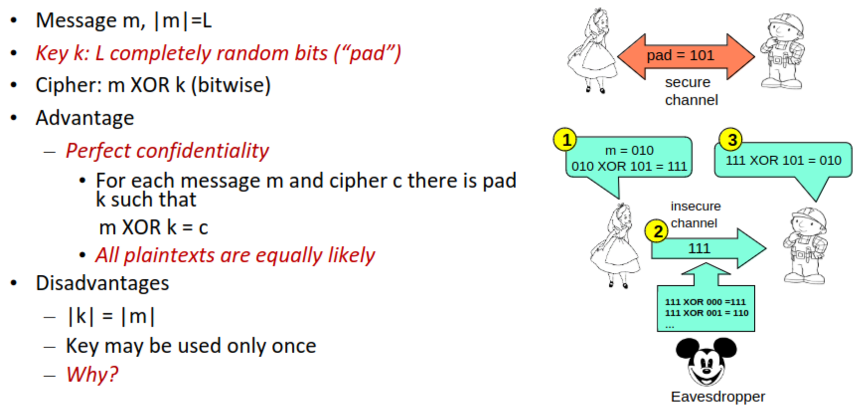
Exist but were not explained in the lecture.
Attacks that do not cryptographically break RC4

Need to know:

WiFi Goals (8 parts)
Infrastructure and Ad-hoc Networks
Wi-Fi Architecture (6 parts)
Connecting to Wi-Fi Networks (3 different ways)
Authentication in Wi-Fi (2 different possibilities)
Scanning for Wi-Fi Networks (3 different ways)
Attack: Alien STA joins network
Attack: Evil Twin Attack
WEP (Goal, how it works)
WEP Encryption and Decyrption
Attack on WEP: - Cryptographic Attack

6. More Wi-Fi: WEP

One Time Pad



- Message m, |m|=L
- *Key k: L completely random bits (“pad”)*
- Cipher: m XOR k (bitwise)
- Advantage
  - *Perfect confidentiality*
    - For each message m and cipher c there is pad k such that
      m XOR k = c
    - *All plaintexts are equally likely*
- Disadvantages
  - |k| = |m|
  - Key may be used only once
  - *Why?*

Cracking Security if Two Times Pad is used

Encrypt:

- m1 XOR k = c1
- m2 XOR k = c2

If c1 and c2 known how to find out m1 and m2?

- c1 XOR c2 = m1 XOR m2

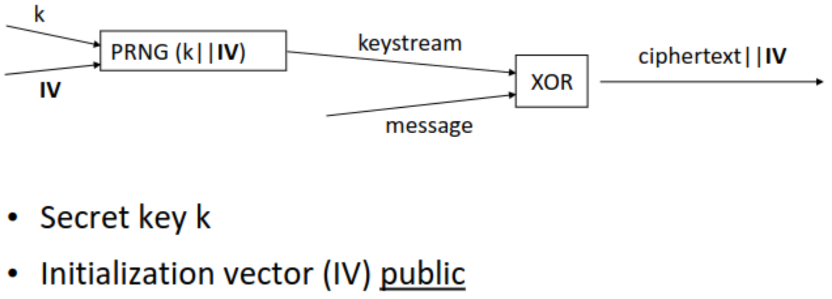
m1 XOR m2 is not (pseudo)random! (in contrast to c1 and c2)

- Can be decrypted via Crib Dragging

Crib Dragging:

- Assume that some common word (e.g., “hello“) appears in m1, starting with position 1
- XOR “hello“ to (m1 XOR m2) starting with position 1
- Result: assumed first 5 characters of plain text of m2
- Try different words for the start of the message

WEP: Initialization Vector (IV)



- Secret key k
- Initialization vector (IV) <u>public</u>

- IVs are used to produce different keystream for each wireless frame
- Sent in the clear together with the frame
- IV needs to be public and unique (= public one-time-pad)

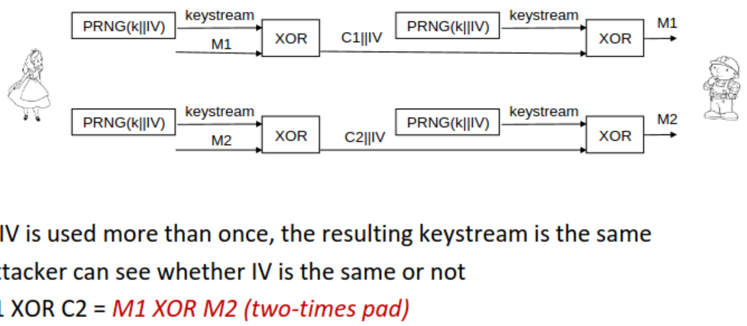
- Why is IV needed: Receiver needs additional information that indicates where to start the PRNG

Attack on WEP: IV Reuse

-

Serious problems if the same IV is used with the same key more than once => “two-times” pad

C1 XOR C2 = (M1 XOR keystream) XOR (M2 XOR keystream) = M1 XOR M2



- If IV is used more than once, the resulting keystream is the same
- Attacker can see whether IV is the same or not
- C1 XOR C2 = *M1 XOR M2 (two-times pad)*

Attacks on WEP: Two other IV-based Vulnerabilities

1. PRNG Restart
    a. WEP PRNG may be restarted every time a laptop is restarted
    b. At restart, IV is set to 0 and incremented with every sent packet
2. IV too short
    a. WEP IVs are 24 bit long
    b. => IVs are reused after around 7 hours (11Mbps sends 700 packets per sec)
    c. APs use the same key for months and years
    d. Many messages are encrypted with the same keystream

Attack on WEP: CRC-based Attack

= Attack on Message Integrity

CRC (Cyclic Redundancy Check):

- a hash function that generates a checksum based on the contents of the data packet or message.
- WEP uses CRC as error-detection code
- CRC is not cryptographically secure, only catches random bit flips, but not bit flips by an intelligent attacker
- CRC is a linear function: CRC(a XOR b) = CRC(a) XOR CRC(b)

The Attack

Attackers exploit weaknesses in CRC by crafting specially manipulated packets that maintain CRC integrity, allowing them to modify packet contents without detection. Due to this vulnerability, attackers can launch attacks such as packet injection or modification, compromising the integrity of the WEP-encrypted communication.

Cryptographic Details about the attack

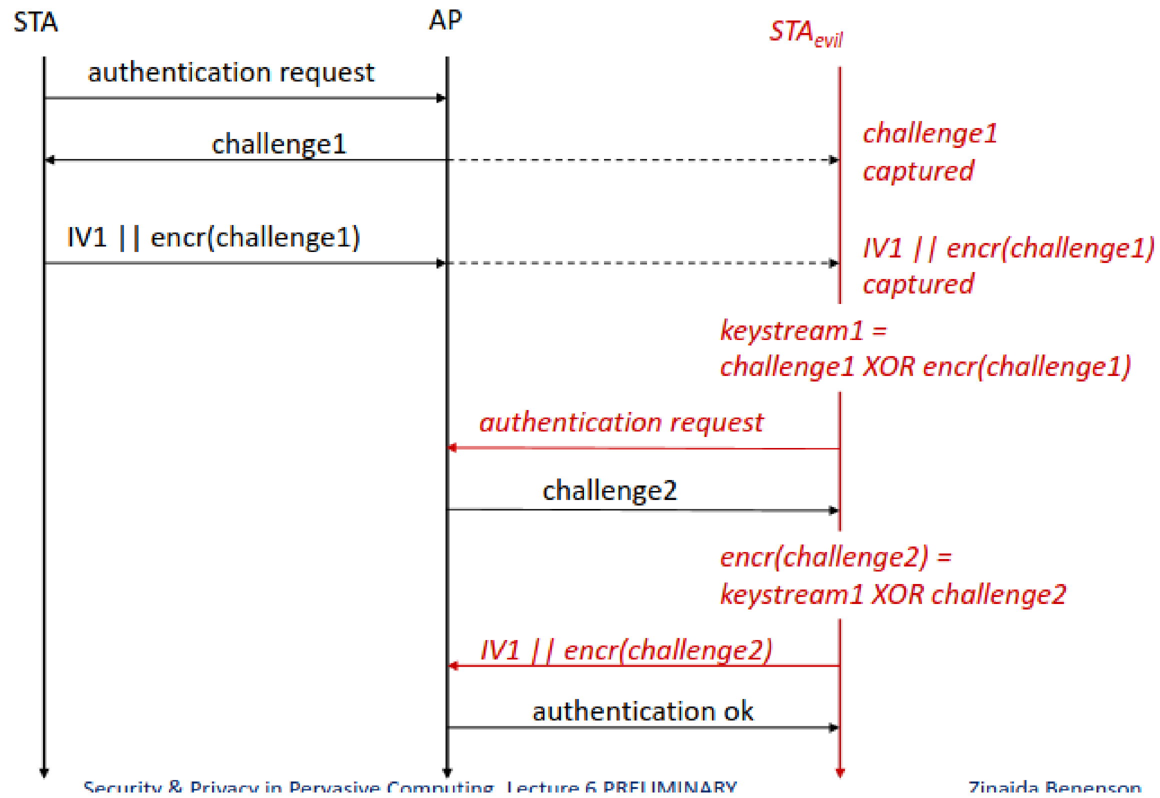
Consider M’ = M XOR d (d can be arbitrarily chosen)

- M not known but construction of a meaningful M’ still possible
  - Example: flip a bit in a message with a payment amount
  - Send M’ with the same IV as M
- The new manipulated message can be calculated via: C XOR (d||CRC(d))
  - Where C is the original message: C = keystream XOR (M || CRC(M)

Attack on WEP: Replay Attack on Access Control

Shared Key Authentication with/without encryption

Challenge-response: demonstrate possession of the WEP key


STA
AP
STA_{evil}
authentication request
challenge1
challenge1 captured
IV1 || encr(challenge1)
IV1 || encr(challenge1) captured
keystream1 = challenge1 XOR encr(challenge1)
authentication request
challenge2
encr(challenge2) = keystream1 XOR challenge2
IV1 || encr(challenge2)
authentication ok
Security & Privacy in Pervasive Computing, Lecture 6 PRELIMINARY
Zinaida Benenson

Summary: 4 WEP Design Issues

Key management

- Global master key per ESSID
  - If key leaks, key replacement in all devices is needed

    - No key management protocol for key replacement
    - No session keys, master key directly used
        - Large amount of traffic is encrypted with the same key
        - Combined with other weaknesses, leads to attacks

IV management

- IV size too small (24 bits): reuse
- Real key size small: WEP keys are 54bit and 128 bit small AND!!! include IV
    - real key is 40 bits long and not 64 => Real Time brute force
    - real key is 104 bit long, no brute-force, but cryptographic attack in real time

Cryptography:

- RC4: flawed usage of IVs makes cryptographic attacks possible
- RC4 is by now considered insecure, but WEP weaknesses could be exploited even before the latest RC4 flaws were discovered
- No cryptographic integrity protection (only CRC) Message change possible

Authentication protocol design

- Replay protection is not guaranteed

Implementations

- IV reuse on restart

WEP: 4 Lessons Learned

3. Don’t use master keys directly to encrypt communication

- Integrate key management into the system
- Key distribution and update

4. When using cryptographic algorithms, always ask experienced cryptographers how to do this properly:

- WEP uses RC4 in an inappropriate way
- Be extremely careful when using stream ciphers
- Think about reuse of initialization vectors and other components that should be used only once

5. Consider replay attacks

6. Always use cryptographically secure integrity protection

- Shared secret key: MAC = Message Authentication Code
- Public key crypto: digital signatures

Need to know:

One Time Pad
Two Times Pad: Why it is insecure + Crib Dragging
WEP: IV
Attack: Three Attacks on WEP by abusing IV (prerequisite and attack)
Attack: CRC-based attack on WEP
Attack: Replay Attack on Access Control of WEP
4 WEP Design Issues
WEP: 4 lessons learned

6. More Wi-Fi: WPA and WPA2

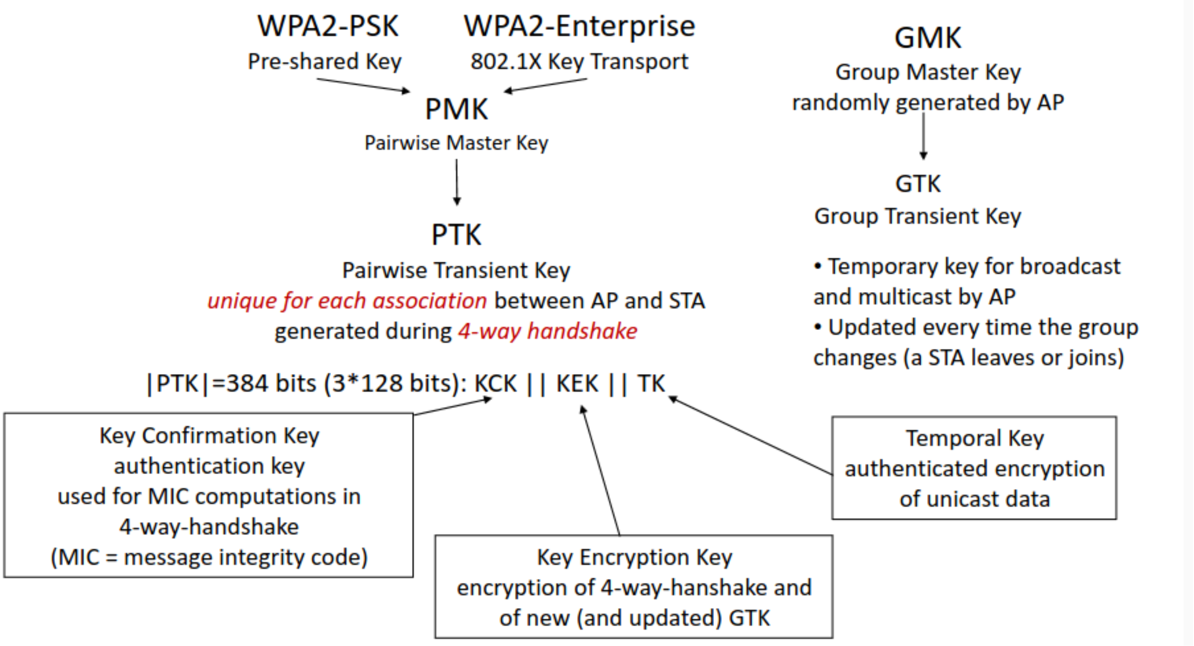
WPA and WPA 2 (Wireless Protected Access)

Improvements of WPA and WPA2 in comparison to WEP

- 128-bit keys, 48-bit IVs
- Temporal keys (derived from master key)
- Cryp. secure MIC (message integrity check)
- 4-way handshake: Authentication and key management

WPA: uses RC4 (temporal measure till WPA2 came out), worked on WEP hardware
WPA: uses AES in CCM mod for authenticated encyrption, required new hardware

WPA2 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)

WPA2-PSK (Pre-Shared Key)

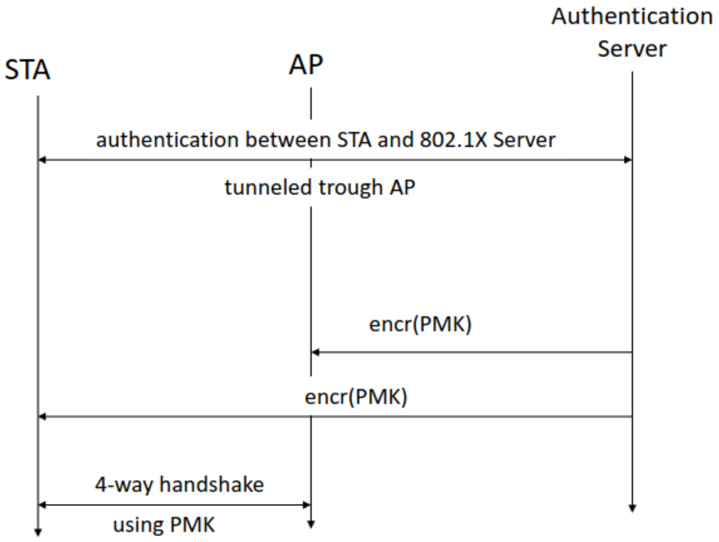
- PMK = PBKDF2(passphrase,salt)
- Salt: ESSID (network name)
- Passphrase: 8 to 63 printable ASCII characters
- PBKDF2: Password-Based Key Derivation Function 2
  - Cryptographically secure function for password generation

WPA2-Enterprise

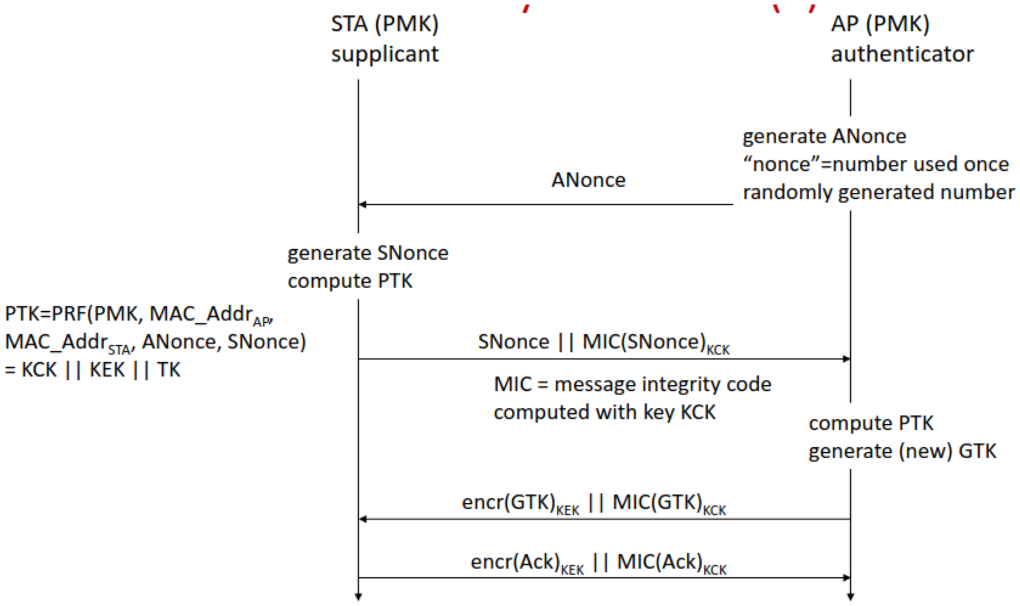
Individual PMK for each STA-AP pair and each session
EAP: Extensible Authentication Protocol

- Authentication and key agreement

- Many variations (Asymmetric/hybrid cryptography (TLS), Symmetric cryptography, Password authentication)
- 802.1X Server = Authentication Server



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

Best practices for Authentication and key management protocols

- Clear security goals and trust assumptions
- Protocol messages should include names of all participants
- Generated cryptographic keys should depend on input by all untrusted participants
- Different keys should be used for encryption and for authentication
    - Exceptions: specifically designed authenticated encryption algorithms, such as AES-CCM used in WPA2 and ZigBee

- Provide replay protection: should not be possible to use message from one protocol instance in another protocol instance
    - Rules for “proper” usage of nonces or timestamps

Attack: WPA2-PSK Key Cracking

Attack:

- PMK = PBKDF(password, salt),   where salt is the ESSID (=network name)
- Capture handshake
- Try out passwords using a dictionary
- Rainbow tables precomputed for some most popular network names
- sid, linksys, NETGEAR, default, …

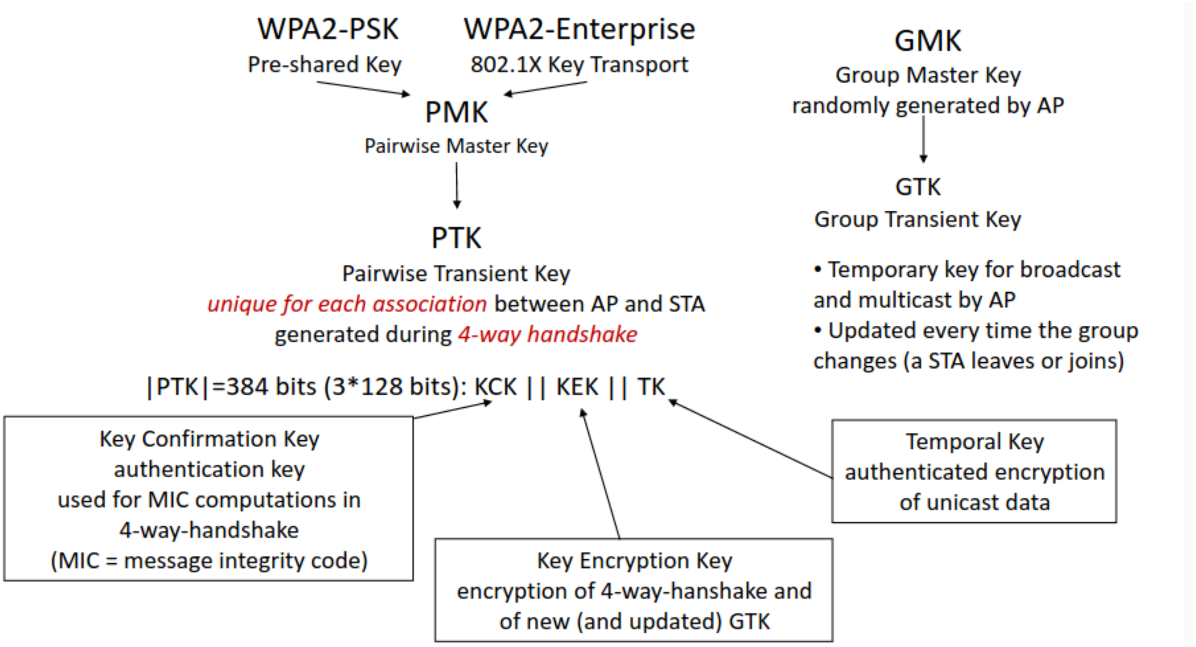
Defense:

- Option 1: SSID: unique, long, sufficiently non-popular…
- Option 2: PSK Passwords
    - Cracking can be done offline + salt is known
    - Depends on computing power
    - Strong passwords make it hard/impossible to crack
    - But: User harassment with strong passwords
- Option 3: Difference to user-selected passwords for online services
    - Offline attacks only possible with database leak
    - Passwords are hashed and salted, salts are unpredictable
    - More defense mechanisms in backend
    - But: User harassment with strong passwords

Attack: Insider Attacks on WPA2-PSK

- Each insider STA can eavesdrop if it captures ANonce and SNonce of another STA
- Because the same PMK is used for all STAs

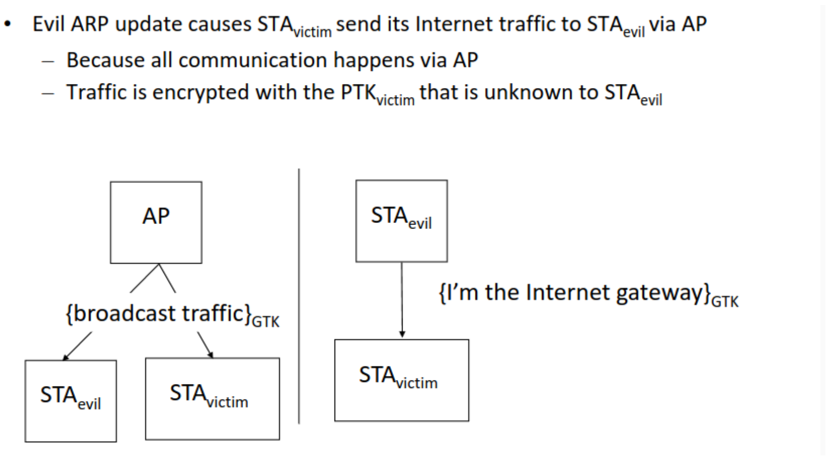
Attack: Insider Attack of WPA2 via Hole 196


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)

Attack:

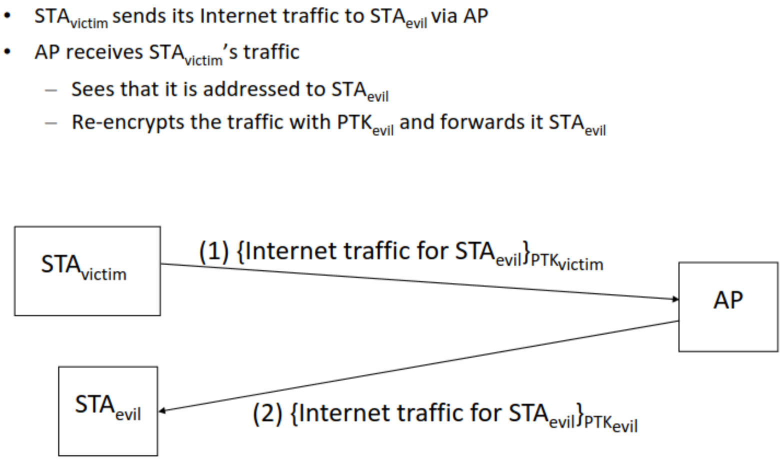
Can be used to eavesdrop traffic even in WPA2-Enterprise

- Individual PTKs for each AP-STA pair are used, but the group key GTK used for broadcast by the AP
- STA$_{evil}$ impersonates AP (using AP's MAC address)
  - Sends false *ARP updates* encrypted with GTK, announcing STA$_{evil}$ as Internet gateway
  - "IP address of the gateway maps to my MAC address"
- ARP: address resolution protocol
  - Translates IP addresses to local Ethernet addresses
- All other STAs start sending their Internet traffic via AP to the fake gateway
- AP decrypts all traffic and re-encrypts it for STA$_{evil}$
  - Because the traffic is destined to the attacker's MAC address
- Result: STA$_{evil}$ is Man-in-the-Middle for Internet access



- Evil ARP update causes STA$_{victim}$ send its Internet traffic to STA$_{evil}$ via AP
  - Because all communication happens via AP
  - Traffic is encrypted with the PTK$_{victim}$ that is unknown to STA$_{evil}$

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



- STA$_{victim}$ sends its Internet traffic to STA$_{evil}$ via AP
- AP receives STA$_{victim}$'s traffic
  - Sees that it is addressed to STA$_{evil}$
  - Re-encrypts the traffic with PTK$_{evil}$ and forwards it STA$_{evil}$

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

Defense:

- Static ARP tables, or monitoring of ARP tables at the STAs
- Wireless intrusion detection systems (e.g., at APs)
  - Attack can be only detected in the air traffic!
- Change WPA2?
  - Usage of individual keys instead of GTK
  - Usage of digital signatures for authentication of broadcast message

Attack: Key Reinstallation Attack

Mut zur Lücke, den scheiß merk ich mir nicht. Hier ne ChatGPT zusammenfassung

- Exploits a vulnerability in the WPA2 protocol's 4-way handshake process.
- Attackers can force reinstallation of an already-in-use encryption key.
- Occurs due to improper handling of cryptographic handshake messages.
- Allows attackers to decrypt and intercept data transmitted over the Wi-Fi network.
- Attackers can also inject malicious content into encrypted traffic.
- KRACK does not require knowledge of the Wi-Fi network's passphrase.
- Vulnerable devices include those running vulnerable implementations of WPA2, affecting a wide range of devices.
- Mitigation involves patching affected devices and updating Wi-Fi access points and client devices.

WPA2 Security Summary

Key hierarchy

- Master key never used for traffic encryption
- Transient keys for each session
- Individual keys for each STA-AP pair
    - WPA2-PSK: insider attacks possible “by design”
    - WPA2-Enterprise: insider eavesdropping is possible through Hole196

Key management

- WPA2-PSK: pre-shared passwords
    - Password should be able to withstand guessing attacks
- WPA2-Enterprise: authentication server

4-way handshake

- “Kracked”, but can be repaired

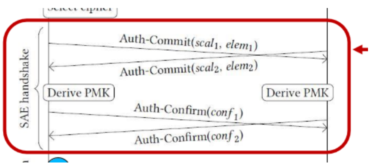
Need to know:

WPA and WPA2 Security Improvements in comparison to WEP (4 parts)
Why WPA is insecure/exists but was still proposed/used
WPA2 Key Hierarchy (7 keys)
WPA2-PSK + PMK + Calculation
WPA2-Enterprise + PMK
Four Way Handshake
Best practices for Auth and key management protocols (4 parts)
Attack: WPA2-PSK Key Cracking (attack + Defense)
Attack: Insider Attacks on WPA2-PSK (Hole 196) + Defense
Attack: Key Reinstallation Attack + KRACK
WPA2 Security SUmmary (3 parts)

7. More Wi-Fi

WPA3

- Designed to fix the WPA2-PSK password cracking attacks
- Uses SAE = Simultaneous Authentication of Equals = “Dragonfly”



  - Executed before the 4-way-handshake
  - Uses public key cryptography
  - For each session: generates a new PMK with high entropy out of the Wi-Fi password
- Based on a well-known provably secure cryptographic protocol called “Dragonfly”

Attack on WPA3: Dragonblood Attacks

- Side channel attacks on SAE which leak information about Wi-Fi password
- Cryptographic Attack
- Password cracking by attempted downgrading to WPA2
  - fMeasures against downgrading are implemented
  - Password cracking still possible

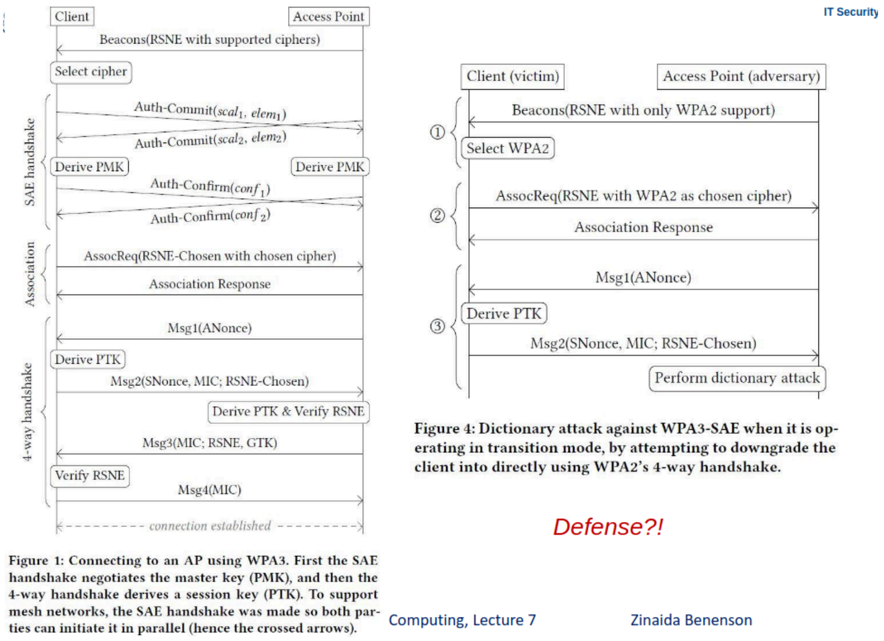


Figure 1: Connecting to an AP using WPA3. First the SAE handshake negotiates the master key (PMK), and then the 4-way handshake derives a session key (PTK). To support mesh networks, the SAE handshake was made so both parties can initiate it in parallel (hence the crossed arrows).

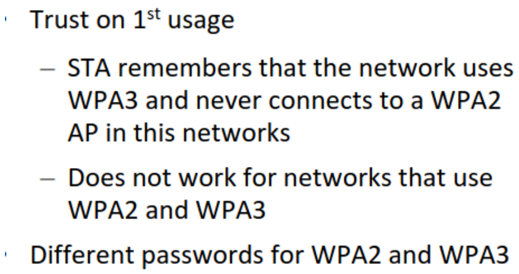
Figure 4: Dictionary attack against WPA3-SAE when it is operating in transition mode, by attempting to downgrade the client into directly using WPA2’s 4-way handshake.

Defense?!

Computing, Lecture 7

Zinaida Benenson

Defenses against Downgrading



- Trust on 1st usage
  - STA remembers that the network uses WPA3 and never connects to a WPA2 AP in this networks
  - Does not work for networks that use WPA2 and WPA3
- Different passwords for WPA2 and WPA3

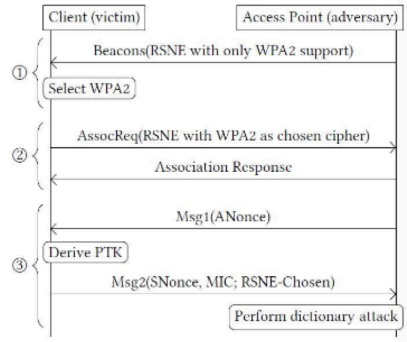

Client (victim)
Access Point (adversary)
①
Beacons(RSNE with only WPA2 support)
Select WPA2
②
AssocReq(RSNE with WPA2 as chosen cipher)
Association Response
③
Msg1(ANonce)
Derive PTK
Msg2(SNonce, MIC; RSNE-Chosen)
Perform dictionary attack

Figure 4: Dictionary attack against WPA3-SAE when it is operating in transition mode, by attempting to downgrade the client into directly using WPA2's 4-way handshake.

Lessons Learned: WPA2/3

- Strong cryptography & established cryptographic protocols offer strong protection
- Designing and implementing interactive security protocols is very difficult
  - Specifications should be very clear and unambiguous (state machines, meticulous pseudo code)
  - Krack attack discovered after 14 years
- WPA2-PSK: User-generated passwords will be weak
  - Brute-forcing passwords should (ideally) not be possible from captured handshakes WPA3
  - Avoid user-generated passwords
- Insider attacks should be considered in the protocol design
- Vulnerabilities in secure protocols can result from unexpected interactions with insecure protocols
  - Hole 196: ARP was never meant to be secure against impersonation
- Design process for cryptographic algorithms should be open
  - WPA3 was designed by the Wi-Fi Alliance without public discussion / comments
- Backwards compatibility usually opens ways for downgrade attacks
  - Downgrade from WPA3 to WPA2 is difficult to prevent in networks that use both

Need to know:

WPA3 (Why it is more secure)
Attack on WPA3: Dragonblood
WPA3: Defenses against Downgrading
7 lessons learned WPA2/3

7. Zigbee

IEEE 802.15.4

Design Goals in comparison to Bluetooth and Wi-Fi

Physical (PHY) + Medium Access Control (MAC) layers

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

ZigBee Versions/Timeline

- ZigBee 2004
- Zigbee 2006 (Cluster library, encryption support, frame authenticity)
- ZigBee Pro (2007): New security model, software features and enhanced security
- ZigBee 3.0 (current): Discontinuation of Profiles, easier and streamlined communication

Clusters & Profiles

- Clusters: sets of message types related to a certain device function, like Protocols, Security, …
- Profiles: sets of message types related to a certain application scenario, like Home Automation (discontinued in ZigBee 3.0)

ZigBee Home Automation (ZHA) Profile

- Interface between devices in a smart home
- Channels don't overlap with Wi-Fi
- Clusters used: General, Lighting, Security, ….
- discontinued in ZigBee 3.0

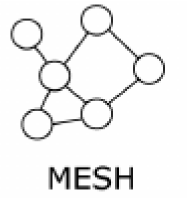
ZigBee Light Link (ZLL) Profile

- For consumer grad lighting devices (smart lamps)
- Similar clusters to ZHA
- Can co-exist with ZHA in theory, in reality it never worked out well due to compatibility issues between different profiles (integrate ZLL smart bulb into a ZHA network)
- discontinued in ZigBee 3.0

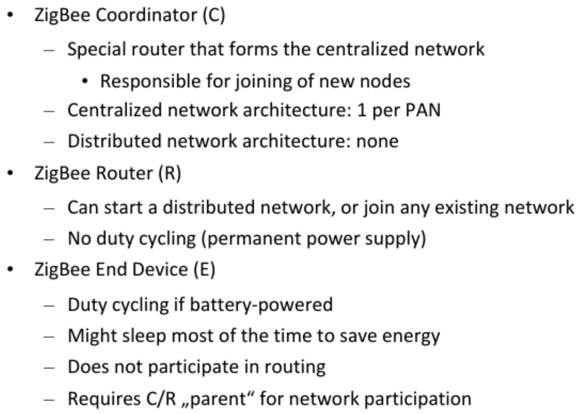
ZigBee PANs (simplified)

- PAN = Personal Area Network

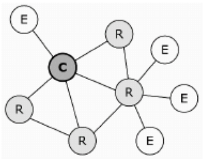
- Each ZigBee deployment consists of at least one PAN, PANs are logically separated (one per household)
- Mesh topology, multiple devices and routing hops, High reliability achieved through multiple paths



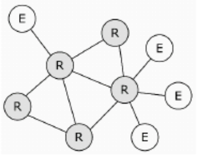
ZigBee Nodes & Networks



- ZigBee Coordinator (C)
  - Special router that forms the centralized network
    - Responsible for joining of new nodes
  - Centralized network architecture: 1 per PAN
  - Distributed network architecture: none
- ZigBee Router (R)
  - Can start a distributed network, or join any existing network
  - No duty cycling (permanent power supply)
- ZigBee End Device (E)
  - Duty cycling if battery-powered
  - Might sleep most of the time to save energy
  - Does not participate in routing
  - Requires C/R „parent“ for network participation



centralized



distributed

Security Goals of ZigBee

- Security goals: ZigBee specifications do not define security goals
  - We define our own one: Legitimate user should always keep control over devices and their data
- Confidentiality: Only legitimate entities are allowed to access data and commands sent within the network.
- Integrity: Data and commands sent within the network are not tampered with.
- Availability: The functionalities and data of the devices in the network are continuously available to all legitimate entities.
- Authenticity: The receiver is able to reject commands and data sent by illegitimate entities.

Attacker Model

- Can eavesdrop on wireless communication
- Can inject packets in wireless communication
- Cannot access nodes physically
- Cannot access nodes remotely via Internet

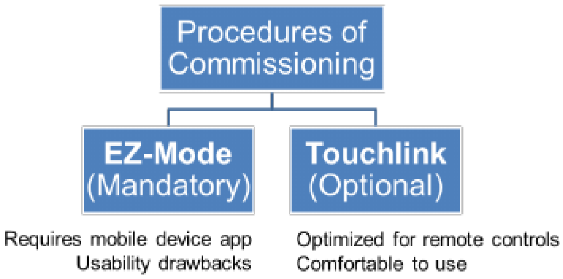
Keys in ZigBee 3.0

Network key (shared among all devices in a PAN)

- = Each PAN uses a shared symmetric key for communication
- Authenticated encryption: AES-CCM (state-of-the-art)
- Node starting a new network generates the network key for this network

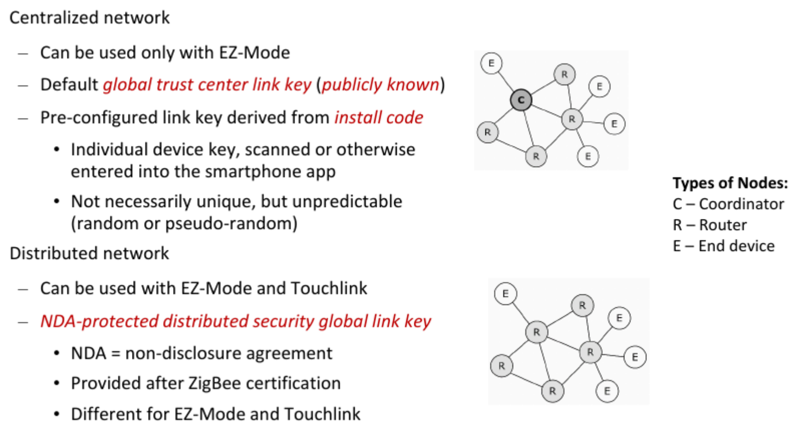
Link Key (used for commissioning)

- Commissioning: process of starting a new network or joining a new node to an existing network
- Network key is transmitted to newly joining devices using a link key
- Secure commissioning = cornerstone of ZigBee network security



Procedures of Commissioning
EZ-Mode (Mandatory)
Touchlink (Optional)
Requires mobile device app
Usability drawbacks
Optimized for remote controls
Comfortable to use

Network Types and Link Keys



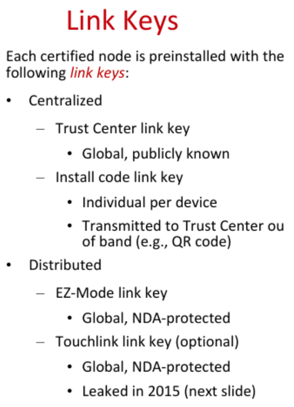
Centralized network

- Can be used only with EZ-Mode
- Default *global trust center link key* (*publicly known*)
- Pre-configured link key derived from *install code*
  - Individual device key, scanned or otherwise entered into the smartphone app
  - Not necessarily unique, but unpredictable (random or pseudo-random)

Distributed network

- Can be used with EZ-Mode and Touchlink
- *NDA-protected distributed security global link key*
  - NDA = non-disclosure agreement
  - Provided after ZigBee certification
  - Different for EZ-Mode and Touchlink

Types of Nodes:
C – Coordinator
R – Router
E – End device



Link Keys

Each certified node is preinstalled with the following *link keys*:

- Centralized
  - Trust Center link key
    - Global, publicly known
  - Install code link key
    - Individual per device
    - Transmitted to Trust Center ou of band (e.g., QR code)
- Distributed
  - EZ-Mode link key
    - Global, NDA-protected
  - Touchlink link key (optional)
    - Global, NDA-protected
    - Leaked in 2015 (next slide)

Touchlink (ZLL) Master Key

- Protected by NDA
- Leaked on Twitter in March 2015

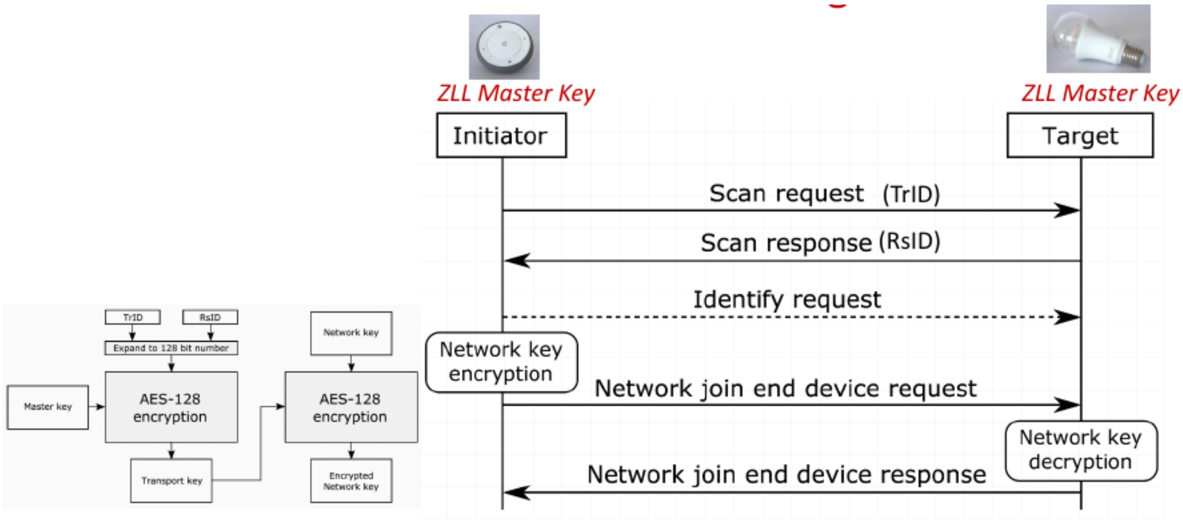
EZ-Mode Commissioning

Invoked by user action (e.g., pushing button on IoT device)

1. IoT device scans for open networks to join (for 3 minutes)
2. If network is open, it responds to device with network information

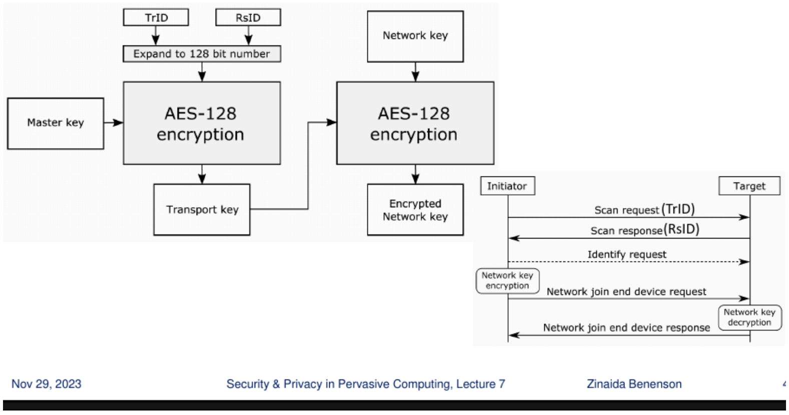
3. IoT devices decides wether to joint he network and which link key to use
4. Network key encrypted with link key is transferred to the joining IoT device

Touchlink Commissioning



- Initiator: usually remote control or router
- Target: light bulb or other ZigBee device with dedicated functions
- Both possess the NDA-protected Touchlink link key, aka:
    - ZLL (ZigBee Light Link) Master Key
- Exchange of identifiers (individual per commissioning event)
    - Initiator sends TrID: transaction identifier, 32-bit, randomly generated
    - Target sends RsID: response identifier, 32-bit, randomly generated
- Identify request: initiator asks target to identify itself if many targets are available
    - Light bulb: blink several times
    - Optional operation
- Network join end device request
    - Initiator sends network key (NWK) to target encrypted with the master key
    - TrID and RsID are used to make encrypted message different for each commissioning

Network Key Encryption

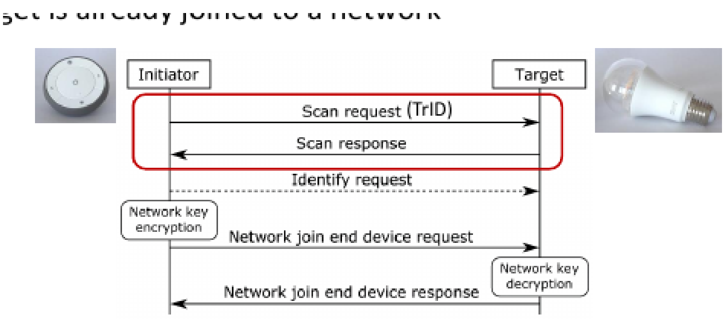


Nov 29, 2023 Security & Privacy in Pervasive Computing, Lecture 7 Zinaida Benenson

Inter-PAN frames

- Special type of ZigBee frames to transmit touchlink command
- Transmission neither encrypted nor authenticated
- => can be abused

Active Device Scan



Not an attack, but a prerequisite for all other attacks

- Touchlink commands are accepted by targets only if previously they received a scan request with the same TrID (transaction ID)
- Scan for touchlink-enabled devices in the wireless range
  - Works even if target is already joined to a network

Attack: Identify Action Attack

- Doesn’t require knowledge of any cryptographic material
- Trigger identify action (e.g., blinking, beeping, dimming) of target device
  - Even if the device is already in a network
  - No authentication (inter-PAN frame)
- Field to specify duration: 16bit ≈ 65000 seconds ≈ 18 hours => Bulb blinks until it runs out of battery, blocks other operations of the lamp
- Recovery: manually disconnect from power source

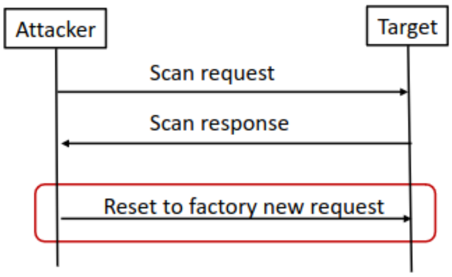
Need to know:

Design Goals of Zigbee
Clusters, profiles, ZHA and ZLL
ZigBee Pans
ZigBee Nodes & Networks
ZigBee 3.0: Networks Key
ZigBee 3.0: Link Key
ZigBee 3.0: Network Types, used keys, which key is known by nodes in which network type
ZLL Master Key
EZ Mode Commissioning
Touchlink Commisioning
Network Key Encyrption
InterPan Frames (what are they, how can they be abused?)
Active Device Scan
Attack: Identify Action Attack

8. More Zigbee

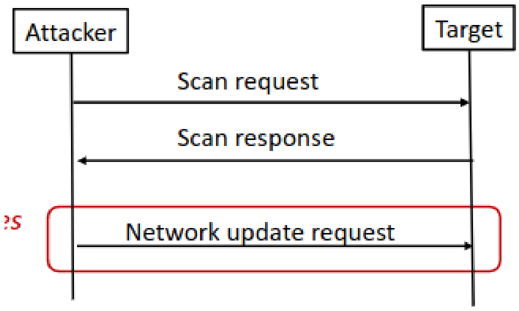
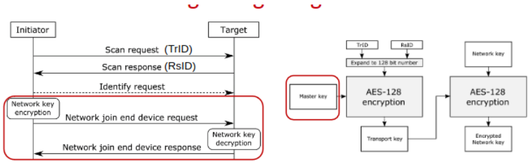
Attack: Reset to Factory-New Attack

- Doesn’t require knowledge of any cryptographic material
- Reset target to the factory-new state
  - Even if the device is already in a network
  - No authentication (inter-PAN frame)
- Threat scenario: access to restricted area
  - Touchlink-enabled door lock
- Reset to factory-new door probably unlocks
- Recovery: recommission the affected devices



Attack: Permanent Disconnect Attack

Two attack possibilities

Change wireless channel of target: network update request	Join target to garbage network
 Attacker Target Scan request Scan response Network update request	 Initiator Target Scan request (TrID) Scan response (RsID) Identify request Network key encryption Network join end device request Network key decryption Network join end device response TrID RsID Expand to 128 bit number Master key AES-128 encryption Transport key Network key AES-128 encryption Encrypted Network key Network key is encrypted using AES-Encryption in ECB-mode Not AES-CCM (authenticated encryption) as for network communication AES-ECB does not support authentication, only encryption (no integrity protection) Attack: – Send a random 128-bit garbage to the target – Target will decrypt “garbage” and join non-existing network with unknown network key

Recovery: physical reset

- Osram Lightify: turn on 3 seconds, off 5 seconds repeat five times
- Philips Hue: no physical reset possibility found, possibly no user- driven recovery
- Attacker can recover anytime using the same toolkit as for the attack

Possible threat scenarios: DoS, ransom

Attack: Hijacking - Attack with knowledge of the leaked global master key

- Active attack: requires interaction
- Join target to attacker‘s network
- Send commands: turn on/off, change color, open/close (e.g., door lock)
- Works even if the device is already joined to another network

Attack: Network Key Extraction - Attack with knowledge of the leaked global master key

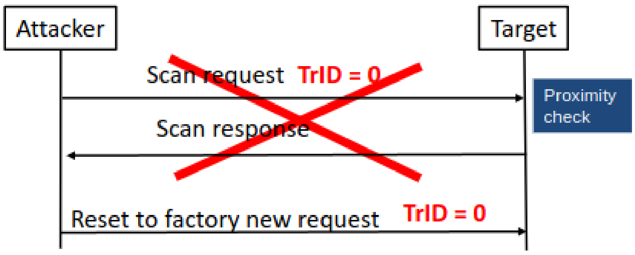
- Passive attack: eavesdropping on touchlink commissioning
- GE and Osram: User has no interface (on smartphone app) to trigger Touchlink
- How long should the attacker wait till user commissions a device?
    - „Motivate“ user to re-commission any device by reset-to-factory-new attack

ZigBee Proximity Check

Limits range of accepting touchlink commands
If receiving signal strength (RSS) > predefined threshold, then send scan response

Bug in Proximity Check in Smart Bulbs

Affected Smart Bulbs of Philipps, Osram and GE



Attack: Using Proximity Chek Bug to factory-reset Devices

- A scan request with TrID=0 for a scan request is invalid
  => rejected if received with scan request
- All other inter-PAN commands, if sent with TrID=0, are accepted by the bulb without proximity check (as result of a programming bug)
- Can reset any bulb to factory new without previous scanning and without proximity check
- If a bulb is reset to factory new, it can be joined to new networks without proximity check

Conclusion of ZigBee Security Analysis

Touchlink commissioning in insecure by design

- A single touchlink device in the network can expose network key

- Global master key cannot be renewed due to backwards compatibility requirements

Recommendations

- Disable touchlink in all ZigBee 3.0 products
- Use EZ-Mode commissioning with install codes (although it has usability drawbacks)

Lessons Learned ZigBee (3+3)

- Never use a global master key
- Never use signal strength for proximity verification
- It takes a very looooong time to fix bugs in software of IoT devices

Lessons confirmed

- Always precisely define security goals and attacker model
- Security by obscurity fails (NDA-protected keys)
- Encryption without integrity protection fails (AES-ECB networks key encryption

Establishing Economic Incentives for Security Patching of IoT Consumer Products

Problem

- Consumers don’t want to pay for security
- Manufacturers are unwilling to provide security
- => Regulation needed to avoid those two problems

Solution: Provide officially regulated “security Label”

- Add options for reliably patching vulnerabilities
- Manufacturers have to define update policy for each product, printed as labels on each product and packaging
    - Madatory “No Security updates” label if Manufacturer doesn’t provide updates at all

Problems:

- Ineffective if flaws can’t be patched, Outsourcing debuggin to consumers (brand image damage), Low User Acceptance (do consumers care about updates)
- Users see smart camera etc as more dangerous as for example smart weather stations, even if both could allow atatckers equal entries into a network and other devices

Need to Know:

Attack: Reset to Factory-New Attack
Attack: Permanent Disconnect Attack (two ways)
Attack: Hijacking - Attack with knowledge of the leaked global master key
Attack: Network Key Extraction - Attack with knowledge of the leaked global master key
Attack: Using Proximity Chek Bug to factory-reset Devices
Conclusion of ZigBee Security Analysis
Lessons Learned ZigBee (3+3)

9. Bluetooth and Device Pairing

Device Pairing

= Key establishment between two devices

- Devices do not share any common secrets
- “Have never met before”

Security goals: secure channel and CIA

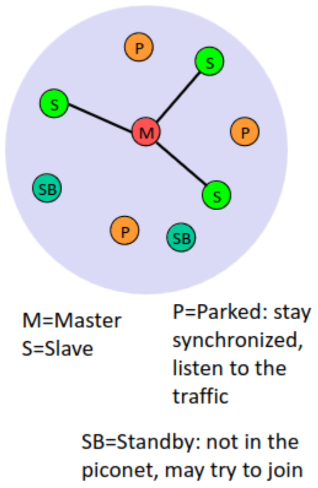
- Authentication of both communication partners
  - No MitM (man-in-the-middle) attacks, no impersonation (Evil Twin)
- Integrity + confidentiality of communication

(Original) Bluetooth Idea

- Wireless Personal Area Network (WPAN)
- Universal radio interface for ad-hoc wireless connectivity
- Short range (10 m), low power consumption
- Voice and data transmission, approx. 1 Mbit/s data rate
- Comparison to Infrared:
  - + Wi-Fi experience can be used
  - + Larger coverage than IR (BT can penetrate walls)
  - + Better bandwidth then IR
  - - limited free frequencies (IR doesn’t need licence)
  - - Shielding of BT more difficult (IR can easily blocked)
  - -More interference with other electrical devices

Later versions of Bluetooth also had a low energy mode, better and more secure pairing/cryptography), better IoT support and functionality

Bluetooth Piconet (System Architecture)



M=Master
S=Slave

P=Parked: stay synchronized, listen to the traffic

SB=Standby: not in the piconet, may try to join

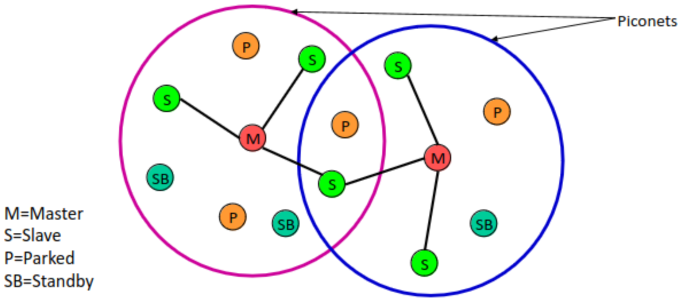
- Collection of bluetooth devices (4 node types)
  - Connected in an ad hoc fashion

    - One unit acts as master and the others as slaves for the lifetime of the piconet
    - Master coordinates traffic, slaves have to synchronize
    - Standby: not in the net, may try to join
    - P: Parked - Stay synchronized, listen to traffic
- Frequency-hopping spread spectrum radio technology (800 hops per second)
    - Each piconet has a unique frequency hopping pattern
    - Depending on master's Bluetooth address (48-bit unique ID)
    - first 16 bits: "non-significant address part": used for
    - frequency hopping
    - next 32 bits: "significant address part": used for other
    - Bluetooth algorithms, including security
- Participation in a piconet = synchronization to hopping sequence
    - Each piconet has one master and up to 7 simultaneous slaves (> 200 could be parked)

Bluetooth profiles

- Represent default solutions for a certain usage model - Basis for interoperability
- Examples: LAN Access Profile, File Transfer Profile, Cordless Telephony profile

Bluetooth Scatternet (System Architecture)



- = Linking of multiple co-located piconets through the sharing of common master or slave devices
- Devices can be slave in one piconet and master of another

Bluetooth Security vs GSM/UMTS/LTE and Wi-Fi

- GSM / UMTS / LTE
    - Predefined and centralized security associations
    - Registered users, SIM cards with preloaded keys
- WiFi
    - Managed security associations (infrastructure mode)
- Bluetooth
    - Ad hoc, spontaneous, unmanaged security associations
    - Device pairing
        - Secure channel between intended devices
        - Key establishment

Legacy Bluetooth Security

- Symmetric Keys, 128-bit
- Encryption based on SAFER+ algorithm
  - Attacks known, less secure than AES
- Custom Stream cipher with man theoretical attacks
  - Actual cipher strength: 60 bit, even with 128 bit key

Bluetooth Key Hierarchy

Initialization key (function of PIN)

- Temporal key for link key establishment protocol
- Used when devices meet for the first time, or if link key “forgotten”
- Devices can only store a limited number of link keys

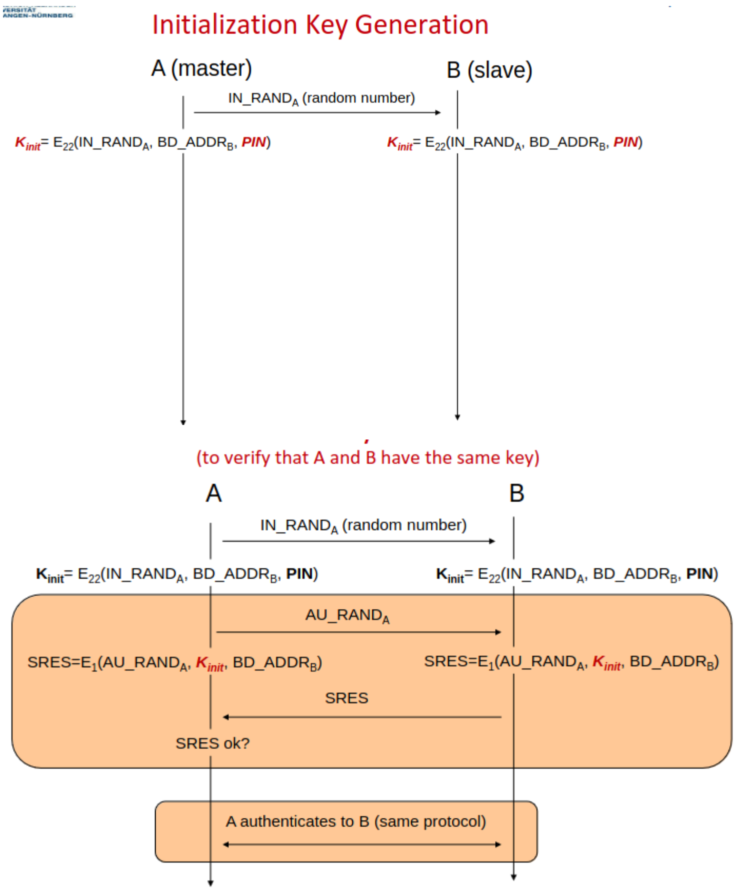
Link key

- Generated by both devices from the initialization key
- When devices meet again, they run a protocol to prove the possession of the link key to each other

Encryption key

- For data encryption, generated from the link key

Initialization Key Generation and Authentication



Initialization Key Generation

A (master)

B (slave)

IN_RAND$_A$ (random number)

K_{init}= E_{22}(IN_RAND$_A$, BD_ADDR$_B$, *PIN*)

K_{init}= E_{22}(IN_RAND$_A$, BD_ADDR$_B$, *PIN*)

(to verify that A and B have the same key)

A

B

IN_RAND$_A$ (random number)

K_{init}= E_{22}(IN_RAND$_A$, BD_ADDR$_B$, **PIN**)

K_{init}= E_{22}(IN_RAND$_A$, BD_ADDR$_B$, **PIN**)

AU_RAND$_A$

SRES=E_1(AU_RAND$_A$, K_{init}, BD_ADDR$_B$)

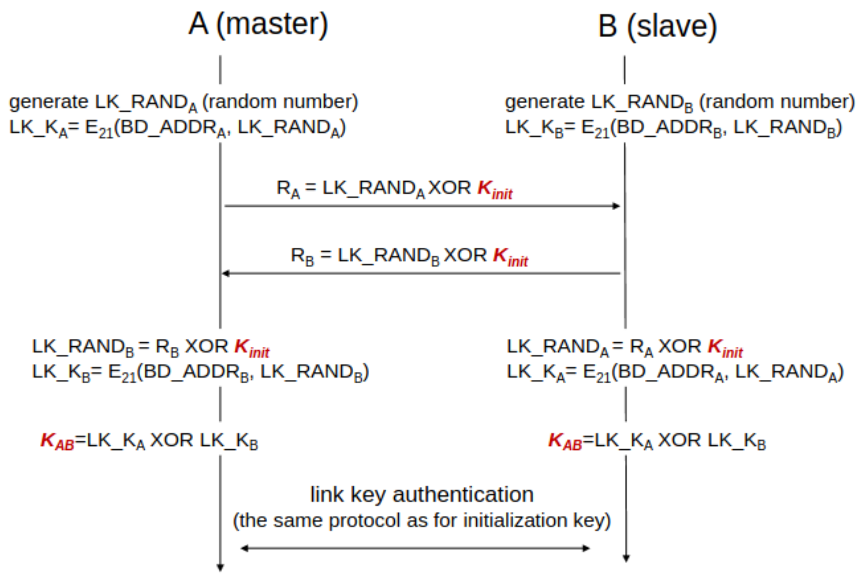
SRES=E_1(AU_RAND$_A$, K_{init}, BD_ADDR$_B$)

SRES

SRES ok?

A authenticates to B (same protocol)

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)

Link Key Update

- Use the same protocol as for the link key generation
- Uses previous link key KAB instead of Kinit
- Helps against attackers that are not present permanently

Encryption Keys

Individual shared key between master A and slave B. How it works:

- Random number generated by A is sent to B
- A and B compute a Key based on the random number and the Link Key

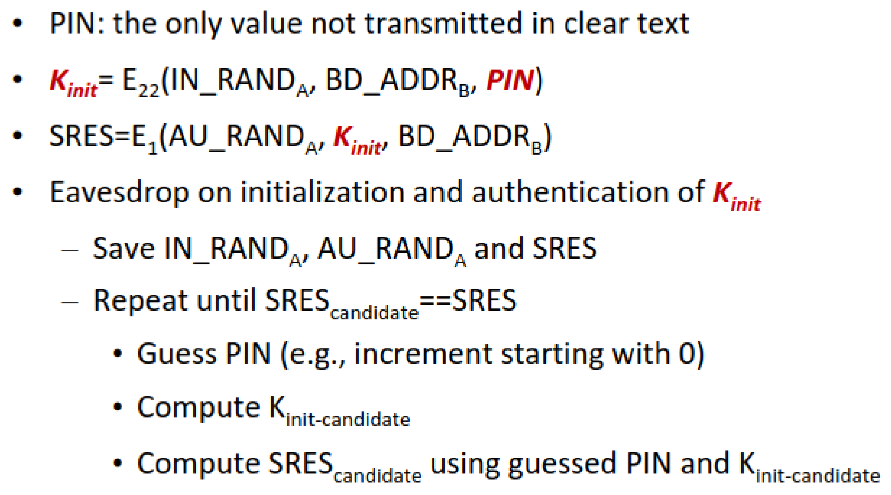
Broadcast Encryption Key. How it works:

- Key generated by master for each session
- Transmitted to slaves using individual shared key

Bluetooth 1.0-2.0 Vulnerabilities

- Secrecy of encryption key depends on PIN
- No PIN management
    - PINs can be too short and/or too simple
    - Especially if user-defined or default
    - Many car hands-free sets use 0000 or 1234
- PIN cracking attacks possible

Attack on older Bluetooth: Passive PIN Cracking



- PIN: the only value not transmitted in clear text
- K_{init}= E_{22}(IN_RAND$_A$, BD_ADDR$_B$, ***PIN***)
- SRES=E_1(AU_RAND$_A$, K_{init}, BD_ADDR$_B$)
- Eavesdrop on initialization and authentication of K_{init}
  - Save IN_RAND$_A$, AU_RAND$_A$ and SRES
  - Repeat until SRES$_{candidate}$==SRES
    - Guess PIN (e.g., increment starting with 0)
    - Compute $K_{init\text{-}candidate}$
    - Compute SRES$_{candidate}$ using guessed PIN and $K_{init\text{-}candidate}$

Attack on older Bluetooth: Active PIN Cracking

Active attack 1

- Initialization key generation: start using any PIN$_X$, compute K_{init_X}
- Initialization key authentication: receive SRES$_B$ from B, perform PIN guessing as above
- SRES$_B$ contain the right PIN!

Active attack 2

- Force A and B to repeat pairing
- Impersonate one if the devices and pretend that the device forgot the link key
- “Forgot key” messages are (necessarily!) sent insecurely

Seven Bluetooth 1.0-2.0 Vulnerabilities

1. Bluetooth devices are often configured to accept connections to arbitrary devices and send their BD_ADDR to them. Suer can be tricked to accept connection
   Attacks: user tracking, malware, data stealing
2. Weak Cryptography (SAFER+ and E_0 weaknesses)
3. Encryption IV reuse (2 times pad, pads repeat after 23 hours of uninterrupted connection)
4. No integrity check in encryption algorithm (CRC used, same as in WEP)
5. No end-to-end encryption (intermediate devices can decrypt traffic)
6. Security can be switched off through negotiation between devices
7. Several Implementation Vulnerabilities and Attacks (BlueSniff, …)
   a. Remote Execution, Man in the middle, etc

Six Bluetooth Lessons Learned (or confirmed)

1. Implement key or PIN management
    a. User-defined “secrets” are usually too weak
    b. Default “secrets” are not secret
2. Use asymmetric crypto for key / PIN management
    a. Symmetric crypto is (always?) susceptible to cracking
3. Use strong cryptography
    a. AES better than SAFER+
4. Protect integrity of encrypted communication
    a. Don’t use CRC for cryptographically secure integrity guarantees
5. Implement properly
    a. Good standards should facilitate good implementations
6. Do not rely on communication range for security
    a. Radio waves propagate beyond the “official” communication range

SSP: Secure Simple Pairing (starting with Bluetooth v2.1)

More secure connection establishment and key exchange via SSP with Diffie-Helmann:

1. Devices initiate pairing process
2. Devices Authenticate themselves to each other
3. Diffie-Hellman is used to securely generate a Encryption Key for further communication

Cryptography: Asymmetric Crypto Key Exchange

Unauthenticated

- Secure against passive eavesdropping
- Vulnerable to man-in-the middle (MitM) and impersonation (Evil Twin) attacks
- Usual realization: Diffie-Hellmann (DH) key exchange

Authenticated

- Secure against MitM & Evil Twin attacks
- Often requires user involvement
- Enter PIN, compare two strings, scan QR code, …
    - Usual realization: authenticated DH key exchange
    - Other possible protocols (not used in Bluetooth): Dragonfly Key Exchange, SAE (Simultaneous Authentication of Equals) many variants, one of them used in WPA3

Diffie-Hellman (DH) Key Exchange

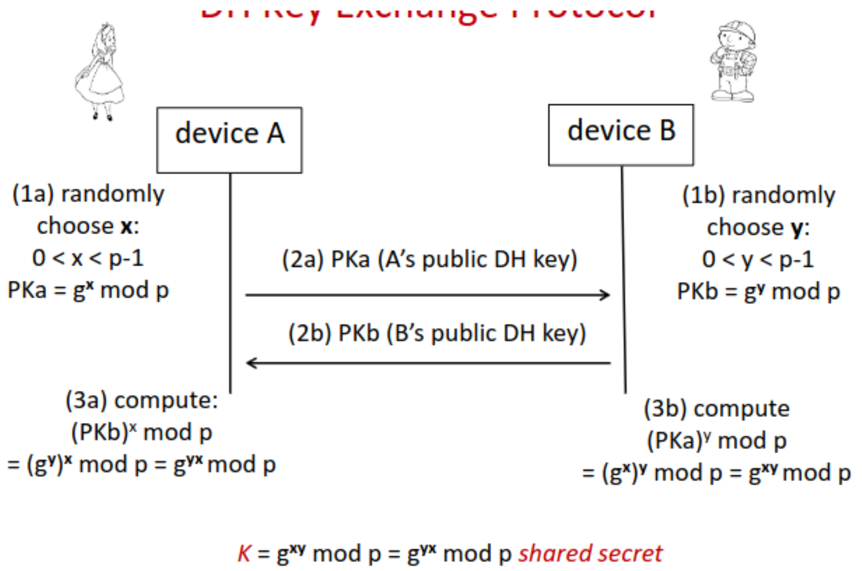
What is DH?

- DH is an asymmetric cryptography process to generate/exchange a key that can then be used for symmetric encryption
- During DH the channel is open, everyone can listen

Public Values

- p: large prime number (1024 bits)
  - Multiplicative group modulo p: {1, 2, …, p-1}
- g: $1 < g < p-1$
  - g is generator of multiplicative group modulo p

DH Key Exchange Protocol



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
$(PKa)^y \bmod p$
$= (g^x)^y \bmod p = g^{xy} \bmod p$

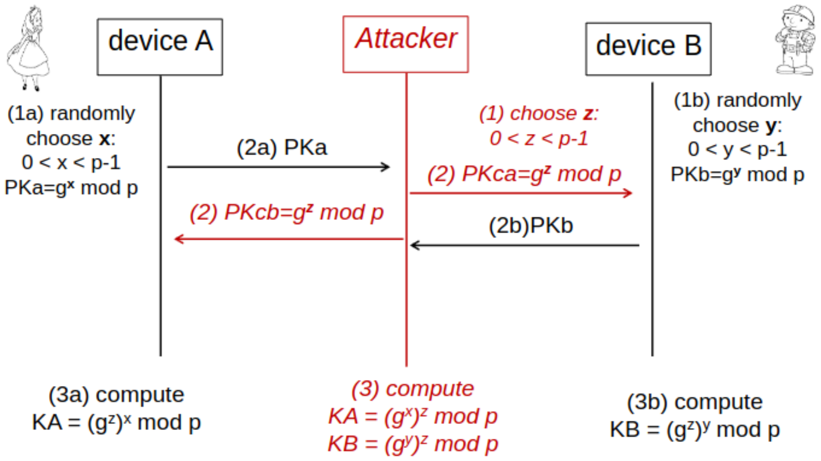
K $= g^{xy} \bmod p = g^{yx} \bmod p$ *shared secret*

DH Key Exchange Security

Save against passive eavesdrop atatcker
Active attacker:

- Can attack and eavesdrop with Man in the middle attack

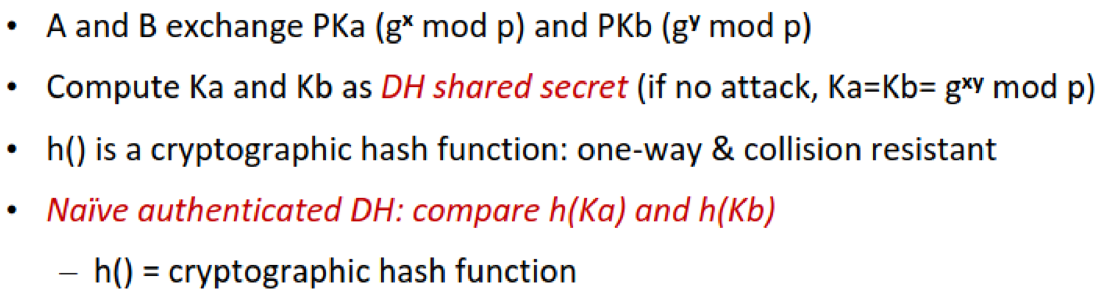
Attack: Man-in-the-Middle Attack on DH



device A
Attacker
device B
(1a) randomly choose **x**: $0 < x < p-1$ PKa=g^x mod p
(1) choose **z**: $0 < z < p-1$
(1b) randomly choose **y**: $0 < y < p-1$ PKb=g^y mod p
(2a) PKa
(2) PKca=g^z mod p
(2) PKcb=g^z mod p
(2b)PKb
(3a) compute KA = $(g^z)^x$ mod p
(3) compute KA = $(g^x)^z$ mod p KB = $(g^y)^z$ mod p
(3b) compute KB = $(g^z)^y$ mod p

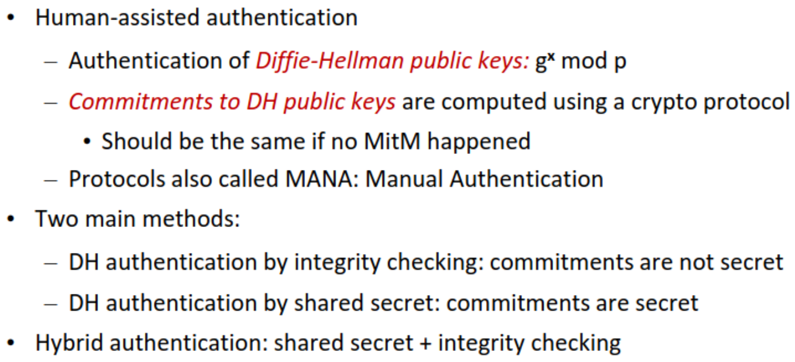
Authenticated DH (+MANA)

=> Against DH MitM attacks
Human-assisted authentication



- A and B exchange PKa (g^x mod p) and PKb (g^y mod p)
- Compute Ka and Kb as *DH shared secret* (if no attack, Ka=Kb= g^{xy} mod p)
- h() is a cryptographic hash function: one-way & collision resistant
- *Naïve authenticated DH: compare h(Ka) and h(Kb)*
  - h() = cryptographic hash function

MitM still possible via a hash collision on h,, but infeasible if output of hash function is long enough. Output needs to be at least 64 bits long



- Human-assisted authentication
  - Authentication of *Diffie-Hellman public keys:* g^x mod p
  - *Commitments to DH public keys* are computed using a crypto protocol
    - Should be the same if no MitM happened
  - Protocols also called MANA: Manual Authentication
- Two main methods:
  - DH authentication by integrity checking: commitments are not secret
  - DH authentication by shared secret: commitments are secret
- Hybrid authentication: shared secret + integrity checking

Need to know:

Device Paring
(Original) Bluetooth Idea
Piconet
Bluetooth Profiles
Bluetooth Scatternet
Bluetooth Security vs GSM/UMTS/LTE and Wi-Fi
Legacy Bluetooth Security
Bluetooth Key Hierarchy
Initialization Key Generation and Authentication
Link Key Generation
Link Key Update
Encryption Keys
Bluetooth 1.0-2.0 Vulnerabilities
Attack on older Bluetooth: Passive PIN Cracking
Attack on older Bluetooth: Active PIN Cracking (2 versions)
Seven Bluetooth 1.0-2.0 Vulnerabilities
Six Bluetooth Lessons Learned
SSP: Secure Simple Pairing (starting with Bluetooth v2.1)
Cryptography: Asymmetric Crypto Key Exchange - Authenticated and Unauthenticated
Diffie-Hellman (DH) Key Exchange (What is DH, Public Values, Exchange Protocol)
DH Key Exchange Security (passive and actzive attacker)
Attack: Man-in-the-Middle Attack on DH
Authenticated DH (+MANA)

10. More Bluetooth

Bluetooth 2.1+ Pairing: Numeric Comparison

Requirement: Both devices must have displays and “yes/no” buttons

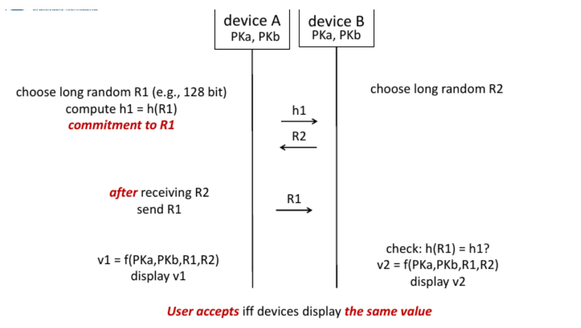
Authenticated DH: integrity checking with the 6-digit number as authenticator

Process:

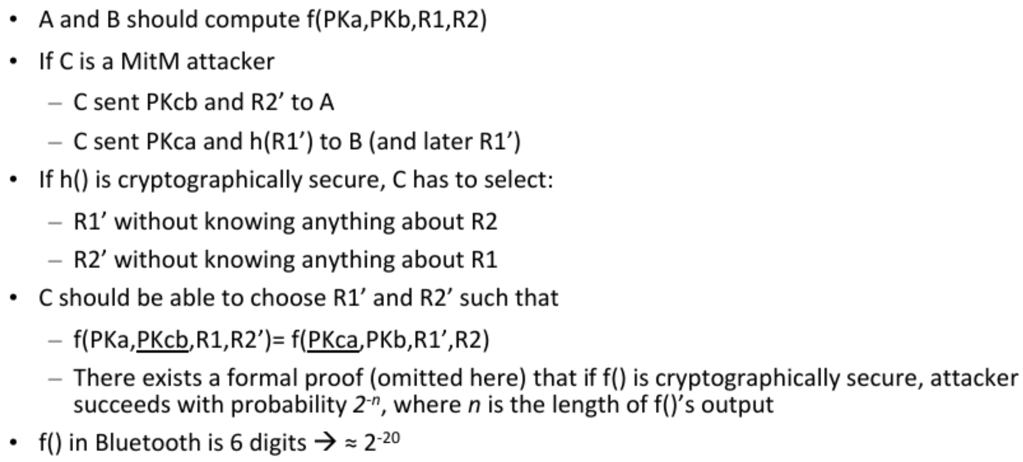
- Both devices must have displays and “yes/no” buttons
- During the pairing procedure, both devices display a 6-digit hash of the public DH keys
- User has to compare the numbers
    - If the numbers are the same, push “yes” on both devices, otherwise push “no”
- This method is provably secure

Basic Numeric Comparison

(Bluetooth Numeric Comparison is a little bit more complicated and secure, we consider a basic protocol

A generates a long random value R1 (e.g., 128 bits), computes h1 = h(R1), and sends h to B B generates a long random value R2, sends it to A A: after having received R2, sends R1 to B B checks if h(R1) = h1 – If yes, B computes v2 = f(PKa,PKb,R1,R2), otherwise it aborts A computes v1 = f(PKa,PKb,R1,R2) Both devices display v1 and v2, user accepts if v1=v2	 device A PKa, PKb device B PKa, PKb choose long random R1 (e.g., 128 bit) compute h1 = h(R1) commitment to R1 choose long random R2 h1 R2 after receiving R2 send R1 R1 v1 = f(PKa,PKb,R1,R2) display v1 check: h(R1) = h1? v2 = f(PKa,PKb,R1,R2) display v2 User accepts iff devices display the same value

Security:



- A and B should compute f(PKa,PKb,R1,R2)
- If C is a MitM attacker
  - C sent PKcb and R2’ to A
  - C sent PKca and h(R1’) to B (and later R1’)
- If h() is cryptographically secure, C has to select:
  - R1’ without knowing anything about R2
  - R2’ without knowing anything about R1
- C should be able to choose R1’ and R2’ such that
  - $f(PKa,\underline{PKcb},R1,R2') = f(\underline{PKca},PKb,R1',R2)$
  - There exists a formal proof (omitted here) that if f() is cryptographically secure, attacker succeeds with probability 2^{-n}, where *n* is the length of f()’s output
- f() in Bluetooth is 6 digits → ≈ 2^{-20}

However: User might fail to compare 6 digit numbers correctly by mistake

Bluetooth 2.1+ Pairing: Passkey Entry

Requirements: Device A has a display, device B has a keypad, or both device have keypads, but no displays

Authenticated DH: 6-digit shared secret

Process (method 1):

- One device displays a randomly generated secret 6-digit number N
- User enters N into another device
- Then the devices authenticate their DH key using N

Process (method 2):

- User “generates” N and enters it into both devices
- Then the devices authenticate their DH key using N

Security:

- Only secure if N is really random, and it N is a nonce

Attack possibilities:

- Devices sometimes have fixed passwords
- USer generated predictabel passwords (like 00000)
- Attacker can interrupt protocol, user starts another run with the same N

Security Assumptions

- Passkey should be difficult to guess
- Passkey can be used only once
- Last step (user checks whether both devices displayed OK) is necessary
  - User has to confirm to A that B displayed “OK”, and vice versa
- Can we avoid user interaction in this protocol if the device that did not display “OK” stops communication?
  - No, this is not sufficient. Assume that B displayed “FAILED” and stopped, then A can still display “OK” due to the attack from the previous slide.

Bluetooth Passkey Entry

- Recommended but not mandatory procedure
  - Random generation of passkey for each pairing

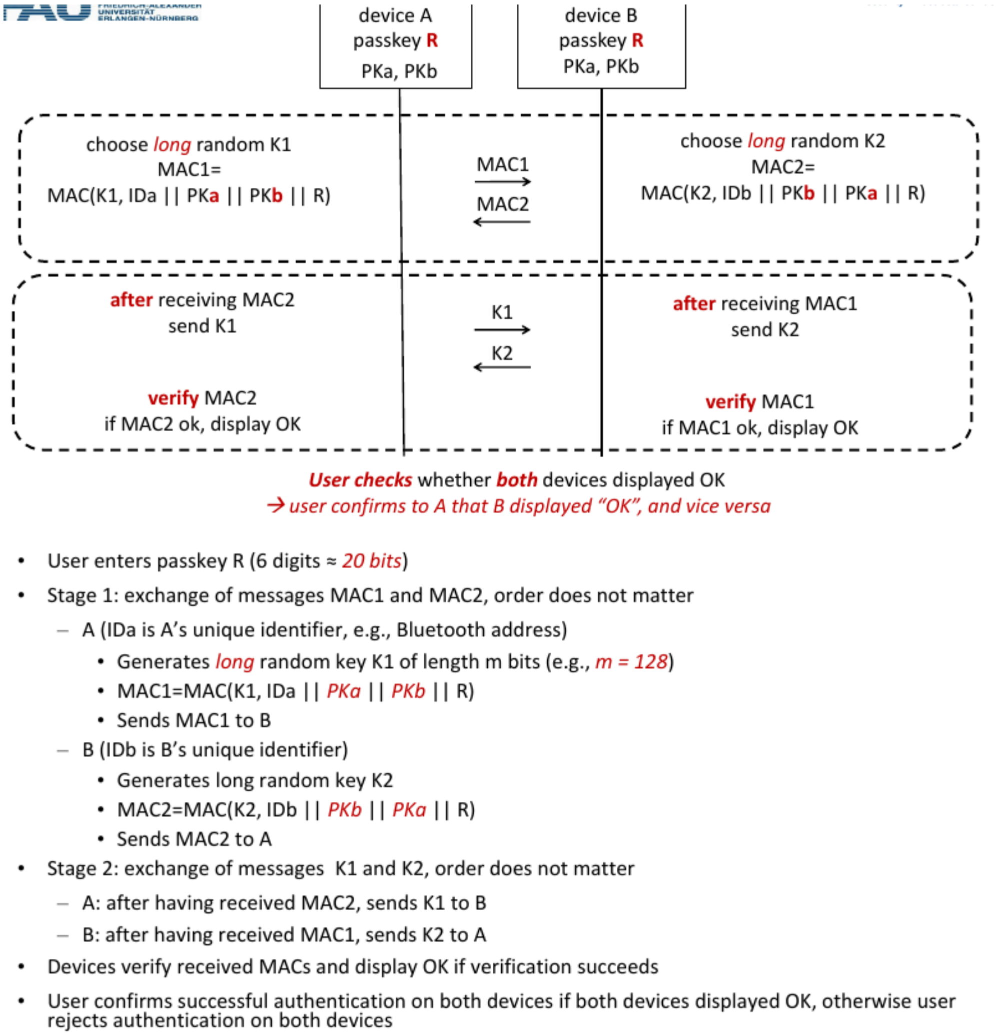
    - One device generates and displays passkey, user enters it into another device
- Vulnerabilities
    - n rounds for n-bit passkey, passkey reuse not prohibited
    - User-generated passkeys possible: will not be random
    - Passkey reuse attack: DoS on the pairing user will likely start again with the
    - same passkey
- General problem: Usage of a cryptographic protocol that violates protocol's security assumptions

Basic Passkey Authentication Protocol

Prerequisites and notations

- Devices A and B both have a keypad and a display
- MAC(K,x): message x encrypted under key K using Message Authentication Code
- || concatenation
- A and B exchanged DH public keys PKa and PKb

Authentification



device A
passkey **R**
PKa, PKb

device B
passkey **R**
PKa, PKb

choose *long* random K1
MAC1=
MAC(K1, IDa || PK**a** || PK**b** || R)

MAC1 →
MAC2 ←

choose *long* random K2
MAC2=
MAC(K2, IDb || PK**b** || PK**a** || R)

after receiving MAC2
send K1

K1 →
K2 ←

after receiving MAC1
send K2

verify MAC2
if MAC2 ok, display OK

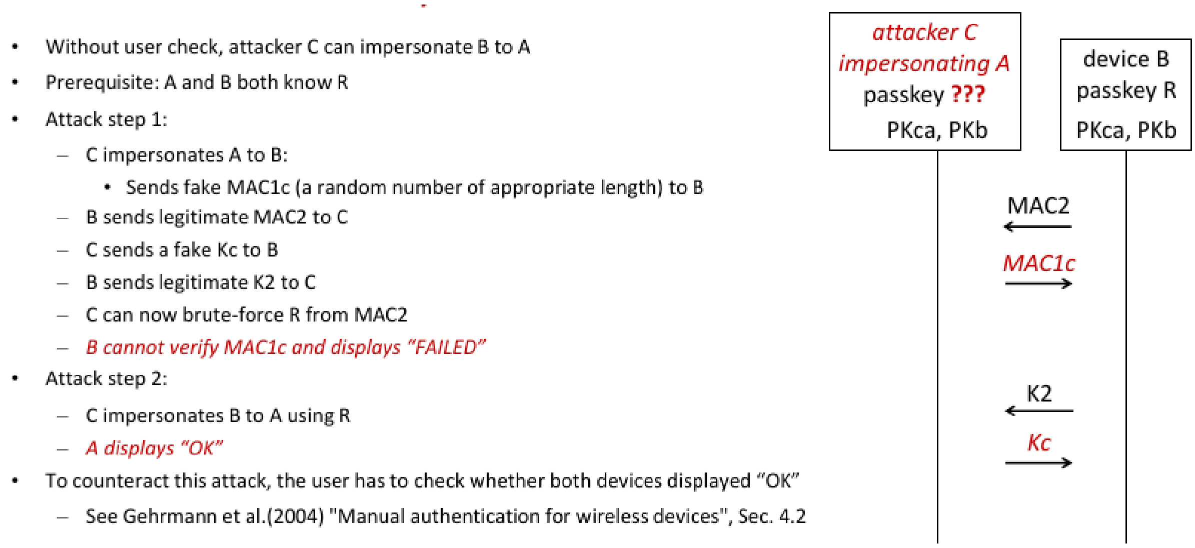
verify MAC1
if MAC1 ok, display OK

User checks whether ***both*** devices displayed OK
→ user confirms to A that B displayed "OK", and vice versa

- User enters passkey R (6 digits ≈ *20 bits*)
- Stage 1: exchange of messages MAC1 and MAC2, order does not matter
    - A (IDa is A's unique identifier, e.g., Bluetooth address)
        - Generates *long* random key K1 of length m bits (e.g., *m = 128*)
        - MAC1=MAC(K1, IDa || *PKa* || *PKb* || R)
        - Sends MAC1 to B
    - B (IDb is B's unique identifier)
        - Generates long random key K2
        - MAC2=MAC(K2, IDb || *PKb* || *PKa* || R)
        - Sends MAC2 to A
- Stage 2: exchange of messages K1 and K2, order does not matter
    - A: after having received MAC2, sends K1 to B
    - B: after having received MAC1, sends K2 to A
- Devices verify received MACs and display OK if verification succeeds
- User confirms successful authentication on both devices if both devices displayed OK, otherwise user rejects authentication on both devices

Necessity of User Check on Both Devices

Only one party can be impersonated, requires brute forcing, and either no user check or accidental user check (=user isn't looking)



- Without user check, attacker C can impersonate B to A
- Prerequisite: A and B both know R
- Attack step 1:
  - C impersonates A to B:
    - Sends fake MAC1c (a random number of appropriate length) to B
  - B sends legitimate MAC2 to C
  - C sends a fake Kc to B
  - B sends legitimate K2 to C
  - C can now brute-force R from MAC2
  - *B cannot verify MAC1c and displays "FAILED"*
- Attack step 2:
  - C impersonates B to A using R
  - *A displays "OK"*
- To counteract this attack, the user has to check whether both devices displayed "OK"
  - See Gehrmann et al.(2004) "Manual authentication for wireless devices", Sec. 4.2

Bluetooth 2.1+ Pairing: OOB (out-of-band)

Requirements: Uses NFC or other OOB channel + user interaction
DH executed over Bluetooth, DH public keys authenticated via OOB

Bluetooth 2.1+ Pairing: Just works

Intended to be used when all other options impossible, e.g., the devices do not have any input capabilities nor OOB

Unauthenticated Diffie-Hellman

Attack: Degradation to Just Works Attack

Man in the middle attack
Spoof device capabilities: Convince devies that communication partner does not have any input options, even if they have them

Attack: Method Confusion Attacks

MitM attack on BLE pairing
– One device executes Passkey Entry (PE) with the MitM device
– Another device executes Numeric Comparison (NC) with the MitM device
– Needs a jammer to suppress some messages
– User unable to spot the difference (user study)
Works for Bluetooth Classic with some restrictions

Unauthenticated DH in BLE

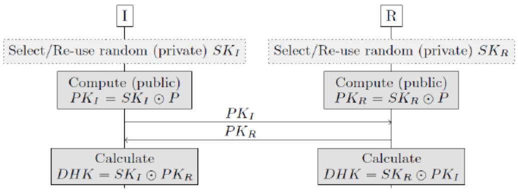

I
R
Select/Re-use random (private) SK_I
Select/Re-use random (private) SK_R
Compute (public) $PK_I = SK_I \odot P$
Compute (public) $PK_R = SK_R \odot P$
PK_I
PK_R
Calculate $DHK = SK_I \odot PK_R$
Calculate $DHK = SK_R \odot PK_I$

Fig. 1. ECDH Public Key Exchange and schematic calculation of shared secret.

- I = Initiator, R = Responder
- Elliptic Curve Diffie-Hellman (ECDH)-based key exchange on curve P-256
- ⊙ = scalar multiplication on the Elliptic Curve (EC) body

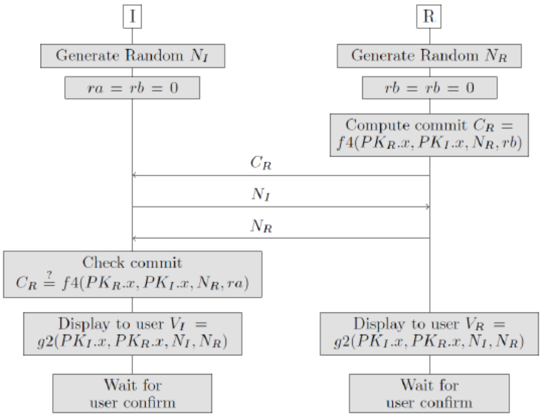

I
R
Generate Random N_I
Generate Random N_R
$ra = rb = 0$
$rb = rb = 0$
Compute commit $C_R = f4(PK_R.x, PK_I.x, N_R, rb)$
C_R
N_I
N_R
Check commit $C_R \overset{?}{=} f4(PK_R.x, PK_I.x, N_R, ra)$
Display to user $V_I = g2(PK_I.x, PK_R.x, N_I, N_R)$
Display to user $V_R = g2(PK_I.x, PK_R.x, N_I, N_R)$
Wait for user confirm
Wait for user confirm

Fig. 3. Numeric Comparison.

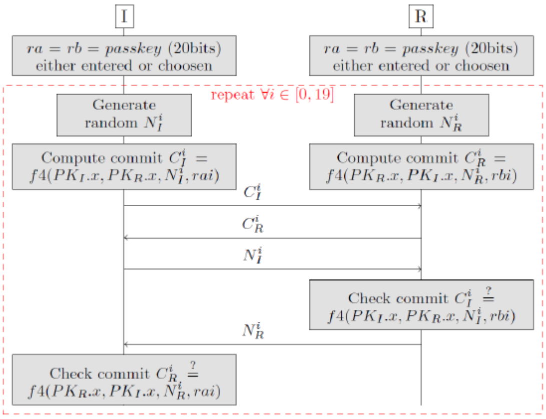

I
R
$ra = rb = passkey$ (20bits) either entered or choosen
$ra = rb = passkey$ (20bits) either entered or choosen
repeat $\forall i \in [0, 19]$
Generate random N_I^i
Generate random N_R^i
Compute commit $C_I^i = f4(PK_I.x, PK_R.x, N_I^i, rai)$
Compute commit $C_R^i = f4(PK_R.x, PK_I.x, N_R^i, rbi)$
C_I^i
C_R^i
N_I^i
Check commit $C_I^i \overset{?}{=} f4(PK_I.x, PK_R.x, N_I^i, rbi)$
N_R^i
Check commit $C_R^i \overset{?}{=} f4(PK_R.x, PK_I.x, N_R^i, rai)$

Fig. 4. Passkey Entry.

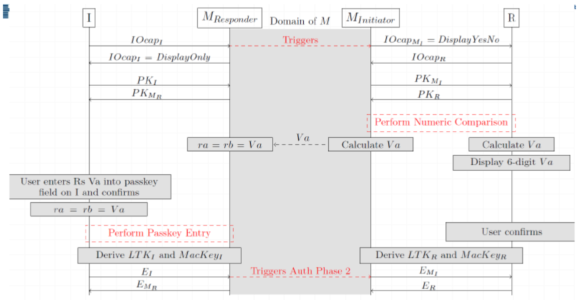


Fig. 6. Passkey on Numeric attack implementation. E_X is the confirmation value as calculated by device X.

Bluetooth Pairing: 6 Lessons Learned

- For ad hoc pairing, use public key crypto
- Be careful when providing less secure options
  - Degradation attacks
- Security assumptions of crypto protocols are important
- Do not rely on user-generated secrets
  - They will not be random
- Simplify or avoid user interaction
  - Reduces mistakes and misunderstanding
- Explicitly embed into the protocols which methods are used
  - Distinct values for PE and NC distinguishable by devices, not by users
  - Example: least significant bit of the passkey for PE = 0, of the hash for NC = 1

Need to know:

Bluetooth 2.1+ Pairing: Numeric Comparison
Basic Numeric Comparison
Bluetooth 2.1+ Pairing: Passkey Entry
Basic Passkey Authentication Protocol

- Prerequisities
- Authentication
- Necessity of User Check on Both Devices

Bluetooth 2.1+ Pairing: OOB (out-of-band)
Bluetooth 2.1+ Pairing: Just works
Attack: Degradation to Just Works Attack
Attack: Method Confusion Attacks
Unauthenticated DH in BLE (Mut zur Lücke?)
Bluetooth Pairing: 6 Lessons Learned

10. Device pairing

Device Pairing

- Key establishment between two devices
  - Devices do not share any common secrets (“Have never met before”)
- Security goals: secure channel
  - Authentication of both communication partners
    - No MitM (man-in-the-middle) attacks, no impersonation (Evil Twin)
  - Integrity + confidentiality of communication
- Idea: use out-of-band (OOB) channels to exchange authentication information
  - OOB channel is different from the primary wireless channel
  - Devices setup a connection over the primary channel, e.g., using unauthenticated DH
  - Use the information exchanged over the OOB channel to prevent MitM attacks

Possible OOB Channels

- User interaction (e.g., MANA protocols)
  - Enter PIN on both devices, push buttons, compare strings
- Additional channels (usually also involve user interaction)
  - Physical connection (USB, docking port)
  - Secondary transmission technology (infrared, NFC)
  - Audio (devices beep)
  - Visual (blinking LEDs, scanning of QR codes)
  - Objects (moving objects between devices)

WPS: WiFi Protected Setup + WPS Options

Goal: Usable and secure pairing of STA and AP

1. Push-Button: unauthenticated DH
   a. Push a button (sometimes a virtual one in the interface) on both STA and AP at (nearly) the same time
   b. Unauthenticated Diffie-Hellmann key exchange is activated
   c. No guarantee against malicious (or unintended) connections to wrong devices
2. PIN: DH authenticated by shared secret
   a. Enter PIN of the AP into the STA interface, or vice versa
3. NFC (Near Field Communication)
   a. Bring devices near to each other, such that STA can read network configuration from the RFID tag on the AP (optional and usually not implemented)

Attack: WPS AP-PIN Cracking

AP-PIN authentication:

- Enter 8-digit PIN of the AP into the STA interface
  - PIN is printed on AP same PIN is used for all STAs
- AP PIN: 8 digits
  - PIN brute forcing requires 10 8 (100 million) attempts
  -

Attack: There is a WPS design flaw (?)

- AP first sends confirmation (or rejection) for first 4 PIN digits
- Then AP sends confirmation (or rejection) for last 4 PIN digits
- Last PIN digit is a checksum for the other 7 digits
- Number of attempts reduced to 11000
- Implementation flaw
  - No back off after unsuccessful attempts or too short back offs
  - Try out PIN=0, if not success, then try out PIN=1 till success
  - Depending on WPS implementations, takes some minutes or some hours to crack the PIN by sending candidate PIN parts to the AP

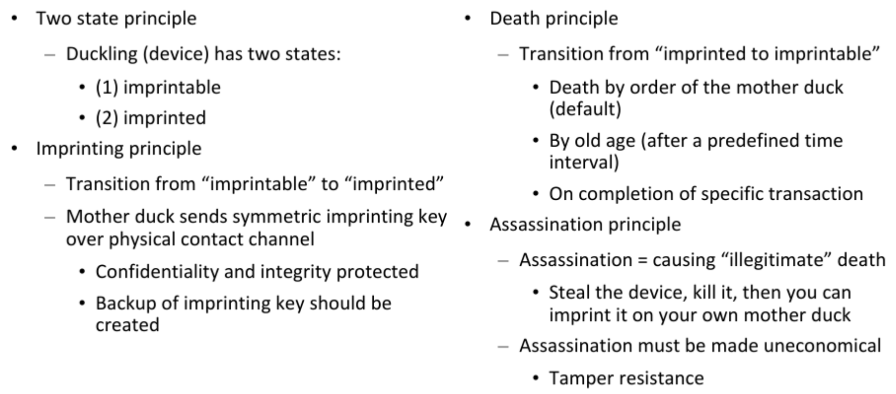
Need to know:

Device pairing (what is it, security goals, general idea and security)
Possible OOB Channels
MANA Protocols
WPS: WiFi Protected Setup + WPS Options
Attack: WPS AP-PIN Cracking

11: Device pairing in research and OOB channels

Resurrecting Duckling

Protect gadgets belonging to the same person or organization
RD is an access control policy
Secure Transient Association
– Secure: devices should obey commands of their users, and only of them
– Transient: resold or broken devices can be de-associated



- Two state principle
  - Duckling (device) has two states:
    - (1) imprintable
    - (2) imprinted
- Imprinting principle
  - Transition from “imprintable” to “imprinted”
  - Mother duck sends symmetric imprinting key over physical contact channel
    - Confidentiality and integrity protected
    - Backup of imprinting key should be created
- Death principle
  - Transition from “imprinted to imprintable”
    - Death by order of the mother duck (default)
    - By old age (after a predefined time interval)
    - On completion of specific transaction
- Assassination principle
  - Assassination = causing “illegitimate” death
    - Steal the device, kill it, then you can imprint it on your own mother duck
  - Assassination must be made uneconomical
    - Tamper resistance

Network-in-a-box (NiaB)

Goal: Enrollment into a secure WiFi network
Problem: Clients suualyl have to download and configure digitial certificates (to avoid evil twin attack), not practical for average user
Idea of NiaB: Use infrared communication as OOB channel

- Place STA (Laptop) near AP to use infrared OOB comms
- AP has a root certificate (public/private key pair)
- STA generates a public/private key pair
- OOB: Exchange hashes of public key
- In-Band: Run WPA-Enterprise enrollment based on TLS

NiaB Advantages:

- Very usable, fast, no user mistakes
- Secure (no evil twin)

NiaB Disadvantages:

- Line of sight needed
- Difficult to adjust devices is AP is placed in a bad spot (like the ceiling)

Seeing Is Believing

Device capabilities

- Device A can generate and display a bar code
- Device B can scan the displayed barcode
- Visual OOB channel

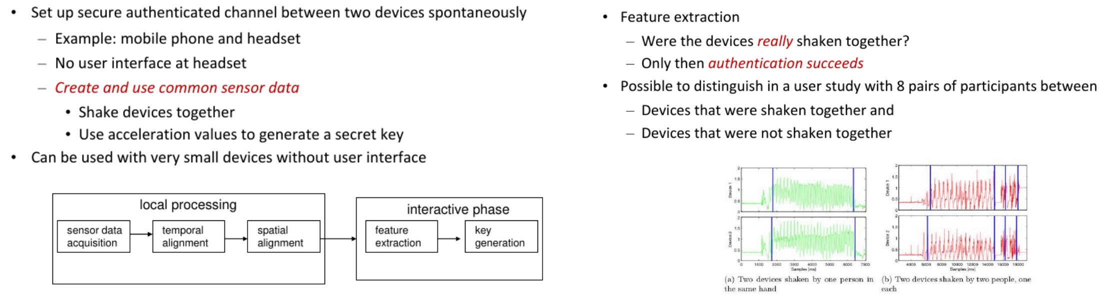
Authentication of a public key by integrity checking

- OOB transfer
- User is not required to enter or compare anything

How it works:

- Device A authenticates its public key to device B
  - Displays hash of the key embedded into the barcode
  - Sends the key to B over radio channel
- Device B
  - Scans the barcode, extracts the hash
  - Compares extracted hash with the hash of the previously received key

Shake Well Before Use



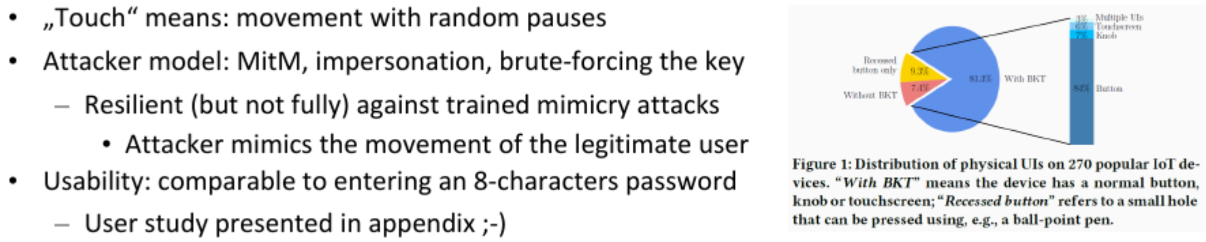
- Set up secure authenticated channel between two devices spontaneously
  - Example: mobile phone and headset
  - No user interface at headset
  - *Create and use common sensor data*
    - Shake devices together
    - Use acceleration values to generate a secret key
- Can be used with very small devices without user interface

local processing: sensor data acquisition → temporal alignment → spatial alignment → interactive phase: feature extraction → key generation

- Feature extraction
  - Were the devices *really* shaken together?
  - Only then *authentication succeeds*
- Possible to distinguish in a user study with 8 pairs of participants between
  - Devices that were shaken together and
  - Devices that were not shaken together

(a) Two devices shaken by one person in the same hand (b) Two devices shaken by two people, one each

T2Pair (Touch to Pair)



- „Touch“ means: movement with random pauses
- Attacker model: MitM, impersonation, brute-forcing the key
  - Resilient (but not fully) against trained mimicry attacks
    - Attacker mimics the movement of the legitimate user
- Usability: comparable to entering an 8-characters password
  - User study presented in appendix ;-)

Figure 1: Distribution of physical UIs on 270 popular IoT devices. "*With BKT*" means the device has a normal button, knob or touchscreen; "*Recessed button*" refers to a small hole that can be pressed using, e.g., a ball-point pen.

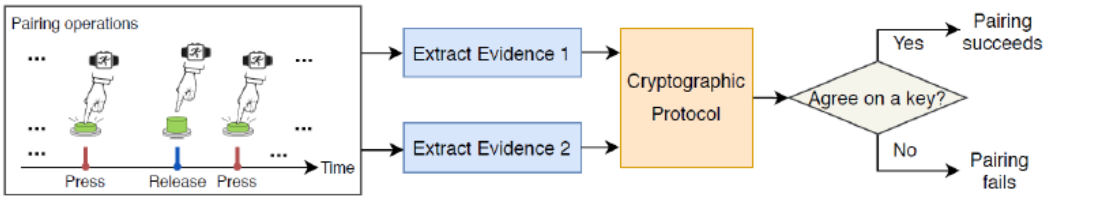
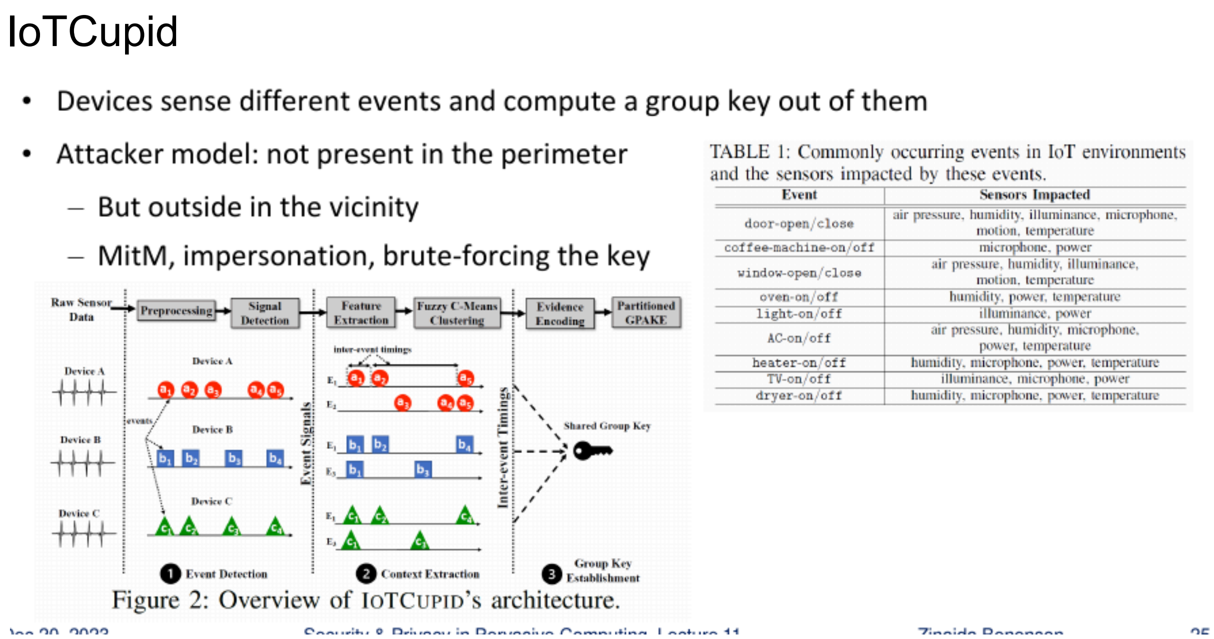


Figure 2: Architecture of T2Pair (a wristband as the helper and an IoT device with a button as an example)



IoTCupid

- Devices sense different events and compute a group key out of them
- Attacker model: not present in the perimeter
  - But outside in the vicinity
  - MitM, impersonation, brute-forcing the key

TABLE 1: Commonly occurring events in IoT environments and the sensors impacted by these events.

Event	Sensors Impacted
door-open/close	air pressure, humidity, illuminance, microphone, motion, temperature
coffee-machine-on/off	microphone, power
window-open/close	air pressure, humidity, illuminance, motion, temperature
oven-on/off	humidity, power, temperature
light-on/off	illuminance, power
AC-on/off	air pressure, humidity, microphone, power, temperature
heater-on/off	humidity, microphone, power, temperature
TV-on/off	illuminance, microphone, power
dryer-on/off	humidity, microphone, power, temperature

Figure 2: Overview of IoTCupid's architecture.

Object-based Pairing: Wanda Idea

Goal:

- Key establishment over the main wireless channel
- No previous secrets shared by devices

Wand idea:

- “Magic wand” imparts credentials to target devices
- – Wand uses two radio antennas
- – Uses radio technology already available on devices (WiFi, Bluetooth, ZigBee, ...)
- – Requires software changes on target devices, but no hardware changes
- Basic operations
- – Detect: Wand detects that it is in close proximity of the target device
- – Impart: After detection, Wand transmits secrets to the target device

Wand Detect

- (1) Both antennas measure RSSI
- (2) Wand computes the *difference* in signal strength between two antennas

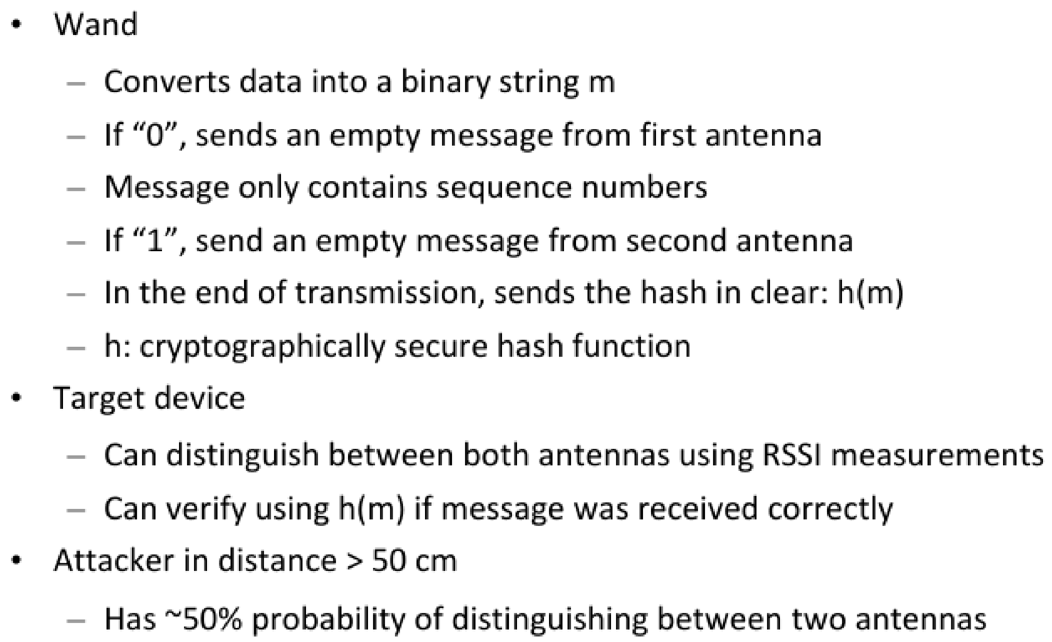
$$P_1 = P_0 - 10\alpha\ \log_{10}(\frac{d_1}{d_0}) + \chi_\sigma$$

$$P_2 = P_0 - 10\alpha\ \log_{10}(\frac{d_2}{d_0}) + \chi_\sigma$$

$$\begin{aligned} P_1 - P_2 &= P_0 - 10\alpha\ \log_{10}(\frac{d_1}{d_0}) + \chi_\sigma \\ &\quad - (P_0 - 10\alpha\ \log_{10}(\frac{d_2}{d_0}) + \chi_\sigma) \\ &= -10\alpha\ \log_{10}(\frac{d_1}{d_2}) \end{aligned}$$

- P1: power measured at distance d1
- P2: power measured at distance d2

Wand Impart



- Wand
  - Converts data into a binary string m
  - If “0”, sends an empty message from first antenna
  - Message only contains sequence numbers
  - If “1”, send an empty message from second antenna
  - In the end of transmission, sends the hash in clear: h(m)
  - h: cryptographically secure hash function
- Target device
  - Can distinguish between both antennas using RSSI measurements
  - Can verify using h(m) if message was received correctly
- Attacker in distance > 50 cm
  - Has ~50% probability of distinguishing between two antennas

Wanda Protocols
•••Common Key
– Impart network credentials on the new device
– Example: add new device to the home WiFi network
– Wand receives WiFi credentials through a USB connection to the AP
– User does not have to remember SSID and password
Unique Key
– Impart a common “fresh” key on two or more devices
Copy & Paste
– Receive information from one device and transfer it to another device

Lessons Learnt

••••For ad hoc pairing, use public key crypto
Use OOB
Security assumptions of crypto protocol are important
Simplify or avoid user interaction
– Reduces mistakes and misunderstanding

Need to know:

Resurrecting Duckling
Network-in-a-box (NiaB)
Seeing Is Believing
Shake Well Before Use
T2Pair (Touch to Pair)
IoTCupid

Object-based Pairing: Wanda Idea

Lessons Learnt

12. RFID

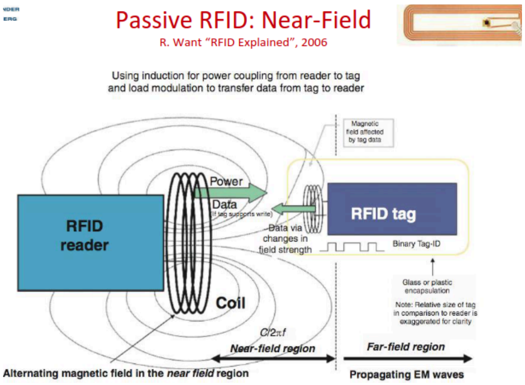
RFID (Radio-Frequency Identification)

=> Identification and tracking using radio waves
Architecture

- Integrated circuit (store and process data, modulate/demodulate signal)
- Antenna (receive power, transmit ID, extended functionality possible)

Passive RFID

- No battery
- With or without extended capabilities (e.g., crypto)

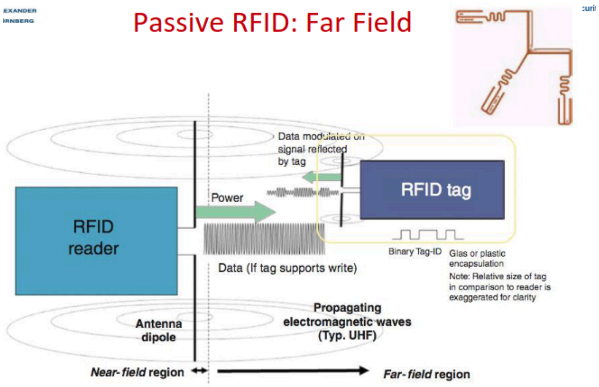


Passive RFID: Near-Field

R. Want "RFID Explained", 2006

Using induction for power coupling from reader to tag and load modulation to transfer data from tag to reader

RFID reader
Power
Data
(If tag supports write)
Data via changes in field strength
Magnetic field affected by tag data
RFID tag
Binary Tag-ID
Glass or plastic encapsulation
Note: Relative size of tag in comparison to reader is exaggerated for clarity
Coil
C/2πf
Near-field region
Far-field region
Alternating magnetic field in the near field region
Propagating EM waves



Passive RFID: Far Field

Data modulated on signal reflected by tag
RFID tag
Power
RFID reader
Data (If tag supports write)
Binary Tag-ID
Glas or plastic encapsulation
Note: Relative size of tag in comparison to reader is exaggerated for clarity
Antenna dipole
Propagating electromagnetic waves (Typ. UHF)
Near-field region
Far-field region

Battery Assisted Passive (BAP) RFID

- Battery
- Does not actively transmit information
  - Wait until a reader contacts them
  - Use battery for answering (larger communication Rage)

Active RFID

- Battery
- Microcontroller, radio, memory
- Active communication patterns possible
  - Transmit without being asked by a reader
  - Communication between tags

Passive RFID with Sensors

- No battery
- Changes in environment set an additional bit in memory
- Applications
  - Monitoring physical parameters
  - Tamper detection

- Detect radiation or bacteria

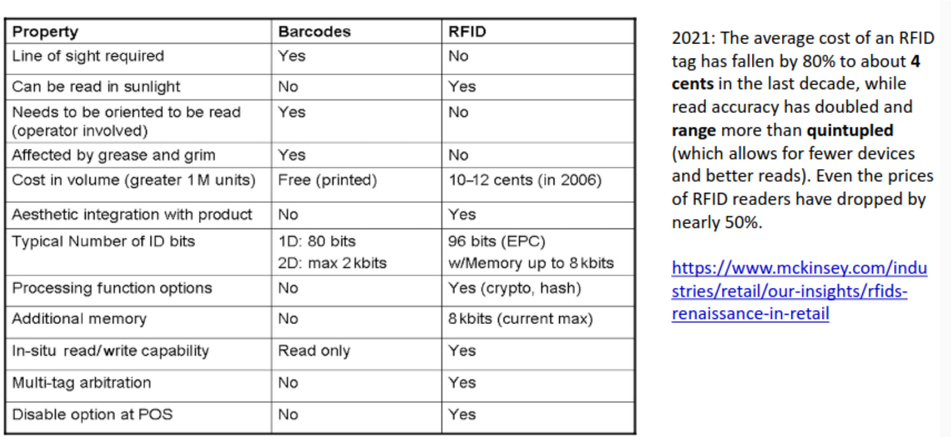
RFID Applications

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

Barcodes

- Optical machine-readable representation of data
  - 1D: Barcode
  - 2D: QR-Code
- Readers (scanners): Cheap and accurate

Barcodes vs. RFID



Property	Barcodes	RFID
Line of sight required	Yes	No
Can be read in sunlight	No	Yes
Needs to be oriented to be read (operator involved)	Yes	No
Affected by grease and grim	Yes	No
Cost in volume (greater 1 M units)	Free (printed)	10–12 cents (in 2006)
Aesthetic integration with product	No	Yes
Typical Number of ID bits	1D: 80 bits 2D: max 2 kbits	96 bits (EPC) w/Memory up to 8 kbits
Processing function options	No	Yes (crypto, hash)
Additional memory	No	8 kbits (current max)
In-situ read/write capability	Read only	Yes
Multi-tag arbitration	No	Yes
Disable option at POS	No	Yes

2021: The average cost of an RFID tag has fallen by 80% to about **4 cents** in the last decade, while read accuracy has doubled and **range** more than **quintupled** (which allows for fewer devices and better reads). Even the prices of RFID readers have dropped by nearly 50%.

https://www.mckinsey.com/industries/retail/our-insights/rfids-renaissance-in-retail

EPC (Electronic Product Code)

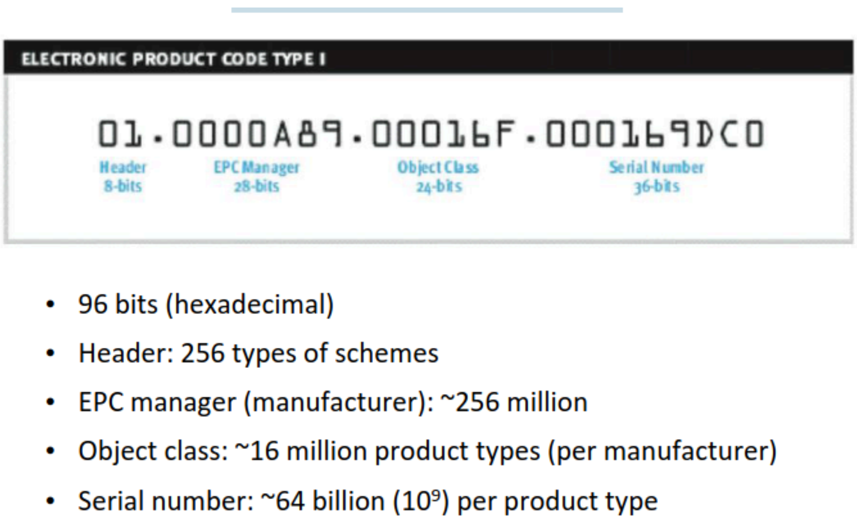
Unique identifier for every physician object anywhere in the world for all time

EPCglobal

- Defines the EPC structure as a freely available standard
- Defines decoding and encoding rules for efficiently storing EPCs on RFID tags and other data storages (like barcodes)
- Generation 1 (deprecated): 0 - Read only, 1 - write once, read many, field programmable
- Generation 2 - Classes:
  - 1: Passive, stores EPC ID, password, kill switch (tag self-destroys after transmitting individual Kill PIN)
  - 2: Passive Extended (memory, authenticated access control)
  - 3: semi passive (battery, sensors)

- active (battery, tag-to-tag communication, ad hoc networking)

EPCglobal Tag Format



- 96 bits (hexadecimal)
- Header: 256 types of schemes
- EPC manager (manufacturer): ~256 million
- Object class: ~16 million product types (per manufacturer)
- Serial number: ~64 billion (10^9) per product type

EPC: Reading Multiple Tags Issues

Example: Retail - all goods in a pallet or shopping car
Challenges:

- Packaging, Water, Plastic reflect and shield RF signal
- Poorly orientated antennas
- Collisions!!!

EPC RFID Tag Collision

RFID anti-collision protocols are necessary because multiple RFID tags may be present in the reader's interrogation field simultaneously, and the reader needs to communicate with each tag individually without interference.

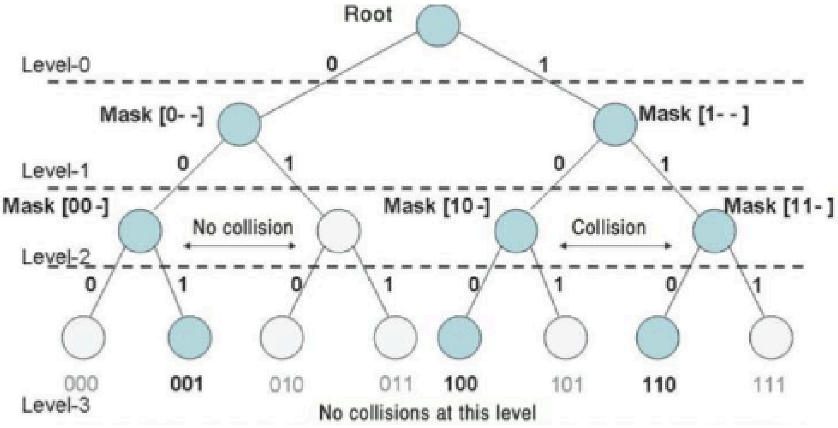
Query Tree Protocol (EPC Generation 1)

= a protocol against tag collision when multiple RFID chips are near the scanner

1. **Initialization**: The reader sends out an initial query command to the RFID tags within its range.
2. **Response from Tags**: Each RFID tag responds with its unique identifier (ID).
3. **Binary Tree Formation**: Based on the received IDs, the reader constructs a binary tree data structure. This tree structure helps in organizing the tags into groups for subsequent interrogation.
4. **Query Tree Execution**: The reader then starts a series of queries based on the binary tree structure. It selectively addresses groups of tags using a binary search algorithm, narrowing down the range of IDs in each query iteration until it identifies the individual tags.

5. **Tag Identification**: Each tag that receives a query responds accordingly, either by acknowledging its presence or by providing additional information as required.

R. Want: RFID Explained, Chapter 4

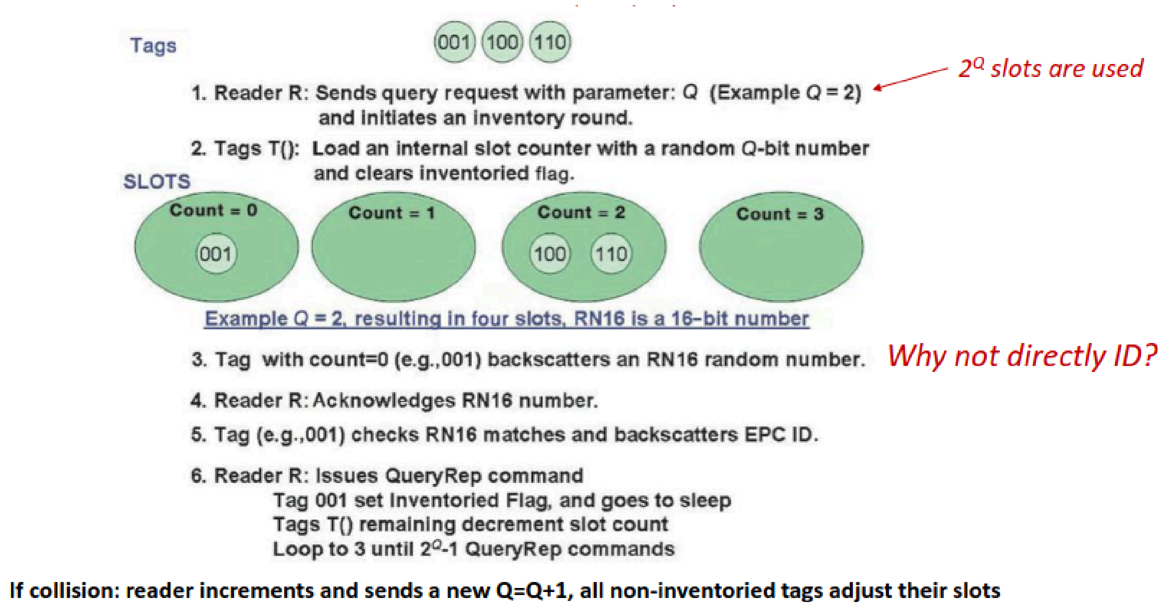


- Reader: broadcasts [0--]
- Tag 001: answers 0
- Reader: [00-]
- Tag 001: answers 1

- Reader: broadcasts [1--]
- Collision (100, 110)
- Reader: [10-]
- Tag 100: answers 0
- Reader: [11-]
- Tag 110: answers 0

Query Slot Protocol (EPC Generation 2)

1. **Initialization**: The RFID reader sends out a Query command to all tags within its range.
2. **Tag Response**: Each tag that receives the Query command calculates a random backoff time based on its EPC (Electronic Product Code) identifier. This helps prevent collisions between tags that might respond simultaneously.
3. **Slotting**: After the backoff time elapses, tags that are ready to respond transmit their data in response to the Query command. The reader receives these responses and acknowledges them.
4. **Tree Formation**: Based on the received responses, the reader organizes the tags into groups. This grouping is based on the binary tree structure, similar to the Query Tree Protocol in EPC Gen 1, but with some enhancements for efficiency.
5. **Query Execution**: The reader continues sending Query commands, using the information from the previous responses to refine its search and identify individual tags.
6. **Tag Identification**: Tags respond to the Query commands, providing their IDs or other relevant information. The reader acknowledges these responses and continues the process until all tags are identified



1. Reader R: Sends query request with parameter: Q (Example Q = 2) and initiates an inventory round.

2^Q slots are used

2. Tags T(): Load an internal slot counter with a random Q-bit number and clears inventoried flag.

Example Q = 2, resulting in four slots, RN16 is a 16-bit number

3. Tag with count=0 (e.g.,001) backscatters an RN16 random number.

Why not directly ID?

4. Reader R: Acknowledges RN16 number.

5. Tag (e.g.,001) checks RN16 matches and backscatters EPC ID.

6. Reader R: Issues QueryRep command
   Tag 001 set Inventoried Flag, and goes to sleep
   Tags T() remaining decrement slot count
   Loop to 3 until 2^Q-1 QueryRep commands

If collision: reader increments and sends a new Q=Q+1, all non-inventoried tags adjust their slots

Query Tree vs. Query Slot

- Query Tree
  - Advantages
    - No state in the tag
    - Deterministic (definitely finishes)
  - Disadvantages
    - Tag ID transmitted by the reader
    - Longer transmission range, less privacy
    - Cannot handle items with exactly the same RFID tag
      - Can happen, e.g., due to a hardware error

- Query Slot
  - Advantages
    - Only tag transmits its ID: shorter transmission range
    - Non-deterministic (can be very efficient)
  - Disadvantages
    - Memory and random number generator on tag

4 Security Threats in EPC

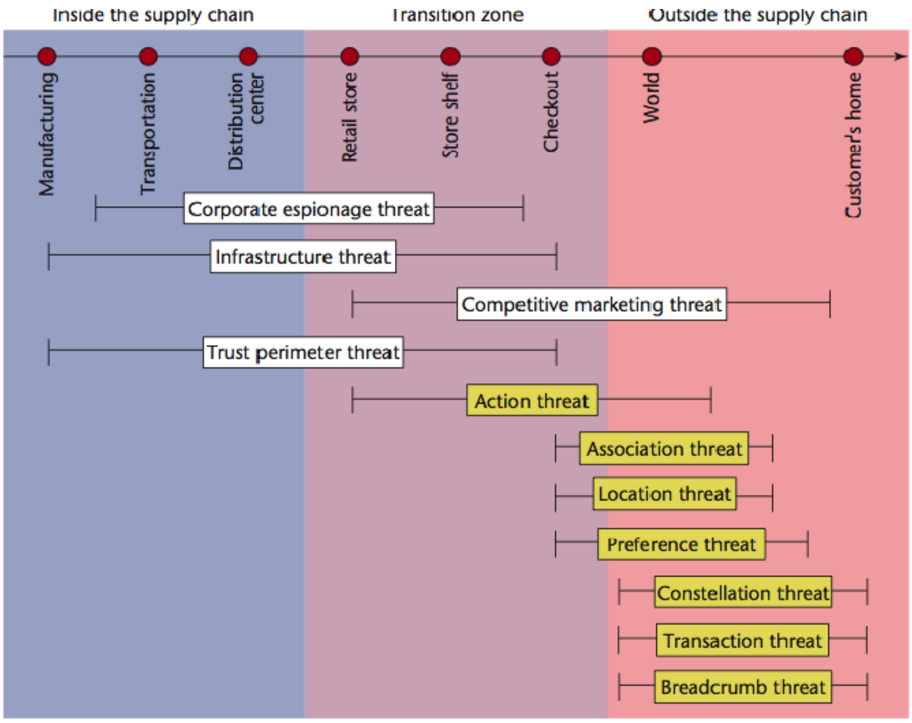
- Corporate espionage (read out confidential data about supply chain organization gathered through unauthorized readers)
- Infrastructure (Jamming Attacks: can disrupt RFIDs of an organization)
- Competitive marketing (unauthorized readers can gather and compile statistics in supermarkets, this is highly confidential marketing research data)
- RFID malware

RFID Malware

- Send invalid ID tag number to scanner
- And then: SQL injection in the RFID data base
- Or: Self replication - inject other RFID tags when sending requests (if the tags are writable), infect other databases

7 Privacy Threats in EPC

- Action: Monitor people's behavior through ther action with tagged objects (check if person tries to steal product from shelf)
- Association: Consumer is associated with the unique tag number of the product
- Location: Secretly track people through their associated tags using secret unauthorized readers
- Preference: Determine consumer preference without asking by scanning their purchases
- Constellation: People are associated with a "cloud" of multiple RFID tags that permanently identifies them (RFID chips in shoes, clothing, …)
- Transaction: Infer transaction between individuals through the movement of their "clouds" or RFID tags
- Breadcrumb: Association between person and ID tag is not broken even if product is discarded (what if the object is then used to commit a crime?)



Need to know:

RFID
Passive RFID
Battery Assisted Passive (BAP) RFID
Active RFID
Passive RFID with Sensors
RFID Applications
Barcodes
Barcodes vs. RFID
EPC (Electronic Product Code)

EPCglobal
EPCglobal Tag Format
EPC: Reading Multiple Tags Issues
EPC RFID Tag Collision
Query Tree Protocol (EPC Generation 1)
Query Slot Protocol (EPC Generation 2)
Query Tree vs. Query Slot
4 Security Threats in EPC
RFID Malware
7 Privacy Threats in EPC

13: More RFID

Tag Killing (Ensuring Privacy?)

- EPC tags have “Killing PIN”: tag self-destorys, tag is disabled forever
- Disadvantage: After-sales use impossible
- RFID after-sales use: Warranty or product return without receipt, reminders about medicine in cabinet, warning by washing machines that soem garments cannot be washed …
- Problem: Consumer might not check or be able to check if tag was killed (=> Metro Future Store issue)

Tag Covering (Ensuring Privacy?)

- Cover and uncover if needed
- Enables after-sale use
- Problem: Inconvvenient, tags ban be hidden by manufacturer, makes theft easier

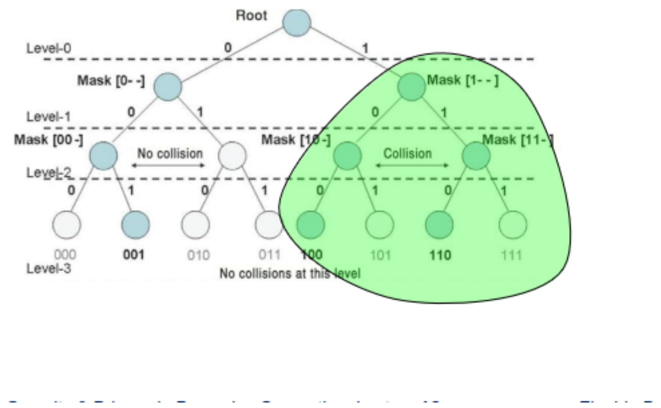
Blocker Tag (Ensuring Privacy?)

- Embedded in a bag or carried in pocket
- Readers sends signal => Blocker responds to block reading
- Problem:
  - Collisions, send random noise, …
  - Blocking Issue: Reader gets stopped (reader trying to resolve collisions forever)
  - Anti-theft protection difficult

Selective Blocker Tags

Blocker tag can be selective

- Privacy zones: Only certain ranges of serial numbers are blocked
- Example: block only part of the tree in the query tree protocol
- Zone mobility: Shops move items into privacy zone upon purchase
- Example: Tags with privacy bit: 1 = blocks identifiers, 0 = allow identifiers
- PIN for privacy bit flipping required to prevent theft



RFID Bill of Right

1. 1. The right to know if a product contains an RFID tag.
2. The right to have embedded RFID tags removed, deactivated, or destroyed when a product is purchased.
3. The right to first-class RFID alternatives. Consumers should not lose other rights (such as the right to return a product or travel on a particular road) if they decide to opt-out of RFID or exercise an RFID tag's kill feature.
4. The right to know what information is stored inside their RFID tags. If this information is incorrect, there must be a means to correct or amend it.
5. The right to know when, where, and why an RFID tag is being read

RFID: Lessons Learned (so far)

- Think before deploying pervasive computing technologies
  - Smallest computing units can have exceptionally big S&P implications
- RFIDs connect physical and electronic world: especially critical
  - Attacks can happen at all parts of the system
  - Analyze interfaces between the parts of the system
- Don't forget the backend (RFID malware)
- Implementing security & privacy without cryptography is difficult
- Watch out for implementation mistakes
  - "Kill" scandal in Metro Future Store
- Watch out for hidden embedded functionality
  - Hidden RFID tags in Metro Payback cards

Electronic Documents with RFID or other proximity cards

- Biometrics passport (reading distance 1m): Face Data, Fingerprints
- Security Issues: Cloning, changing, …
- Privacy Issue: Unauthorized reading

Passport Access Control Protocols

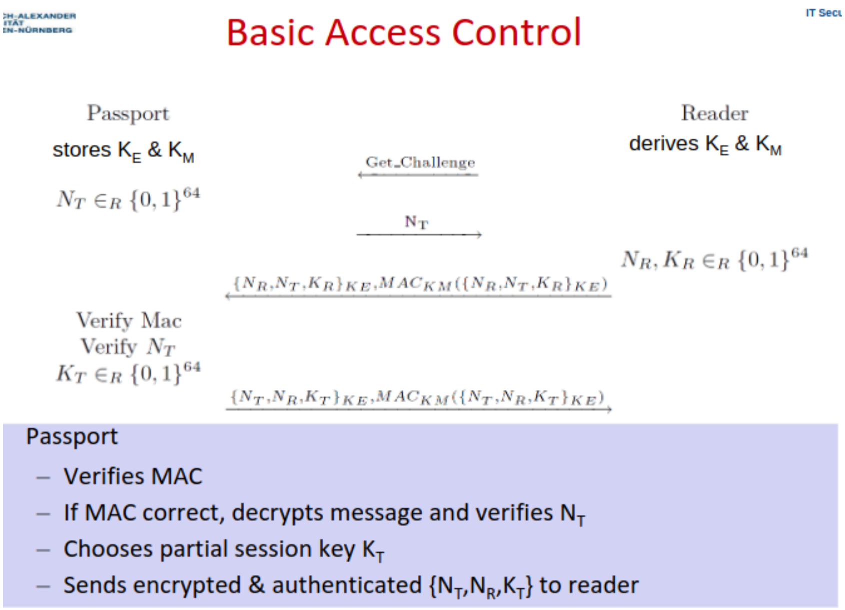
Basic Access Control (BAC):

- Mandatory, symmetric crypto for mutual authentication
- Key derived from machine-readable passport info
- Prevents scanning from third party

Active Authentication / Extended Access Control (E)

- Optional; Mandatory in EU
- Asymmetric Crypto
- Protects biometric info
- Information is digitally signed to prevent forgery
- Reader authenticates to the passport with a digital certificate

Basic Access Control



Basic Access Control

IT Secu

Passport

stores K_E & K_M

$N_T \in_R \{0,1\}^{64}$

Reader

derives K_E & K_M

$\xleftarrow{\text{Get\_Challenge}}$

$\xrightarrow{N_T}$

$N_R, K_R \in_R \{0,1\}^{64}$

$\xleftarrow{\{N_R,N_T,K_R\}_{KE}, MAC_{KM}(\{N_R,N_T,K_R\}_{KE})}$

Verify Mac

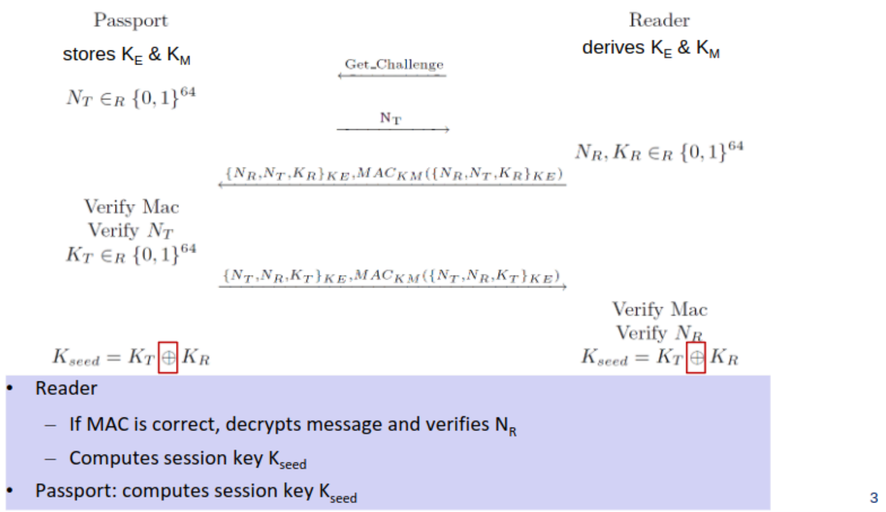
Verify N_T

$K_T \in_R \{0,1\}^{64}$

$\xrightarrow{\{N_T,N_R,K_T\}_{KE}, MAC_{KM}(\{N_T,N_R,K_T\}_{KE})}$

Passport

- Verifies MAC
- If MAC correct, decrypts message and verifies N_T
- Chooses partial session key K_T
- Sends encrypted & authenticated $\{N_T,N_R,K_T\}$ to reader



Passport

stores K_E & K_M

$N_T \in_R \{0,1\}^{64}$

Reader

derives K_E & K_M

$\xleftarrow{\text{Get\_Challenge}}$

$\xrightarrow{N_T}$

$N_R, K_R \in_R \{0,1\}^{64}$

$\xleftarrow{\{N_R,N_T,K_R\}_{KE}, MAC_{KM}(\{N_R,N_T,K_R\}_{KE})}$

Verify Mac

Verify N_T

$K_T \in_R \{0,1\}^{64}$

$\xrightarrow{\{N_T,N_R,K_T\}_{KE}, MAC_{KM}(\{N_T,N_R,K_T\}_{KE})}$

Verify Mac

Verify N_R

$K_{seed} = K_T \boxed{\oplus} K_R$

$K_{seed} = K_T \boxed{\oplus} K_R$

- Reader
  - If MAC is correct, decrypts message and verifies N_R
  - Computes session key K_{seed}
- Passport: computes session key K_{seed}

3

Example: Cloning E-Passport, 2006

Required physician access to the real card, could fool machine but nor human controller

Example: Cloning E-Passport, 2007

No need of physical access to the real passport, done by predicting BAC key if passport info (date of birth etc) is known

Example: Cloning and Changing E-Passport, 2008

- No physician access needed, worked with EAC (extended access control)

- Why did this work? Some readers accept self-signed certificates from anybody…

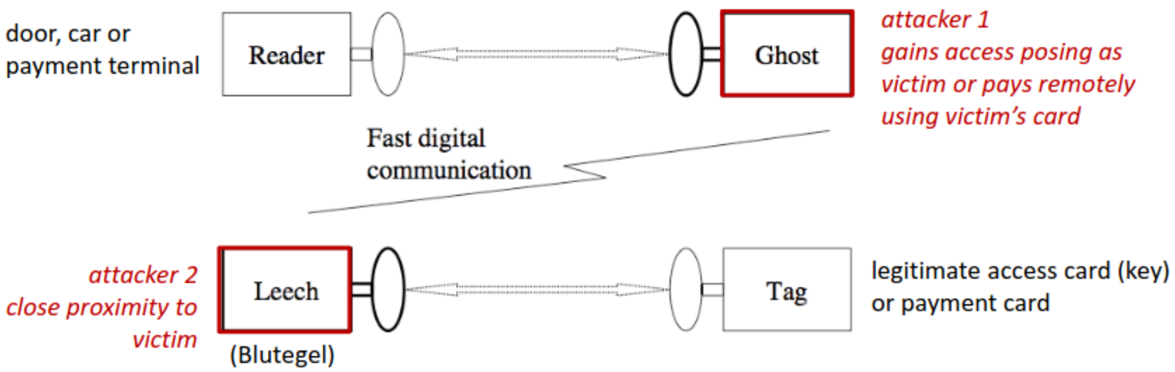
Tracking via Replay Attack

• Eavesdrop on a legitimate session between a passport and a reader
– Record the encrypted message from the reader that contains the passport's nonce
• To identify a particular passport, replay this message
– If replayed message is rejected because the MAC check failed, then this is not the same passport, as the MAC key is unique to each passport.
– If the message is rejected because of a failed nonce, then the MAC check succeeded, and therefore it is the same passport.
• A failed MAC check is reported noticeably sooner than a failed nonce

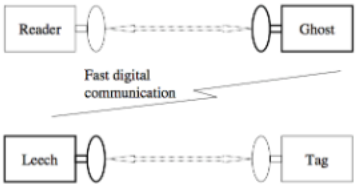
E Documents - Lessons learned

• Consider S&P consequences of the usage of a particular (wireless) technology
– Why are e-passports RFID-based?
• Crypto keys should be (pseudo)random
– Non-random cryptographic material can be predicted
• Side-channel attacks can exploit:
– Underspecified answers to commands
– Timing of protocol messages

Ghost-and-Leech Attacks on Access Control and Payment Systems


door, car or
payment terminal
Reader
Ghost
attacker 1
gains access posing as victim or pays remotely using victim's card
Fast digital communication
attacker 2
close proximity to victim
Leech
(Blutegel)
Tag
legitimate access card (key) or payment card

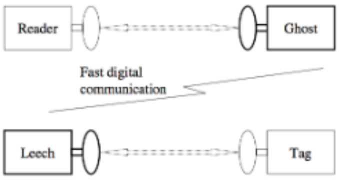
- Man-in-the-middle ("relay") attack on *RFID authentication and payment* protocols
  - Card authenticates to the reader using a *secure* challenge-response protocol
    - Secure = cannot be broken cryptographically
- Attack scenario
  - Ghost: *RFID card emulator* that runs the authentication protocol with the genuine reader
  - Leech: *Reader emulator* that runs the authentication protocol with the genuine RFID card
  - Protocol messages can be very quickly exchanged between the ghost and the leech
- Result: the fake card (ghost) passes the authentication using responses of the genuine card to the challenges
- Defense: *distance-bounding protocols*



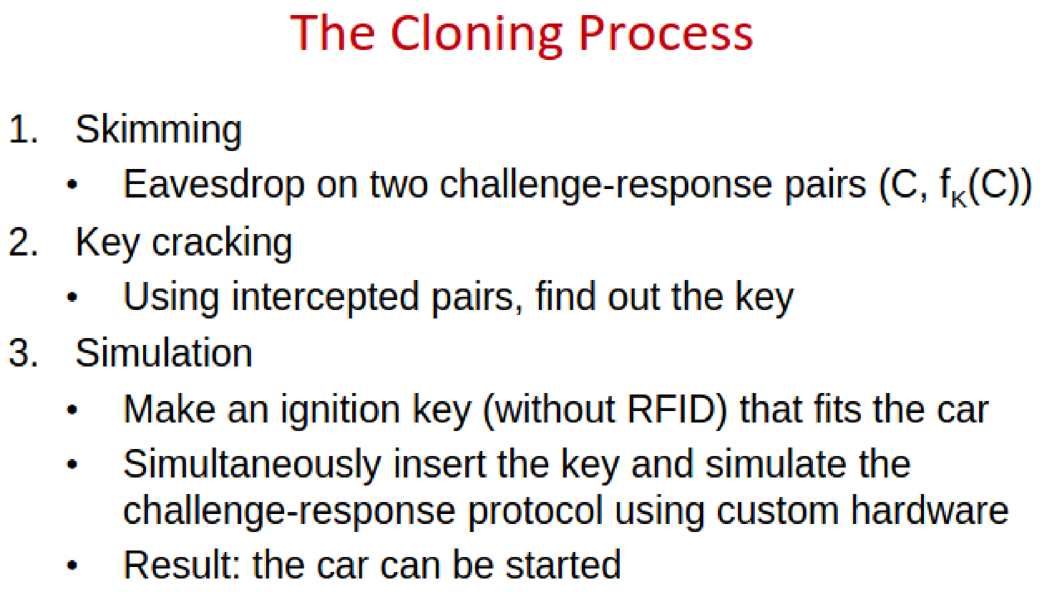
Distance-Bounding Protocols

Boureanu & Vaudenay. "Challenges in distance bounding." IEEE Security & Privacy 2015

- Prover (tag) & verifier (reader)
- Security goal
  - Make "ghost" impossible
  - Prover (tag) proves its proximity to the verifier (reader)
- Distance bounding principle
  - Several rounds (~100-200)
  - In each round, verifier sends a challenge and waits for the response to arrive within specified time bounds ("time-of-flight" for the allowed distance)
  - Serious implementation constraints: computing (prover) and processing (verifier) any response (even 1 bit) takes orders of magnitude longer than time-of-flight
    - Computing: microseconds, communication: nanoseconds



Skim Clone RFID Tags of car key



The Cloning Process

1. Skimming
   - Eavesdrop on two challenge-response pairs (C, $f_K(C)$)
2. Key cracking
   - Using intercepted pairs, find out the key
3. Simulation
   - Make an ignition key (without RFID) that fits the car
   - Simultaneously insert the key and simulate the challenge-response protocol using custom hardware
   - Result: the car can be started

Lessons Learned

• Consider S&P consequences of the usage of a particular (wireless) technology
– Why are e-passports RFID-based?
• Crypto keys should be (pseudo)random
– Non-random cryptographic material can be predicted
• Side-channel attacks can exploit:
– Underspecified answers to commands
– Timing of protocol messages
• Protect against relay attacks in challenge-response protocols
• Security through obscurity fails!

Need to know:

Tag Killing (Ensuring Privacy?)
Tag Covering (Ensuring Privacy?)
Blocker Tag (Ensuring Privacy?)
Selective Blocker Tags
RFID: Lessons Learned (so far)
Electronic Documents with RFID or other proximity cards
Passport Access Control Protocols
Basic Access Control (Passport ⇔ Reader)
Example: Cloning E-Passport, 2006
Example: Cloning E-Passport, 2007
Example: Cloning and Changing E-Passport, 2008
Tracking via Replay Attack
E Documents - Lessons learned

Ghost-and-Leech Attacks on Access Control and Payment Systems
Skim Clone RFID Tags of car key
Lessons Learned

Social Lectures

“I would suggest looking for the key terms of Mark Weiser's vision, difficulties with them today, and at the current and future developments:”

- Invisible / disappearing computing
- Calm technology
- Messiness of everyday life
- Seamless versus seamful
- Privacy / control

“Es ist möglich, dass dieses Jahr auch Fragen zu Smart City dazu kommen.”

- Cybernetics
    - Cybernetics is a central historical precursor and guiding idea behind many contemporary phenomena, especially smart cities, but also other forms of ubiquitous computing
        - Systems thinking
        - Governance as steering
        - Horizontal networking
        - Organicist thinking
            - Fiber optic cables and sensors – “nervous system”
            - Command & control center – “brain”
            - Fusion of city, machine & nature
        - Frictionless, apolitical understanding of process
- Smart City
-

“Explain the vision behind the term “ubiquitous computing” as it was discussed by Mark Weise. What is the role of “calm technology”?”

"Ubiquitous computing," as discussed by Mark Weiser, refers to the idea of integrating computing technology seamlessly into everyday objects and environments, to the point where they become invisible to the user.

Calm technology refers to technology that remains in the background of users' attention, providing information or assistance when needed without causing distractions or disruptions.

3 Device Types: Tabs (organizer, diary, …), Pad (“scrap computers), Board (blackboard).
Capabilities: Wireless networking, cooperation, context-aware (location)

Exam Questions/Tasks

Achtung: Die Antworten sind nicht immer unbedingt optimal. Bitte versuche den Scheiß auch selbst zu beantworten, um sicher zu gehen dass ihr volle Punktzahl in der Klausur bekommt.

Introduction

Define the three security goals + authentication (4P)

Confidentiality: Protect against unauthorized (reading) access
Integrity: Protect from unauthorized (writing) changes
Availability: Make data always available on request by an authorized entity
Authentification: Distinguish between authorized and unauthorized entities

Define information privacy

The claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.

Explain the term “Internet of Things” (2P)

Things of our daily live contain computers and are connected via e.g. the Internet or other networks to exchange data and information. Additionally, sensors of the devices deliver information about the physical environment. Actors (actuators) respond to the physical environment.

What are (six) special characteristics of security and privacy in IoT in comparison to the traditional Internet?

1. Devices are pervasive, everywhere and often invisible
2. Bodily and territorial privacy are important topic in IoT S&P.
3. Different/new quality and quantity of data
4. Profiling: habits, emotions (detected and processed via cameras and audio)
5. Devices observe and interact with the physician environment
6. Unprecedented data collection scale & attack surface

Explain Greenfield’s principle X (e.g., “be deniable”) and provide an example of a system that satisfies / does not satisfy it.

Be deniable = Opt-out always possible.

This is not guaranteed for the Windows IoT OS and the normal Windows OS, as they always collect telemetry data and other data when the user interacts with the system. It is not possible to opt-out of that data collection, even when some of them can endanger the privacy of the user.

Consider system X (e.g., GSM). Perform a security and privacy assessment of X.

System Description (Actors, Assets and Data):

- Telecom. Companies
- Customers
- Manufacturers
- State
- Assets&Data: Devices and their content, communication metadata, communication content, billing, infrastructure

S&P Goals:

- Confidentiality, Integrity and Availability of all assets
- Non-repudiation of calls, privacy of subscribers

Other Goals:

- Fast Connection Establishment, calls to devices in different networks (fixed networks or networks of other companies), voice calls, text messages, transparent billing system, online banking
- Relatively Simple Authentication and Attachment of Mobile Stations to Network

Attackers/Threat Models:

- Criminals, terrorists, (foreign) State(s), rival companies
- Threats: Eavesdropping, Billing Fraud, Voice Call/Test Message manipulation, loss/theft of devices, tracking of customers/mobile stations
- Attackers don’t know parts of the security and encryption process (Security by Obscurity)

Trade-Offs of Goals:

- Security vs.State: GSM is designed insecurely so that law enforcement can track and eavesdrop on criminals
- Cheaper and Easier System vs. Security: GSM uses Security by Obscurity via the A3/A8 algorithm, which was cheaper, but less secure, and eventually lead to massive costs to fix security issues due to the possibility of breaking the encryption
- Simplicity vs Security: GSM Connection Establishment and Authentication is simple (less steps between Mobile Station and GSM network), but less secure

How would you secure such a system X?

- No “security by obscurity” in cryptography
- Provide mutual authentication
- Provide security (confidentiality + integrity) in every part of the system
- Crypto algorithms should be easy to change
- Consider future technology developments and adequately powerful attackers in
- threat analysis
- Provide transparent technology development processes

Explain why your security measures meet your security and privacy goals for system X.

- No “security by obscurity”: Ensures CIA goals by ensuring security even if it is published hows encryption and security system works:
- Provide mutual authentication: Avoid billing fraud attack to ensure that
- Provide security (confidentiality + integrity) in every part of the system: Avoid compromising CIA and to allow exploits/atztacks
- Crypto algorithms should be easy to change: To avoid extenbsive costs when crypto algorithms get insecure in future developoment

Explain the IoT design principle “be self-disclosing”. Give an example of an IoT system/device that does not achieve that goal, and briefly explain why. (3P)

Ownership, usage and capabilities must be easy to find out.
Modern cars: It is not specified which data is collected, stored and processed on the servers of the manufacturers and 3rd party services. Sometimes even the manufacturers are not sure which data is stored (capabilities are not easy to find out)

Explain the vision behind the term “ubiquitous computing” as it was discussed by Mark Weise. What is the role of “calm technology”?

Ubiquitous Computing:

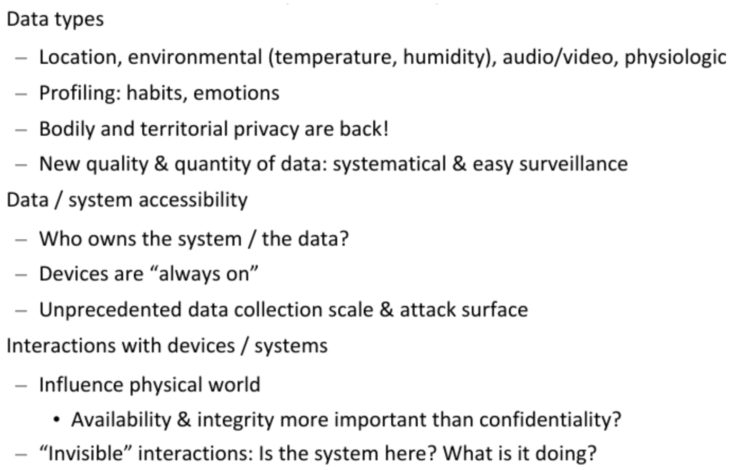
- “technologies that disappear”
- “ubiquitous invisible computing”
- “computers invisible to common awareness”
- “computers informally enhancing every room”
- “computers entering invisibly into people’s lives”
- “machines that fit the human environment instead of forcing humans to enter theirs”
- 3 Device Types: Tabs (organizer, diary, …), Pad (“scrap computers), Active-Badges (+ Board (blackboard) in SMart Cities).
- Capabilities: Wireless networking, cooperation, context-aware (location)
- Calm Technology: Technology recedes intot he background of user’s attention

Outline three special characteristics of security and privacy in IoT systems in comparison to security and privacy of the traditional Internet (1.5P)

??? Ist damit dass hier gemeint?
Meine Antwort:
Data: Other data types (environmental data), new quality and quantity of data (systematic and easy surveillance)
System Accessibility: IoT devices are “always on”, unprecedented data collection scale & attack surface

Interactions between systems: Influence and observe physician world, devices are connected and invisible

Folie:



Data types

- Location, environmental (temperature, humidity), audio/video, physiologic
- Profiling: habits, emotions
- Bodily and territorial privacy are back!
- New quality & quantity of data: systematical & easy surveillance

Data / system accessibility

- Who owns the system / the data?
- Devices are “always on”
- Unprecedented data collection scale & attack surface

Interactions with devices / systems

- Influence physical world
  - Availability & integrity more important than confidentiality?
- “Invisible” interactions: Is the system here? What is it doing?

Cellular Communication

How does GSM authentication work? (8 P)

Include:

- needed system components;
- cryptographic secrets, cryptographic algorithms and other essential information;
- content of exchanged messages

GSM (Global System for Mobile Communications) (2P)

- segmentation of the area into cells (100m-35km big, radio areas overlap)
- Different frequencies used in neighbouring cells
- Handover of the connection to the next cell if mobile user is traveling

Authentification: (2P)

- Components: SIM Card (IMSI; TMSI)
- Preshared key K_{SIM} in Simcard (128bit)
- symmetric algorithm

Authentification Process (4P):

1. Phone (MS) sends TMSI and its encryption capabilities to Base Station Controller (BSC)
2. BSC asks home location register (HLR) for auth. data of the IMSI (Communication between BSC and HLR are forwarded by the Mobile Service Switching Center (MSC)
3. HLR gets K_{SIM} from its data and generates data for challenge response:
    a. Choose random number RAND
    b. Encrypt RAND with K_{SIM} using A3/A8
    c. Output is SRES (A3: Response to challenge) and K_C (A8: symmetric key to encrypt data)
4. HLR sends RAND, SRES and K_c back to BSC
5. BSC sends RAND and used encryption to mobile devise (MS)
6. Device calculates challenge with K_{SIM} => SRES1, K_C1
7. Devices(MS) sends SRES1 to BSC
8. BSC verifies if result from HLR and MS are the same
    a. if yes: auth successful
    b. if not: terminate auth as unsuccessful

Which vulnerabilities does GSM have? Which of these vulnerabilities were corrected in UMTS, and how?

Free Call Attack: Backend Eavesdropping: Attack possible to due enencrypted microwav link between BSC and MSC, this was fixed in UMTS

Security by Obscurity: A3/A8 algorithms was cryptographically insecure and hidden, UMTS

Eavesdropping via IMSI Catcher: Insecure Communication Encryption algorithm A5/1, was replaced by A5/3 and A5/4 in UMTS, Mutual AUthentication was added to UMTS as well

Additionally Better Integrity Checks were added in UMTS (which were vulnerable though

Eavesdropping the A5 key in microwave link: Was fixed by encrypting microwave link im UMTS

Sim Card Cloning: GSM allowed sim card cloning due to using the bad A3/A8 algorithm, allowing a cryptographic attack to clone the sim card

Which possibilities for eavesdropping of GSM communication do you know?

2 ways:

Eavesdropping communication via IMSI-Catcher and decrypting it due to bad encryption algorithms

Eavesdropping the A5 key to decrypt communication

What is an IMSI-catcher? How does a IMSI-catcher work? (2P)

- An IMSI catcher is a portable fake (!!!) base station
- Mobile stations (phones) always connect to base station with strongest signal
- => bring IMSI catcher close to target mobile station
- It can require the MS/phone to send its real IMSI instead of its TMSI
- Allows tracking and eavesdropping

Outline how an IMSI-catcher can be used to eavesdrop on a phone call in GSM. (2P)

- IMSI catcher impersonates the BSC for the MS to get data from MS
- Impersonates MS in front of real network
- set encryption capabilities of fake MS to A5/0 (no encryption => eavesdrop)
- if A5/0 is refused, set to A5/1, record traffic, and break A5/1

Explain why eavesdropping via IMSI-catcher does not work in UMTS.(2P)

In UMTS, the MS checks the integrity of security algorithms, and the encryption algorithms A5/3 and A5/4 are stronger than A5/1 (2P).

Which attacks do IMSI-catchers enable?

Eavesdropping off Communication and Tracking of Mobile Stations.

Which possibilities to clone SIM cards do you know? Which attacks does SIM card cloning enable?

Option 1: Extract the key K_{SIM} from the smart card…but SIM Cards are smart cards, and smart cars are tamper proof => Secret hard to extract

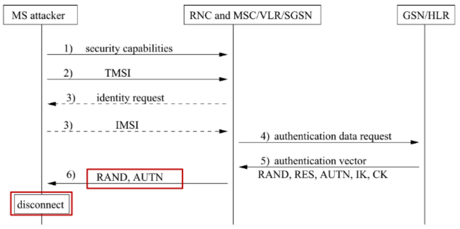
Option 2: Cryptographic Attack

- Option 2.1: Get physical access to SIM card, break the crypto algorithms A3/A8
  - Submit many RAND queries, analyze SRES response
  - If A3/A8 is cryptographically secure against chosen plaintext attacks, this attack should be infeasible
    - A3/A8 is NOT cryptographically secure
- Option 2.2: Over-the-air cloning (OTA): find out K_{SIM} from communication
  - Same as above, but with a more restricted number of RAND; SRES pairs
  - Eavesdrop on (RAND, SRES) paris, break the crypto algorithms A3/A8

Outline how the GSM downgrading attack on UMTS works.

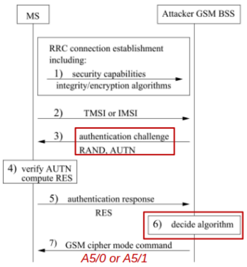
• Assumption: MS implements both GSM and UMTS
• IMSI-catcher impersonates MS in UMTS mode to RNC
• RNC sends to MS: “fresh” authentication token AUTN
– IMSI-catcher breaks up the connection, saves AUTN

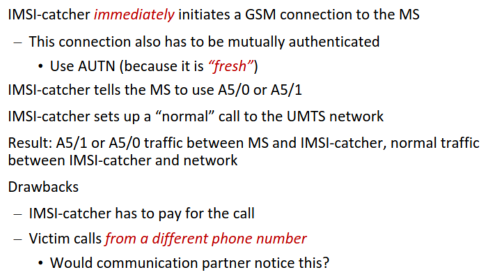
Step 1: use TIMSI / IMSI to get valid AUTN



Now we have a valid AUTN…

Step 2:
- Authenticate to target MS using AUTN
- *What next?*


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



IMSI-catcher *immediately* initiates a GSM connection to the MS

- This connection also has to be mutually authenticated
  - Use AUTN (because it is *"fresh"*)

IMSI-catcher tells the MS to use A5/0 or A5/1

IMSI-catcher sets up a "normal" call to the UMTS network

Result: A5/1 or A5/0 traffic between MS and IMSI-catcher, normal traffic between IMSI-catcher and network

Drawbacks

- IMSI-catcher has to pay for the call
- Victim calls *from a different phone number*
  - Would communication partner notice this?

How can users be tracked in cellular communication? Which countermeasures against user tracking exist?

IMSI Catcher: Pinpoint targeted person with precision up to several meters
TMSI: TMSI can be switched off by BS, TMSI is rarely changed
IMEI: BS can ask mobille phone to transmit their IMEI (feature against phone theft)

Why should location data be protected? What makes it important?

Location data can be used to determine which person is when and where. It allows tracking and prediction about where a person will be or which person is at a location. it is a privacy issue.

What is SS7? Which attacks are possible using SS7? How do these attacks work?

SS7 is a protocol used for communication between telecom operators

Attacks require buying access to SS7

Rogue SS7 Operator: Buy access to SS7, allows 4 attacks

- Locate & track:
  - Ask HLR for IMSI of phone number
  - Ask HLR which MSC is this IMSI
  - Ask MSC: At which BTS is this IMSI
- Eavesdrop: "Please send authentication and encryption keys for this TMSI"
- Manipulate: "This IMSI wants its calls/SMS forwarded to my network"
- Steal Money from Online Banking
  - Attacker gains control over victim's online banking account (via phising or malware)
  - Attacker looks up victim's phone number (Online banking with OTP(one-time-password) via SMS)
  - Sets up SMS redirect
  - Logs into online banking
  - Start translation, use SMS OTP to verify transaction => Money!

Which attacks on security and privacy we considered for LTE / 5G? How do they work? (in a nutshell)

LTE - Impersonation/Billing Fraud: Impersonate UE by man-in-the-middle attrack to trick Commercial Network to use unencrypted communication, and by capturing and ascending authentication challenge response from the UE that should be impersonated

LTE- aLTEr Attack: Manipulate Stream Cipher Data integrity, allows DNS redirects and manipulation of data
5G - ReVoLTE Attack: Abuse reuse of keystreams in voice calls, capture traffic between two MS (Alice and Bob), then immediately call Alice, use data to figure out keystream and use to decrypt captured call between ALice and bob

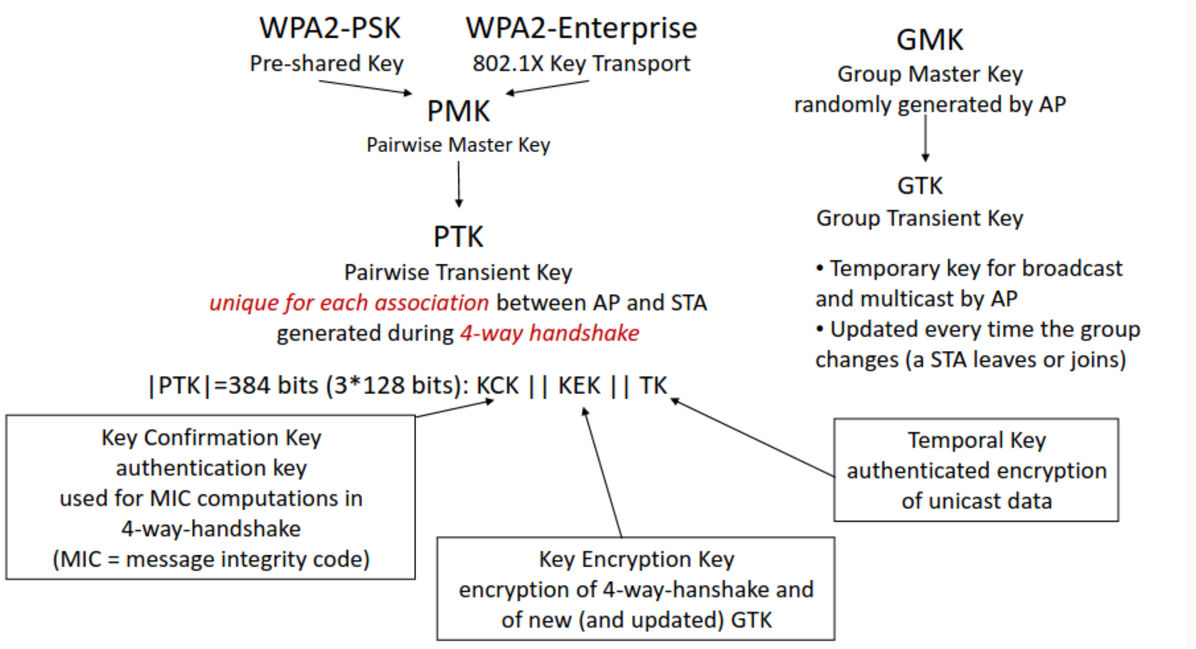
What are most important lessons learned about pervasive security and privacy on the example of cellular communication?

12 Lessons:

- No “security by obscurity” in cryptography
- Provide mutual authentication
- Provide security (confidentiality + integrity) in every part of the system
- Crypto algorithms should be easy to change
- Consider future technology developments and adequately powerful attackers in threat analysis
- Provide transparent technology development processes
- Implementing backward compatibility can leave old vulnerabilities exploitable
- Management of pseudonyms should be specified & implemented very carefully
- Management of non-secure modes should be specified & implemented very carefully
- Specifications should warn clearly about possibilities of insecure implementations
- Availability, reliability, performance measures as well as new applications can have unforeseen security & privacy consequences
- Specifications should precisely define security goals and threat model

Wi-Fi

Outline the WPA2 key hierarchy. Describe for which purpose each key is used. (6P)



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)

Briefly outline the difference in PMK generation between WPA2-PSK and WPA2- Enterprise. (2P)

WPA2-PSK: PMK is based on a passphrase (1P)
WPA2-Enterprise: PMK is generated during the authentication (1P)

Describe an attack that is possible in WPA2-PSK, but not in WPA2-Enterprise due to the above difference in PMK generation.(2P)

In WPA2-PSK, the PMK is based on

- a password and
- salt (=network name)

This allows a key cracking attack. Capture the handshake, and try out passwords using a dictionary/tables with the most popular network names and passwords

In Enterprise the PMK isn't based on a password, attack is not possible.

What are the most important lessons learned about pervasive security and privacy on the example of WEP?

1. Don't use master keys directly to encrypt communication

- Integrate key management into the system
- Key distribution and update

2. When using cryptographic algorithms, always ask experienced cryptographers how to do this properly:

- WEP uses RC4 in an inappropriate way
- Be extremely careful when using stream ciphers
- Think about reuse of initialization vectors and other components that should be used only once

3. Consider replay attacks

4. Always use cryptographically secure integrity protection

- Shared secret key: MAC = Message Authentication Code
- Public key crypto: digital signatures

How do security measures „hidden network“ and „MAC address filtering“ work in Wi-Fi? Which security goals do they have? How can these security goals be attacked?

Goal: Prevent alien STAs from joining the network

Hidden Network: APs do not send beacon data, and waits for STAs to actively search for AP by asking for a particular ESSID
Attack: Sniff till some STA sends probe request, replay probe request

MAC Address Filtering: AP only answers/accepts probes or authentication requests from STAs with know MAC addresses
Attack: Sniff allowed MAC addresses, change/spoof your MAC address to sniffed allowed MAC address

Both “security” measures are bad because they rely on security by obscurity, and can easily be defeated

How does Evil Twin attack work in public Wi-Fi networks? Which further attacks does it enable?

Setup the Attack:

- Attacker sets up AP, this AP impersonates the legitimate AP
- Broadcasts beacon signal with the ESSID of the legitimate AP
- => Evil Twin

Attack:

- Device connect to the AP with the strongest signal
- Evil Twin beacon can be made the strongest (by going near the victim’s device)

What can be done with an Evil Twin:

- Send fake login pages to user device, steal passwords etc.
- Forward Internet traffic (=sniff all clear text information, including login info)

- Phishing: Use DNS spoofing to redirect to evil servers (redirect traffic from My Bank to evil fake version of the Bank)

Why can Wi-Fi traffic be eavesdropped on much easier than traffic in cellular networks?

Weak Passwords for Wi-Fi Networks allow to crack encryption between STAs and APs, and ot eavesdrop their communication

Wi-Fi relies on passwords, while cellular communication relies on SIM Cards with unique keys and identifiers

For which purpose are IV (initialization vectors) used by WEP?

It is used as an offset for the PRNG Function (Pseudo-Random Number Generator) to let both AP and STA use the right offset for the keystream generation to encrypt and decrypt traffic

Which attacks does IV reuse enable? How can IV reuse happen in WEP?

IV Reuse: IV is reused (two times pad), allows to brute force and decrypt communication between AP and STA

PRNG Restart: WEP PRNG may be restarted every time a laptop it restarted, at restart IV is set to 0, and incremented with every sent packet

IV too short: IVs are reused after a few hours

Explain using WEP as example why data encryption is not sufficient for achieving data integrity.

ey management

- Global master key per ESSID
    - If key leaks, key replacement in all devices is needed
    - No key management protocol for key replacement
    - No session keys, master key directly used
        - Large amount of traffic is encrypted with the same key
        - Combined with other weaknesses, leads to attacks

IV management

- IV size too small (24 bits): reuse
- Real key size small: WEP keys are 54bit and 128 bit small AND!!! include IV
    - real key is 40 bits long and not 64 => Real Time brute force
    - real key is 104 bit long, no brute-force, but cryptographic attack in real time

Cryptography:

- RC4: flawed usage of IVs makes cryptographic attacks possible
- RC4 is by now considered insecure, but WEP weaknesses could be exploited even before the latest RC4 flaws were discovered
- No cryptographic integrity protection (only CRC) Message change possible

Authentication protocol design

- Replay protection is not guaranteed

Implementations

- IV reuse on restart

Which design weaknesses does WEP have? What can be learned from them?

9. Don’t use master keys directly to encrypt communication
- Integrate key management into the system
- Key distribution and update

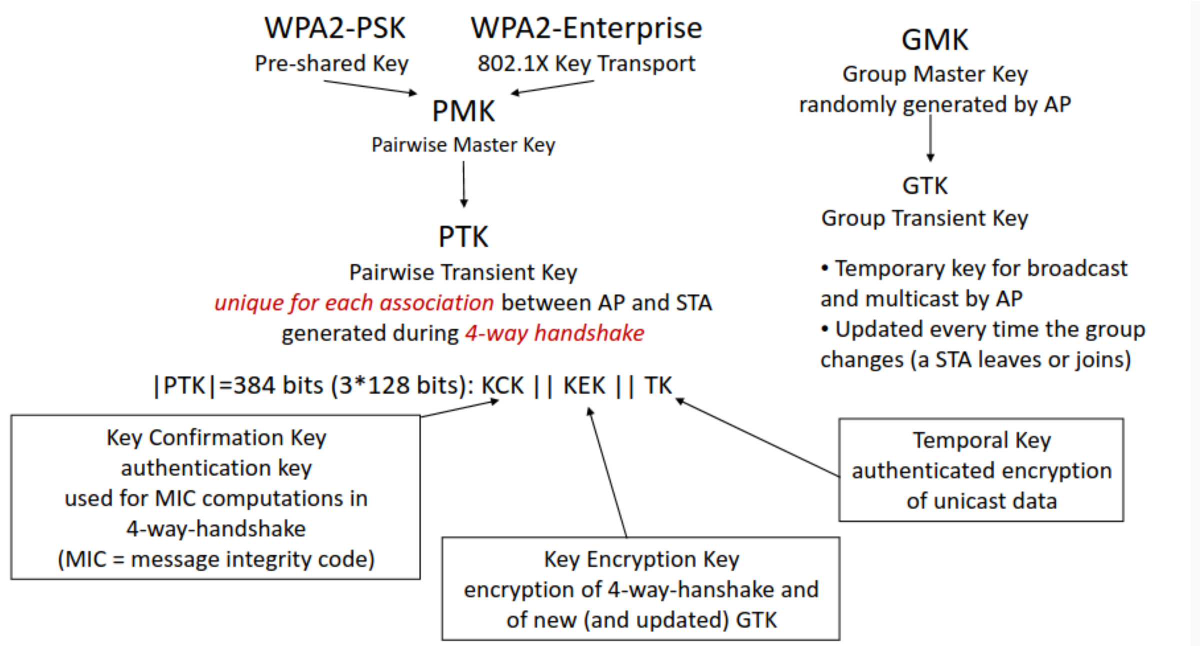
10. When using cryptographic algorithms, always ask experienced cryptographers how to do this properly:
- WEP uses RC4 in an inappropriate way
- Be extremely careful when using stream ciphers
- Think about reuse of initialization vectors and other components that should be used only once

11. Consider replay attacks

12. Always use cryptographically secure integrity protection
- Shared secret key: MAC = Message Authentication Code

What are security goals of WPA2?

Present key hierarchy of WPA2, explain fro which purposes are all keys used.



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

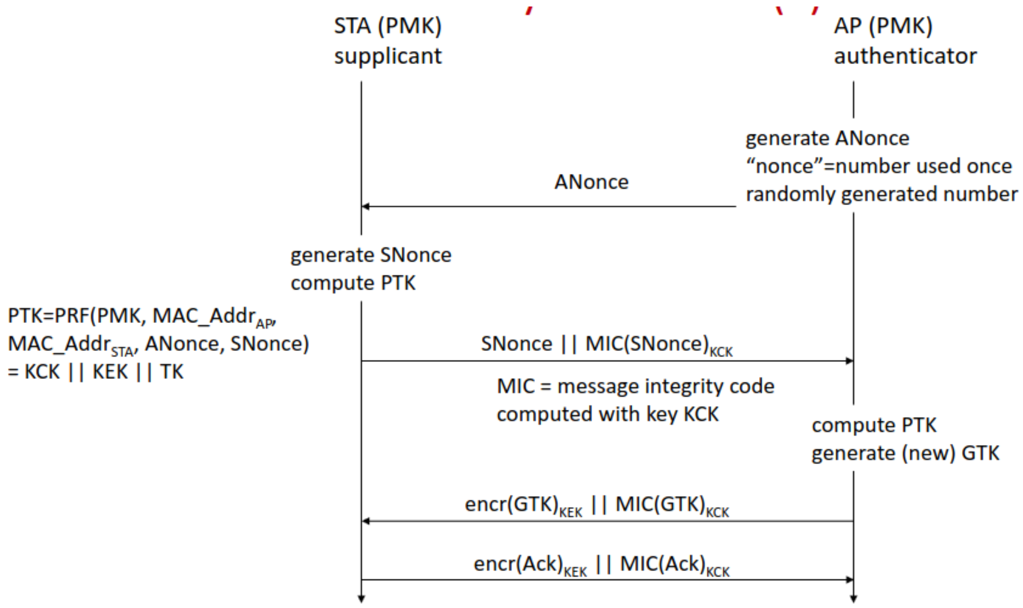
Temporal Key
authenticated encryption
of unicast data

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)

How does 4-way handshake“ work?



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}

What are the differences in security guarantees between WPA2-PSK and WPA2-Enterprise?

WPA2-Enterprise: Individual PMK for each STA-AP pair and each session
WPA2-PSK: Only one PMK

WPA2-Enterprise: Authentication and key agreement via Authentication Server, lower chance of impersonating devices

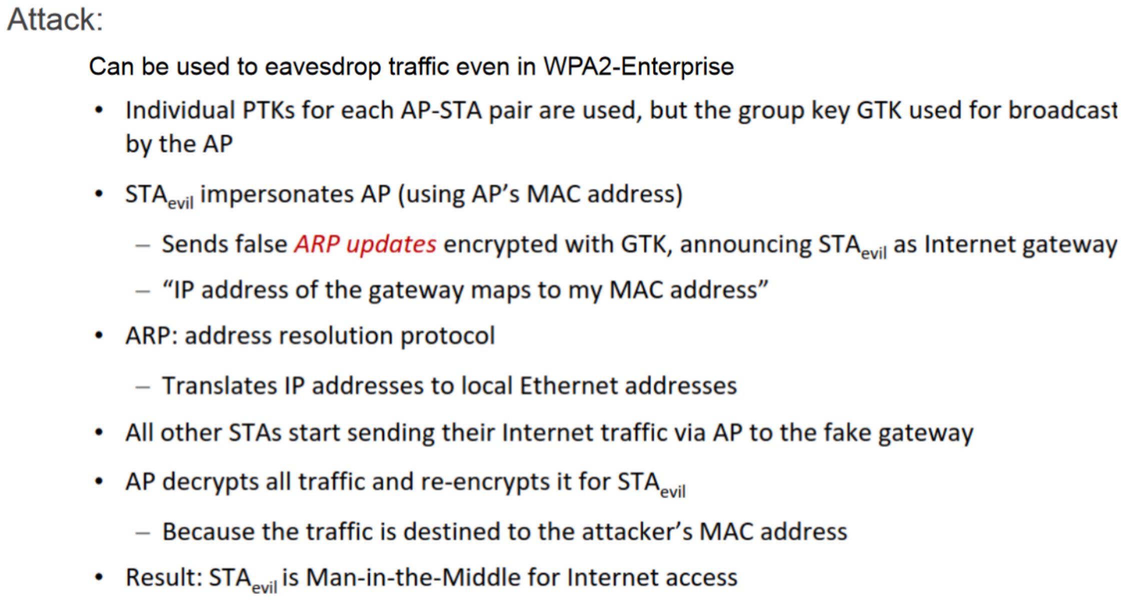
How does WPA2 key cracking work?

Attack:

- PMK = PBKDF2(password, salt),   where salt is the ESSID (=network name)
- Capture handshake
- Try out passwords using a dictionary
- Rainbow tables precomputed for some most popular network names
- sid, linksys, NETGEAR, default, …

Which insider attacks are possible in WPA2-PSK and in WPA2-Enterprise? Which countermeasures should be taken against them?

Insider Attack of WPA2 via Hole 196: Works on both WPA2 variants - Defense: Static ARP Tables, Wireless Intrusion Detection System at ARPs, Use individual keys or digital signatures instead of GTK



Attack:

Can be used to eavesdrop traffic even in WPA2-Enterprise

- Individual PTKs for each AP-STA pair are used, but the group key GTK used for broadcast by the AP
- STA_{evil} impersonates AP (using AP's MAC address)
  - Sends false *ARP updates* encrypted with GTK, announcing STA_{evil} as Internet gateway
  - "IP address of the gateway maps to my MAC address"
- ARP: address resolution protocol
  - Translates IP addresses to local Ethernet addresses
- All other STAs start sending their Internet traffic via AP to the fake gateway
- AP decrypts all traffic and re-encrypts it for STA_{evil}
  - Because the traffic is destined to the attacker's MAC address
- Result: STA_{evil} is Man-in-the-Middle for Internet access

Explain the principle behind the WPA2 Key Reinstallation Attack (KRACK) attack.

- Exploits a vulnerability in the WPA2 protocol's 4-way handshake process.

- Attackers can force reinstallation of an already-in-use encryption key.
- Occurs due to improper handling of cryptographic handshake messages.
- Allows attackers to decrypt and intercept data transmitted over the Wi-Fi network.
- Attackers can also inject malicious content into encrypted traffic.
- KRACK does not require knowledge of the Wi-Fi network's passphrase.
- Vulnerable devices include those running vulnerable implementations of WPA2, affecting a wide range of devices.
- Mitigation involves patching affected devices and updating Wi-Fi access points and client devices.

What are differences in security design between WPA2 and WPA3? Against which attacks does WPA3 protect, compared to WPA2?

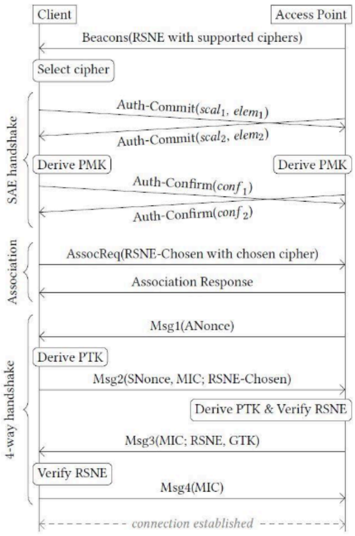
WPA3 does this, WPA2 doesn't do this:

- Designed to fix the WPA2-PSK password cracking attacks
- Uses SAE = Simultaneous Authentication of Equals = "Dragonfly"
- Executed before the 4-way-handshake
- Uses public key cryptography
- For each session: generates a new PMK with high entropy out of the Wi-Fi password

Explain the principle behind the Dragonblood attack.

- Side channel attacks on SAE which leak information about Wi-Fi password
- Cryptographic Attack
- Password cracking by attempted downgrading to WPA2
  - fMeasures against downgrading are implemented
  - Password cracking still possible

IT Security



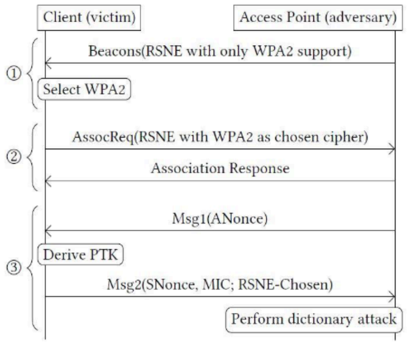

Client (victim)
Access Point (adversary)
①
Beacons(RSNE with only WPA2 support)
Select WPA2
②
AssocReq(RSNE with WPA2 as chosen cipher)
Association Response
③
Msg1(ANonce)
Derive PTK
Msg2(SNonce, MIC; RSNE-Chosen)
Perform dictionary attack

Figure 4: Dictionary attack against WPA3-SAE when it is operating in transition mode, by attempting to downgrade the client into directly using WPA2's 4-way handshake.

Defense?!

Figure 1: Connecting to an AP using WPA3. First the SAE handshake negotiates the master key (PMK), and then the 4-way handshake derives a session key (PTK). To support mesh networks, the SAE handshake was made so both parties can initiate it in parallel (hence the crossed arrows).

Computing, Lecture 7 Zinaida Benenson

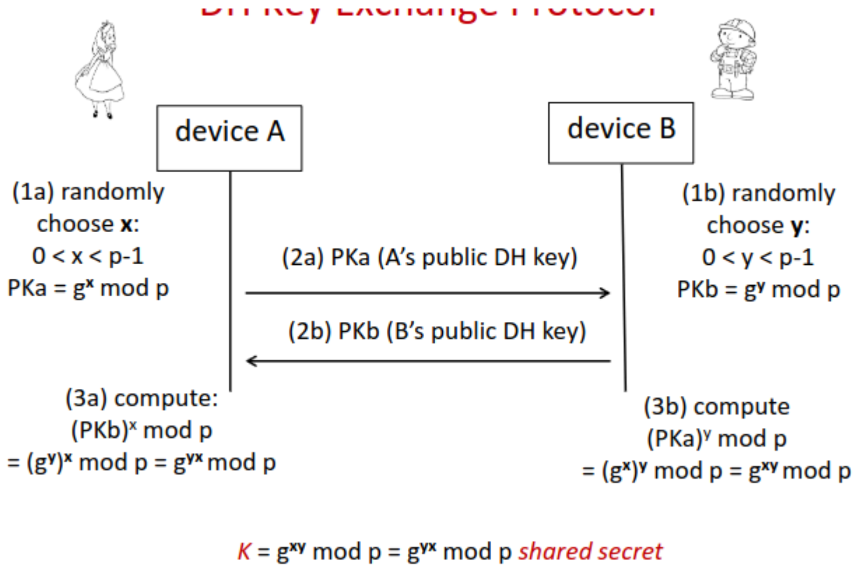
Key Exchange and Pairing

Outline how unauthenticated Diffie-Hellman key exchange between two parties A and B works. You can use a drawing, a text, or a combination thereof for this task (8P)

Include in your explanation:
• public system parameters;
• values chosen by A and B;
• calculations done by A and B;
• content of exchanged messages.

Public Values

- p: large prime number (0.5P)
- g: generator, 1<g<p-1 (0.5P)


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

(7P)

Explain the difference in security properties between unauthenticated and authenticated Diffie-Hellman key exchange. (2P)

Unauthenticated:

- Secure against passive eavesdropping (0.5P)
- Not secure against man-in-the-middle attack and evil twin attack (0.5P)

Authenticated:

- secure against passive eavesdropping, man-in-the-middle and evil twin (0.5P)
- often requires user involvement (like QR code scanning) (0.5P)

Outline how authenticated Diffie-Hellman key exchange is used for device pairing in Bluetooth Numeric Comparison from the user perspective. Provision of cryptographic protocols is not needed here. (4P)

Include in your explanation:
• Minimum capabilities of devices;
• Values that are being authenticated;
• Actions that the user performs.

Answer:

- Both devices need a display and “yes” and “no” buttons (1P)
- During paring, both devices exchange diffie hellmann public keys (1P)
- Both devices do integrity checking with 6 digit number (1P)
- Both devices show the 6-digit code, user compares the codes, and has to click yes or no on both devices depending on if the code is matching (1P)

What are security goals of device pairing?

Security goals: secure channel and CIA

- Authentication of both communication partners
  - No MitM (man-in-the-middle) attacks, no impersonation (Evil Twin)
- Integrity + confidentiality of communication

What are OOB channels used for?

They are used as a second communication channel or action between devices, in order to authenticate DH public keys while DH is executed over the main communication channel (bluetooth).

Which OOB channels do you know? Provide examples of device pairing methods that use OOB channel X.

1. User Interaction (Manual Authentication Protocols)
   a. Enter PIN on both devices, push buttons
2. Additional Channels
   a. Physical connection (USB, docking port)
   b. Secondary transmission technology (infrared, NFC)
   c. Audio (devices beep)
   d. Visual (blinking LEDs, scanning of QR codes)
   e. Objects (moving objects between devices)

How does method Y for device pairing works? Which security guarantees does it provide?

MANA Protocol: Compare PINs of two dervices, press yes/no

Additional Channels:

- Resurrecting Duckling
- Network-in-a-box (NiaB)
- Seeing is Believing
- Shake Well Before Use
    - -
- T2Pair (Touch to Pair)
    - Touch: Movement with random pauses
    - Secure if Attacker can not replicate/mimic movement
- IoTCupid
    - Devices sense different events and compute a group key out of them
    - Is Secure if Attacker is not present in the perimeter
- Wanda Idea: Object-based Pairing
    - Uses (two) Antennas
    - – Detect: Wand detects that it is in close proximity of the target device
    - – Impart: After detection, Wand transmits secrets to the target device

What are the root causes of security vulnerabilities in WPS?

Attack: There is a WPS design flaw (?)

- AP first sends confirmation (or rejection) for first 4 PIN digits
- Then AP sends confirmation (or rejection) for last 3 PIN digits
- Last PIN digit is a checksum for the other 7 digits
- Number of attempts reduced to 11000
- Implementation flaw
    - No back off after unsuccessful attempts or too short back offs
    - Try out PIN=0, if not success, then try out PIN=1 till success
    - Depending on WPS implementations, takes some minutes or some hours to crack the PIN by sending candidate PIN parts to the AP

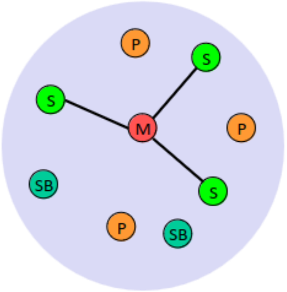
Bluetooth

What were the goals of developing Bluetooth technology?

- Wireless Personal Area Network (WPAN)
- Universal radio interface for ad-hoc wireless connectivity
- Short range (10 m), low power consumption
- Voice and data transmission, approx. 1 Mbit/s data rate

What does Bluetooth system architecture look like? (2 Architectures)

Piconet:



M=Master
S=Slave

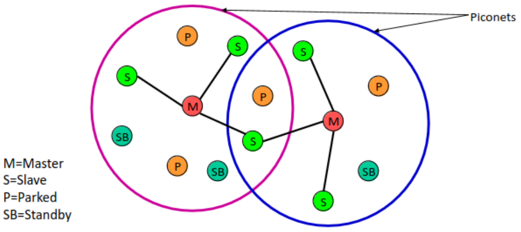
P=Parked: stay synchronized, listen to the traffic

SB=Standby: not in the piconet, may try to join

- Collection of Bluetooth devices
  - Master Node (one per Piconet): Coordinates traffic
  - Slaves (7 slaves can exits): Synchronize with Master

  - Parked (200 can be parked): Sty Synchronized, listen to traffic
  - Standby: non in the piconet, might join
- Each piconet has unique frequency hopping pattern
- Participation in a piconet = synchronization to hopping sequence => Slaves

Scatternet:



- = Linking of multiple co-located piconets through the sharing of common master or slave devices
- Devices can be slave in one piconet and master of another

What does device pairing mean? Which security goals does it have?

= Key establishment between two devices

- Devices do not share any common secrets
- "Have never met before"

Security goals: secure channel and CIA

- Authentication of both communication partners
  - No MitM (man-in-the-middle) attacks, no impersonation (Evil Twin)
- Integrity + confidentiality of communication

How does device pairing work in Bluetooth Versions 1.0-2.0?

General Info:

- Symmetric Keys, 128-bit
- Encryption based on SAFER+ algorithm
- Custom Stream Cipher (60 bit strength)

Explain key hierarchy in Bluetooth 1.0-2.0

Initialization key (function of PIN)

- Temporal key for link key establishment protocol
- Used when devices meet for the first time, or if link key "forgotten"
- Devices can only store a limited number of link keys

Link key

- Generated by both devices from the initialization key
- When devices meet again, they run a protocol to prove the possession of the link key to each other

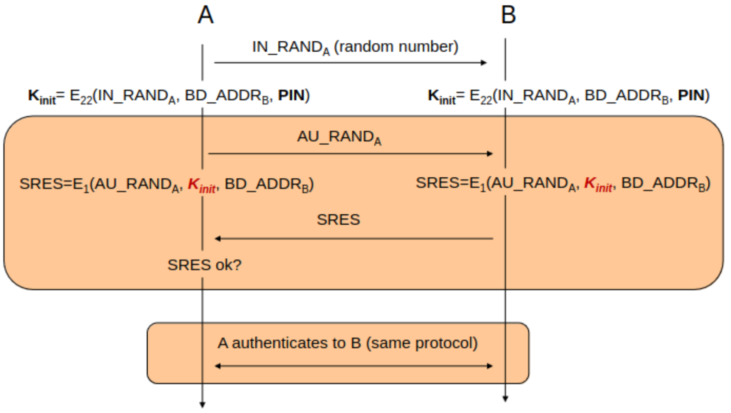
Encryption key

- For data encryption, generated from the link key

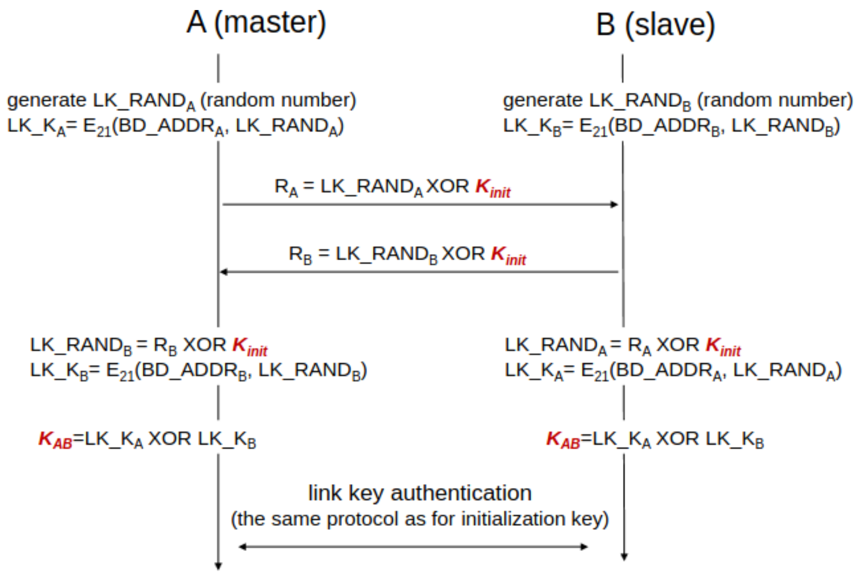
Explain the three Blueooth 1.0-2.0 key generation protocols.

Initialization key Generation and Authentication:

(to verify that A and B have the same key)


A
B
IN_RAND_A (random number)
$K_{init}= E_{22}(IN\_RAND_A, BD\_ADDR_B,$ **PIN**$)$
$K_{init}= E_{22}(IN\_RAND_A, BD\_ADDR_B,$ **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)

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)

Link Key Update

- Use the same protocol as for the link key generation
- Uses previous link key KAB instead of Kinit
- Helps against attackers that are not present permanently

Encryption Keys

Individual shared key between master A and slave B. How it works:

- Random number generated by A is sent to B
- A and B compute a Key based on the random number and the Link Key

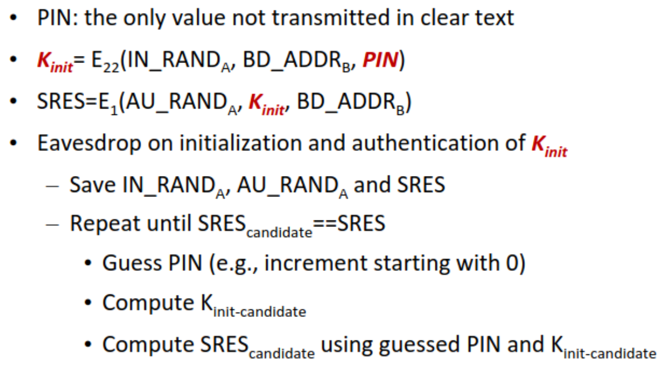
Broadcast Encryption Key. How it works:

- Key generated by master for each session
- Transmitted to slaves using individual shared key

Which 8 vulnerabilities does Bluetooth 1.0-2.0 security have?

1. PIN Cracking Attacks are possible
2. Bluetooth devices are often configured to accept connections to arbitrary devices and send their BD_ADDR to them. User can be tricked to accept connection
   a. Attacks: user tracking, malware, data stealing
3. Weak Cryptography (SAFER+ and E_0 weaknesses)
4. Encryption IV reuse (2 times pad, pads repeat after 23 hours of uninterrupted connection)
5. No integrity check in encryption algorithm (CRC used, same as in WEP)
6. No end-to-end encryption (intermediate devices can decrypt traffic)
7. Security can be switched off through negotiation between devices
8. Several Implementation Vulnerabilities and Attacks (BlueSniff, …)
   a. Remote Execution, Man in the middle, etc

Explain Bluetooth 1.0-2.0 Passive PIN cracking.



- PIN: the only value not transmitted in clear text
- K_{init}= E_{22}(IN_RAND$_A$, BD_ADDR$_B$, ***PIN***)
- SRES=E_1(AU_RAND$_A$, K_{init}, BD_ADDR$_B$)
- Eavesdrop on initialization and authentication of K_{init}
  - Save IN_RAND$_A$, AU_RAND$_A$ and SRES
  - Repeat until SRES$_{candidate}$==SRES
    - Guess PIN (e.g., increment starting with 0)
    - Compute $K_{init\text{-}candidate}$
    - Compute SRES$_{candidate}$ using guessed PIN and $K_{init\text{-}candidate}$

Explain Bluetooth 1.0-2.0 the two Active PIN cracking attacks.

Active attack 1

- Initialization key generation: start using any PIN$_X$, compute K_{init_X}
- Initialization key authentication: receive SRES$_B$ from B, perform PIN guessing as above
- SRES$_B$ contain the right PIN!

Active attack 2

- Force A and B to repeat pairing
- Impersonate one if the devices and pretend that the device forgot the link key
- “Forgot key” messages are (necessarily!) sent insecurely

Explain SSP: Secure Simple Pairing (starting with Bluetooth v2.1)

More secure connection establishment and key exchange via SSP with Diffie-Helmann:

4. Devices initiate pairing process
5. Devices Authenticate themselves to each other
6. Diffie-Hellman is used to securely generate a Encryption Key for further communication

What are advantages and disadvantages of symmetric crypto for Bluetooth pairing compared to asymmetric crypto?

Unauthenticated

- Secure against passive eavesdropping
- Vulnerable to man-in-the middle (MitM) and impersonation (Evil Twin) attacks
- Usual realization: Diffie-Hellmann (DH) key exchange

Authenticated

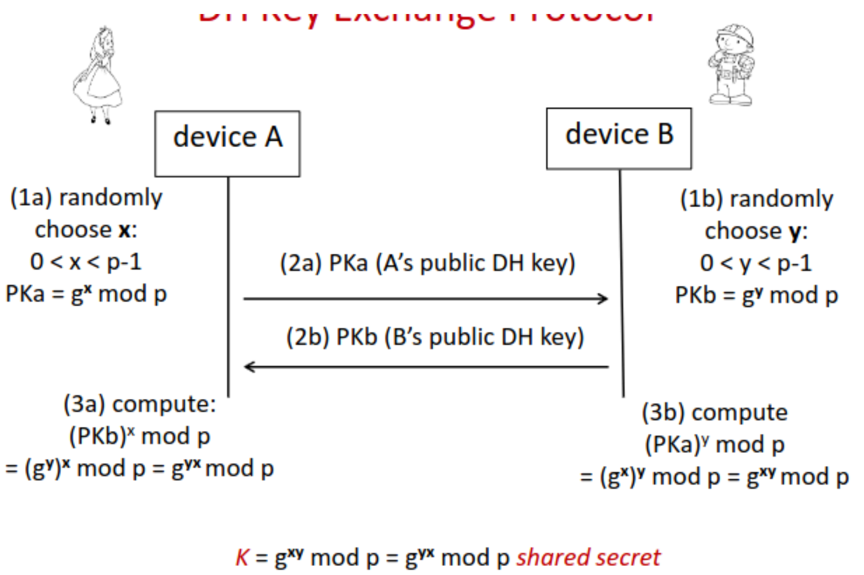
- Secure against MitM & Evil Twin attacks
- Often requires user involvement
- Enter PIN, compare two strings, scan QR code, …
  - Usual realization: authenticated DH key exchange

How does unauthenticated DH work?

Public Values

- p: large prime number (1024 bits)
  - Multiplicative group modulo p: {1, 2, …, p-1}
- g: 1 < g < p-1
  - g is generator of multiplicative group modulo p

DH Key Exchange Protocol



DH Key Exchange Protocol

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*

Explain the difference in attacker models between unauthenticated and authenticated DH.

Unauth. DF:

- Protects against passive eavesdropping
- Active Eavesdropping still possible
- Man in the middle attack possible

Auth DH:

- Human-assisted authentication
- Protects against active and passive eavesdropping and man in the middle attack

Which values are authenticated during the authentication phase of MANA protocols?

Numeric Comparison:

Authenticated DH: integrity checking with the 6-digit number as authenticator

Passkey Entry:

Authenticated DH: 6-digit shared secret

OOB:

DH executed over Bluetooth, DH public keys authenticated via OOB (

Just works

Unauthenticated Diffie-Hellman

How does SSP method X work from the user perspective? On which assumptions does it rely for security? Which user interaction is required?

Numeric Comparison

- User has two devices with displays and yes/no buttons
- User wants to the two devices
- During pairing, both devices show 6 digit hash of the public DH keys
- User has to compare the numbers
    - If the numbers are the same, push "yes" on both devices, otherwise push "no"
- Assumption: User actually compares the numbers probably, and presses the buttons accordingly

Passkey Entry

- Device A has a display, device B has a keypad, or both device have keypads, but no displays

    Process (method 1):
    - One device displays a randomly generated secret 6-digit number N
    - User enters N into another device
    - Then the devices authenticate their DH key using N

    Process (method 2):
    - User "generates" N and enters it into both devices
    - Then the devices authenticate their DH key using N
- Assumption:
    - Only secure if N is really random, and it N is a nonce
    - Passkey should be difficult to guess
    - Passkey can be used only once
    - Last step (user checks whether both devices displayed OK) is necessary

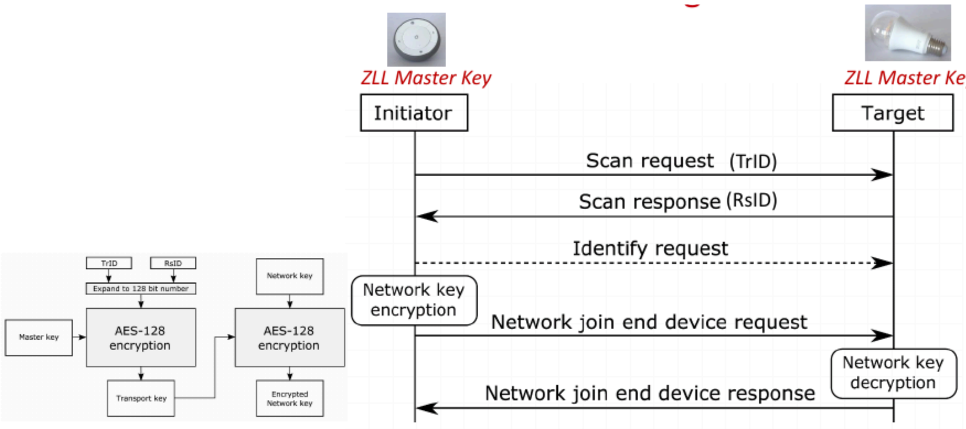
OOB (out-of-band)

- Different User Actions possible
    - Use PINs to enter in both devices, use RFID and hold devices close together, use infrared and hold devices closer together
    - Example: "Seeing is Believing"
        - Device A can generate and display a bar code
        - Device B can scan the displayed barcode
        - => Visual OOB channel
- Assumptions:
    - OOB channel is different from the primary wireless channel
    - Devices setup a connection over the primary channel, e.g., using unauthenticated DH

Zig-Bee

Outline how Zigbee 3.0 Touchlink Commissioning uses a global master key for joining new devices to an existing network. You can use a drawing, a text, or a combination thereof for this task. (4P)

Include in your explanation:
• participating devices and their roles;
• cryptographic secrets, cryptographic algorithms and other essential information;
• content of exchanged messages.
Answer:



- Initiator: usually remote control or router
- Target: light bulb or other ZigBee device with dedicated functions
- Both possess the ZLL (ZigBee Light Link) Master Key
  - Initiator sends TrID: transaction identifier, 32-bit, randomly generated
  - Target sends RsID: response identifier, 32-bit, randomly generated
- Identify request: initiator asks target to identify itself if many targets are available
- Network join end device request
  - Initiator sends network key (NWK) to target encrypted with the master key
  - TrID and RsID are used to make encrypted message different for each different for each commissioning

Which two attacks were made possible due to the leakage of the ZigBee Touchlink Commissioning master key? (2P)

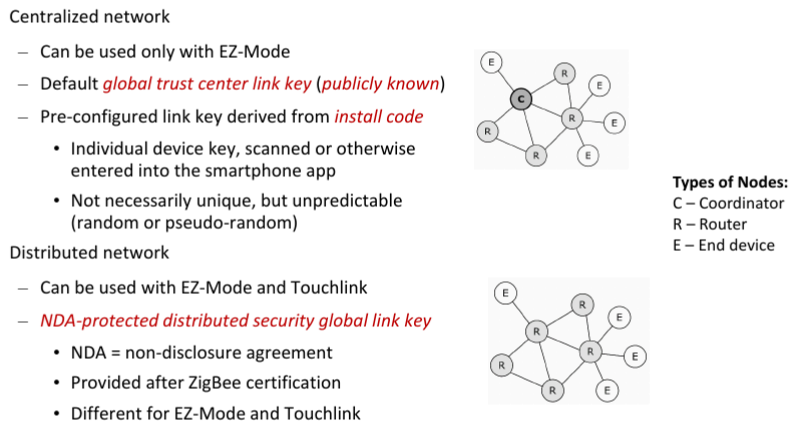
Hijack Attack: (0.5P)

- Join target to attacker‘s network (0.25P)
- Send commands: turn on/off, change color, open/close (e.g., door lock) (0.25)

Network key extraction: (0.5P)

- eavesdrop until user joins a new device to the network (or force user to do so by reset-to-factory-new-atatck) (0.5P)

Outline security architectures of centralized and distributed ZigBee 3.0 networks and compare their security.

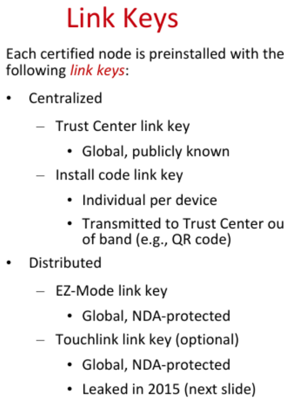

Centralized network

- Can be used only with EZ-Mode
- Default *global trust center link key* (*publicly known*)
- Pre-configured link key derived from *install code*
  - Individual device key, scanned or otherwise entered into the smartphone app
  - Not necessarily unique, but unpredictable (random or pseudo-random)

Distributed network

- Can be used with EZ-Mode and Touchlink
- *NDA-protected distributed security global link key*
  - NDA = non-disclosure agreement
  - Provided after ZigBee certification
  - Different for EZ-Mode and Touchlink

Types of Nodes:
C – Coordinator
R – Router
E – End device


Link Keys

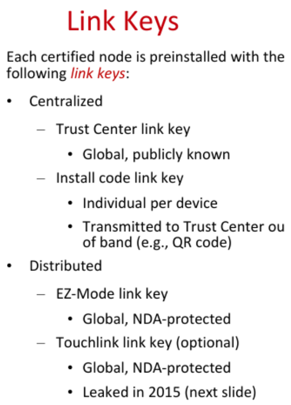
Each certified node is preinstalled with the following *link keys*:

- Centralized
  - Trust Center link key
    - Global, publicly known
  - Install code link key
    - Individual per device
    - Transmitted to Trust Center ou of band (e.g., QR code)
- Distributed
  - EZ-Mode link key
    - Global, NDA-protected
  - Touchlink link key (optional)
    - Global, NDA-protected
    - Leaked in 2015 (next slide)

What are link keys in ZigBee 3.0 networks are used for? Which link keys are installed on each ZigBee 3.0 node? Which of them are required, and which are optional?

Q: What are link keys in ZigBee 3.0 networks are used for?
A: Linkey keys are using for commissioning (=starting a new network or joining a new node to the network). It is also used to distribute the network key to newly joining devices

Q: Which link keys are installed on each ZigBee 3.0 node? Which of them are required, and which are optional?


Link Keys

Each certified node is preinstalled with the following *link keys*:

- Centralized
  - Trust Center link key
    - Global, publicly known
  - Install code link key
    - Individual per device
    - Transmitted to Trust Center ou of band (e.g., QR code)
- Distributed
  - EZ-Mode link key
    - Global, NDA-protected
  - Touchlink link key (optional)
    - Global, NDA-protected
    - Leaked in 2015 (next slide)

What are usability advantages of Touchlink commissioning compared to EZ-Mode with install codes?

???

What is the ZLL master key used for?

The ZLL master key was protected by an NDA, but got leaked on twitter.
It is an (optiona) global key used by distributed ZigBee networks, and can be used for EZ-Mode and Touchlink (although each ahs a different ZLL Master Key).

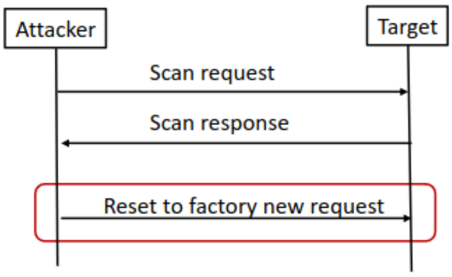
Which attacks on Touchlink do not require the knowledge of the ZLL master key?

Attack: Identify Action Attack

- Doesn't require knowledge of any cryptographic material
- Trigger identify action (e.g., blinking, beeping, dimming) of target device
  - Even if the device is already in a network
  - No authentication (inter-PAN frame)
- Field to specify duration: 16bit ≈ 65000 seconds ≈ 18 hours => Bulb blinks until it runs out of battery, blocks other operations of the lamp
- Recovery: manually disconnect from power source

Attack: Reset to Factory-New Attack

- Doesn't require knowledge of any cryptographic material
- Reset target to the factory-new state
  - Even if the device is already in a network
  - No authentication (inter-PAN frame)
- Threat scenario: access to restricted area
  - Touchlink-enabled door lock
- Reset to factory-new door probably unlocks
- Recovery: recommission the affected devices



Which attacks on Touchlink require the knowledge of the ZLL master key?

Attack: Hijacking - Attack with knowledge of the leaked global master key

- Active attack: requires interaction
- Join target to attacker's network
- Send commands: turn on/off, change color, open/close (e.g., door lock)
- Works even if the device is already joined to another network

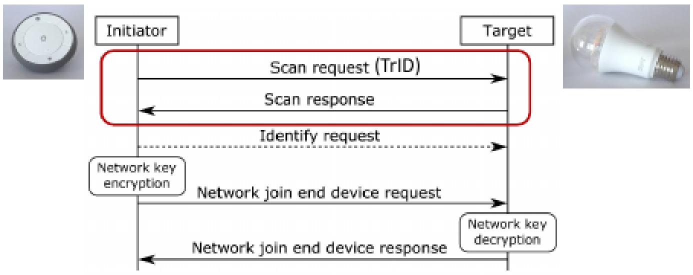
Attack Network Key Extraction - Attack with knowledge of the leaked global master key

- Passive attack: eavesdropping on touchlink commissioning
- GE and Osram: User has no interface (on smartphone app) to trigger Touchlink
- How long should the attacker wait till user commissions a device?
  - „Motivate" user to re-commission any device by reset-to-factory-new attack

How does <insert attack here> work?

Prerequisite of ALL attacks: Active Device Scan

get is already joined to a network



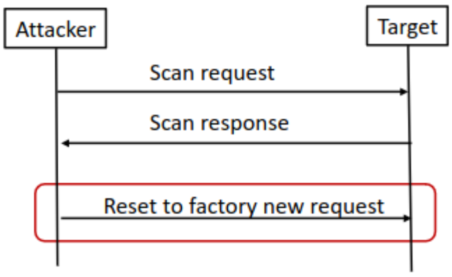
- Touchlink commands are accepted by targets only if previously they received a scan request with the same TrID (transaction ID)
- Scan for touchlink-enabled devices in the wireless range
    - Works even if target is already joined to a network

Attack: Identify Action Attack

- Doesn't require knowledge of any cryptographic material
- Trigger identify action (e.g., blinking, beeping, dimming) of target device
    - Even if the device is already in a network
    - No authentication (inter-PAN frame)
- Field to specify duration: 16bit ≈ 65000 seconds ≈ 18 hours => Bulb blinks until it runs out of battery, blocks other operations of the lamp
- Recovery: manually disconnect from power source

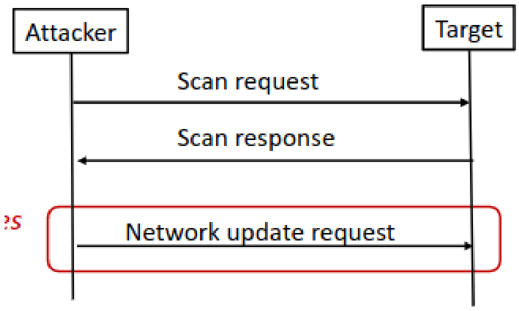
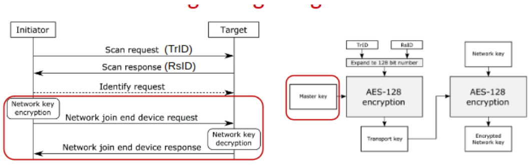
Attack: Reset to Factory-New Attack

- Doesn't require knowledge of any cryptographic material
- Reset target to the factory-new state
    - Even if the device is already in a network
    - No authentication (inter-PAN frame)
- Threat scenario: access to restricted area
    - Touchlink-enabled door lock
- Reset to factory-new door probably unlocks
- Recovery: recommission the affected devices



Attack: Permanent Disconnect Attack

Two attack possibilities

<table>
<tr><td>Change wireless channel of target: network update request</td><td>Join target to garbage network</td></tr>
<tr><td>
</td><td>

Network key is encrypted using AES-Encryption in ECB-mode

Not AES-CCM (authenticated encryption) as for network communication

AES-ECB does not support authentication, only encryption (no integrity protection)

Attack:
– Send a random 128-bit garbage to the target
– Target will decrypt “garbage” and join non-existing network with unknown network</td></tr>
</table>

	key

Recovery: physical reset

- Osram Lightify: turn on 3 seconds, off 5 seconds repeat five times
- Philips Hue: no physical reset possibility found, possibly no user- driven recovery
- Attacker can recover anytime using the same toolkit as for the attack

Possible threat scenarios: DoS, ransom

Attack: Hijacking - Attack with knowledge of the leaked global master key

- Active attack: requires interaction
- Join target to attacker‘s network
- Send commands: turn on/off, change color, open/close (e.g., door lock)
- Works even if the device is already joined to another network

Attack: Network Key Extraction - Attack with knowledge of the leaked global master key

- Passive attack: eavesdropping on touchlink commissioning
- GE and Osram: User has no interface (on smartphone app) to trigger Touchlink
- How long should the attacker wait till user commissions a device?
    - „Motivate“ user to re-commission any device by reset-to-factory-new attack

Reset to Factory Attack

See next question

How is proximity check used in ZigBee 3.0 Touchlink? To which attacks is it susceptible?

ZigBee Proximity Check

Limits range of accepting touchlink commands

If receiving signal strength (RSS) > predefined threshold, then send scan response

Bug:

- A scan request with TrID=0 for a scan request is invalid
  => rejected if received with scan request
- All other inter-PAN commands, if sent with TrID=0, are accepted by the bulb without proximity check (as result of a programming bug)

Reset to Factory Attack:

- Can reset any bulb to factory new without previous scanning and without proximity check
- If a bulb is reset to factory new, it can be joined to new networks without proximity check

What are root causes of insecure commissioning modes in ZigBee 3.0?

Touchlink commissioning in insecure by design

- A single touchlink device in the network can expose network key

- Touchlink commands are accepted by targets only if previously they received a scan request with the same TrID (transaction ID) => Attacker sends scan requests, Scan for touchlink-enabled devices in the wireless range
    - Works even if target is already joined to a network
- Global master key cannot be renewed due to backwards compatibility requirements

RFID

Briefly outline how RFID is used in supply chain management. (2P)

RFID is used to keep track of objects without having to scan them manually (1P)
An RFID scanner is used to scan for objects along the supply chain, each objects has its own (unique EPC) ID code, and the scanner can then send the IDs and other data to a database used for tracking (1P)

Give one example of a privacy threat that arises in supply chain management due to RFID usage. Justify your answer. (2P)

Spying on people/competitors (1P)
Use unauthorized RFID readers (0.5P) to gather data and information about supply chain management (0.5P)

Outline privacy implications of electronic documents using e-passports as example. Which two privacy attacks did we consider in the lecture, how do they work at a conceptual level? (without going into the details of messages) (4P)

1. Fingerprinting Passport Nationality (1P)
    a. Passports of different countries answer differently to various commands from the readers
    b. Possible to determine nationality of the owner (1P)
2. Tracking via Replay Attack (1P)
    a. Eavesdrop on a legitimate session between a passport and a reader
    b. Record the encrypted message from the reader that contains the passport's nonce
    c. To identify a particular passport, replay this message (1P)

What are RFID tags, which types and standards do you know, which applications are possible?

RFID (Radio-Frequency Identification) tags are small tags with integrated circuits with an antenna which can transceive power, and they store, send, process and receive data.

Three types:
Passive (no battery),
BAP (Battery Powered Passive): Only responds if it receives incoming data, longer range than passive due to battery
Active Battery Powered: Active communication patterns without “outside stimuli” from a reader
Passive RFID with sensors: Changes in environment set and additional bit in memory

Applications: STAP - Security&Safety, Tracking, Authentication, Payment

How does Supply Chain Management using EPC (Electronic Product Code) works?

What are advantages and disadvantages of EPC compared to barcodes?

EPC is more expensive (bardcodes are printed and basically “free”)
EPC works without the need of a line of sight, proper alignment of the reader and product.
EPC has several potential privacy issues, that can make it possible to track and identify people who own/carry products with RFID EPC tags.
…

Explain protocol X for reading EPC tags.

Query Tree
Query Slot

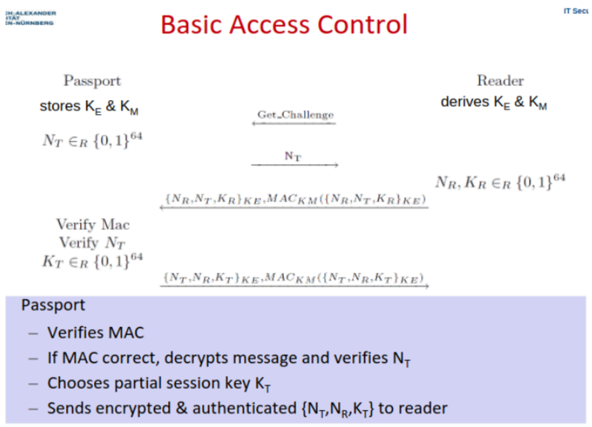
Which security and privacy threats exist in EPC systems? Which technical and sociotechnical countermeasures (e.g., RFID Bill of Rights) do you know?

Which security and privacy aspects should be considered when designing RFID-based documents?

Explain how BAC protocol works. Explain how attack X on electronic passport documents works.

Passport Access Control Protocols
Basic Access Control (BAC):

- Mandatory, symmetric crypto for mutual authentication
- Key derived from machine-readable passport info
- Prevents scanning from third party



Basic Access Control

Passport

stores K_E & K_M

$N_T \in_R \{0,1\}^{64}$

Reader

derives K_E & K_M

$\xleftarrow{\text{Get\_Challenge}}$

$\xrightarrow{N_T}$

$N_R, K_R \in_R \{0,1\}^{64}$

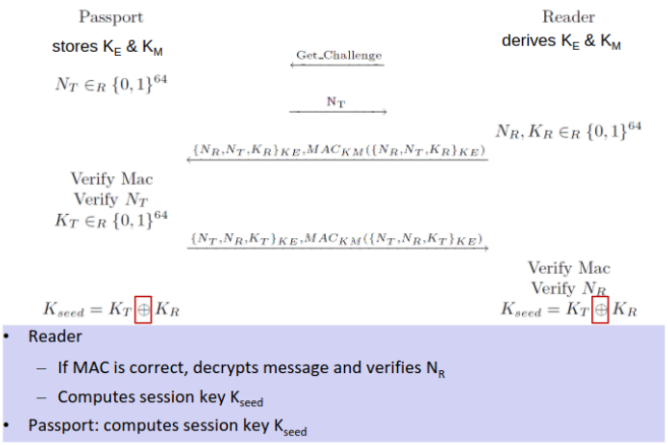
$\xleftarrow{\{N_R,N_T,K_R\}_{KE},MAC_{KM}(\{N_R,N_T,K_R\}_{KE})}$

Verify Mac
Verify N_T
$K_T \in_R \{0,1\}^{64}$

$\xrightarrow{\{N_T,N_R,K_T\}_{KE},MAC_{KM}(\{N_T,N_R,K_T\}_{KE})}$

Passport

- Verifies MAC
- If MAC correct, decrypts message and verifies N_T
- Chooses partial session key K_T
- Sends encrypted & authenticated $\{N_T,N_R,K_T\}$ to reader



Passport

stores K_E & K_M

$N_T \in_R \{0,1\}^{64}$

Reader

derives K_E & K_M

$\xleftarrow{\text{Get\_Challenge}}$

$\xrightarrow{N_T}$

$N_R, K_R \in_R \{0,1\}^{64}$

$\xleftarrow{\{N_R,N_T,K_R\}_{KE},MAC_{KM}(\{N_R,N_T,K_R\}_{KE})}$

Verify Mac
Verify N_T
$K_T \in_R \{0,1\}^{64}$

$\xrightarrow{\{N_T,N_R,K_T\}_{KE},MAC_{KM}(\{N_T,N_R,K_T\}_{KE})}$

Verify Mac
Verify N_R

$K_{seed} = K_T \boxed{\oplus} K_R$

$K_{seed} = K_T \boxed{\oplus} K_R$

- Reader
  - If MAC is correct, decrypts message and verifies N_R
  - Computes session key K_{seed}
- Passport: computes session key K_{seed}

Attacks:

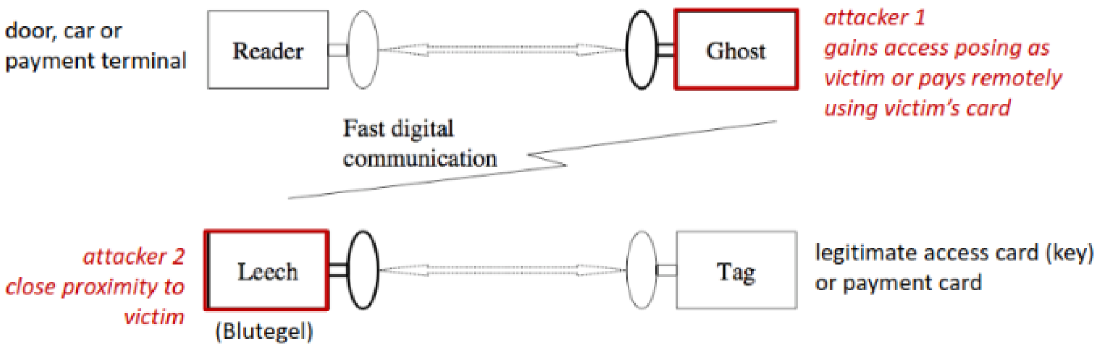
- Cloning
- Tracking via Replay
- Ghost-and-Leech

Which attacks on RFID-based access control systems do you know? How should countermeasures be implemented?

CLoning E-Passport: Predict BAC key by using passport info like date of birth. Countermeasure: Use Extended Access Control and Certificates.

Tracking via Replay Attack: Eavesdrio encrypted message, replay it, and measure the time it takes to receive a rejection (Side-Channel Attack): If the rejection comes faster, it got rejected because of a wrong MAC, if it takes longer, then it failed due to the nonce

Ghost-And-Leech: Impersonate Reader and Card


door, car or payment terminal
Reader
Ghost
attacker 1 gains access posing as victim or pays remotely using victim's card
Fast digital communication
attacker 2 close proximity to victim
Leech
(Blutegel)
Tag
legitimate access card (key) or payment card

Countermeasure: Distance Bounding Protocols to make "ghost" impossible

Why do tags in Query Slot Protocol don't send their ID directly?

If two or more tags are in the same slot, then all tags in that slot send their random 16-bit number to the reader => collision. Then the reader sends a number back (which is either a mixture of both/all RN16 numbers or only one of the RN16 numbers), so all or multiple tags get the wrong RN16 number back from the reader.

The tags then notice that the reader got the wrong number back from the reader, and tell that to the reader.

The reader then chooses a new Q=Q+1, sends it to the tags, and all tags that haven't been inventoried yet adjust/choose new slots.

Random numbers have a lower chance of colliding with each other, which is not the case with using the IDs directly.

It also has a privacy advantage if a random number is used: In Query Tree, the reader also sends the ID of a tag, and usually a higher transmission range than a tag. This gives an attacker two chances to eavesdrop an ID. With query tree, only the tag sends the ID, with a smaller transmission range than a reader.

Need to know

Pervasive Computing (5 parts)

IoT Enablers (4 parts)

IoT (5 parts)

WAN (+3/4 Examples)

WLAN (+ 1 Examples)

WPAN (+ 3 Examples)

Security (Definition)

Security Goals (CIA) + Authentication

4 Parts of evaluating attackers

Privacy (3 definitions)

S&P in IoT (3 parts)

5 steps of S&P Assessment

5 Design Principles of pervasive systems (A. Greenfield)

GSM (+4 of its features/properties)

GSM cellular network

GSM Architecture:

* NSS with OSS

    * HLR

    * EIR

    * AuC

    * MSC

    * VLR

    * GMSC

* RSS

    * MS

    * BTS

    * BSC

Handover Decision

Call setup

GSM S&P Assesement (4 parts)

GSM Threats, Victims (4 examples) and Attackers

4 GSM Security features

2 GSM Security Risks

9 steps of GSM athentification and voice/SMS encryption

* Grafik!

* IMEI

* IMSI

* TMSI

* BSC

* MSC

* HLR

* Encryption Algorithms

Free Call Attack: Ross Anderson's Hack (7 steps)

Free Call Attack: SIM Card Cloning (2 Options)

A3/A8 Algorithms (secure?)

Kerckhoff's Six Principles (name the mos timportant principle)

Security by Obscurity (+how to do it right)

AES

Attack: Stolen or Lost MS and Countermearues

Attack: Backend Eavesdropping

IMSI Catcher + Countermeasure

UMTS

UMTS Architecture

MS

USIM

NodeB

RNC

MSC

SGSN

VLR

HLR

EIUR

AUC

SS7

UMTS Authentication and Key Agreement (14 steps)

Man-in-the-middle GSM-UMTS Degradation Attack (7 steps)

Why does GSM-UMTS Degradation Attack work?

Problem of backward compatibility

Attack: Eavesdropping Encrypted LTE Calls (ReVoLTE)

5G Security Issues (in a nutshell)

5G Authentification

5G AKA Protocol

Cellular Security & Privacy: 12 Lessons Learned

One Time Pad

Two Times Pad: Why it is insecure + Crib Dragging

WEP: IV

Attack: Three Attacks on WEP by abusing IV (prerequisite and attack)

Attack: CRC-based attack on WEP

Attack: Replay Attack on Access Control of WEP

4 WEP Design Issues

WEP: 4 lessons learned

WPA3 (Why it is more secure)

Attack on WPA3: Dragonblood

WPA3: Defenses against Downgrading

7 lessons learned WPA2/3

Attack: Reset to Factory-New Attack

Attack: Permanent Disconnect Attack (two ways)

Attack: Hijacking - Attack with knowledge of the leaked global master key

Attack: Network Key Extraction - Attack with knowledge of the leaked global master key

Attack: Using Proximity Chek Bug to factory-reset Devices

Conclusion of ZigBee Security Analysis

Lessons Learned ZigBee (3+3)

Bluetooth 2.1+ Pairing: Numeric Comparison

Basic Numeric Comparison

Bluetooth 2.1+ Pairing: Passkey Entry

Basic Passkey Authentication Protocol

Prerequisities

Authentication

Necessity of User Check on Both Devices

Bluetooth 2.1+ Pairing: OOB (out-of-band)

Bluetooth 2.1+ Pairing: Just works

Attack: Degradation to Just Works Attack

Attack: Method Confusion Attacks

Unauthenticated DH in BLE (Mut zur Lücke?)

Bluetooth Pairing: 6 Lessons Learned

Resurrecting Duckling

Network-in-a-box (NiaB)

Seeing Is Believing

Shake Well Before Use

T2Pair (Touch to Pair)

IoTCupid

Object-based Pairing: Wanda Idea

Lessons Learnt

Tag Killing (Ensuring Privacy?)

Tag Covering (Ensuring Privacy?)

Blocker Tag (Ensuring Privacy?)

Selective Blocker Tags

RFID: Lessons Learned (so far)

Electronic Documents with RFID or other proximity cards

Passport Access Control Protocols

Basic Access Control (Passport ⇔ Reader)

Example: Cloning E-Passport, 2006

Example: Cloning E-Passport, 2007

Example: Cloning and Changing E-Passport, 2008

Tracking via Replay Attack

E Documents - Lessons learned

Ghost-and-Leech Attacks on Access Control and Payment Systems

Skim Clone RFID Tags of car key

Lessons Learned