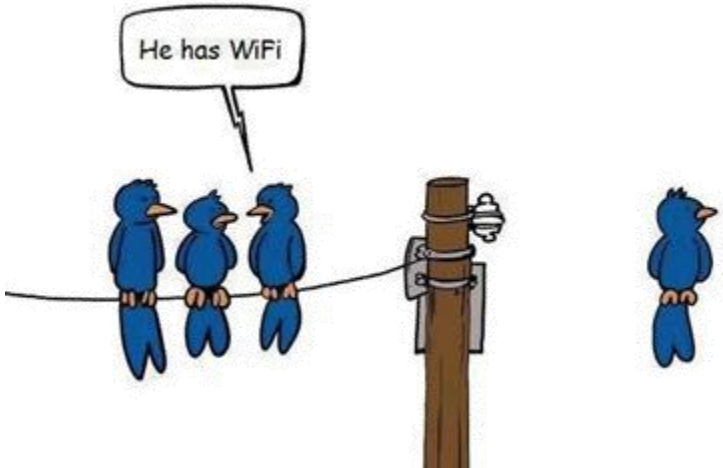
COM-405: Mobile Networks

Lecture 11.0: Wireless Security

Haitham Hassanieh

Slides courtesy of Jean-Pierre Hubaux (with gratitude to Ludovic Barman and Sylvain Chatel who produced some of the slides and Eduard Vlad who recently edited the slides)



EPFL

SENS
Laboratory of SEnsing
& Networking Systems

This course contains material related to security and privacy, including the description of attacks. The purpose is to help you learn how to best protect networks. **In no case should you feel entitled to perpetrate such attacks**, neither against the EPFL network, nor anywhere else.

Outline

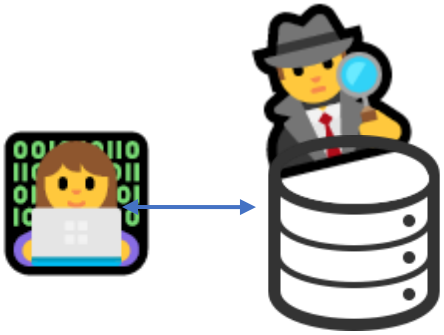
- Brief Reminder of Security and crypto principles
- Challenges in Wireless Networks Security
- Cellular Network Security
- WiFi 802.11 Network Security
- Wireless Pairing & Bluetooth

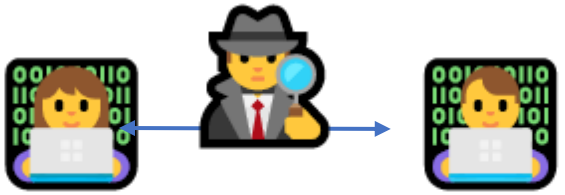
Cryptography in Data Security & Privacy

Cryptography is used to enforce **data security**:

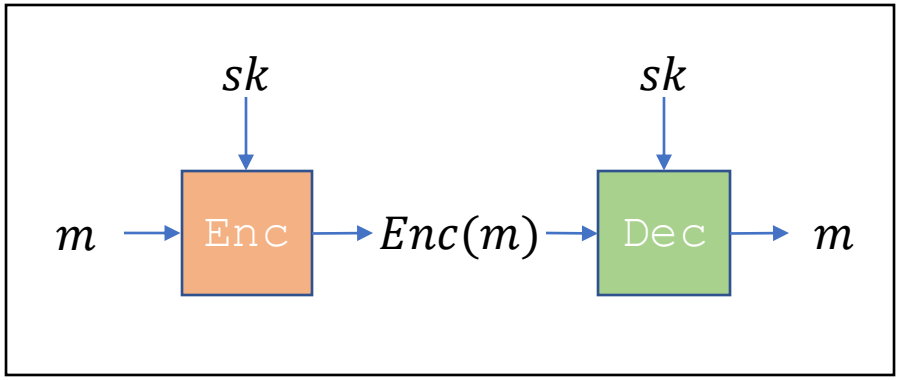
- **Confidentiality**
  - ✓ Encryption
- **Integrity** & **Non-repudiation**
  - ✓ Hash-and-sign

…at rest and ***in transit***



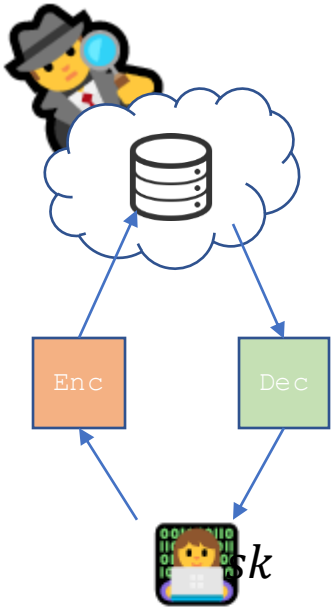


Cryptographic tools – Symmetric encryption

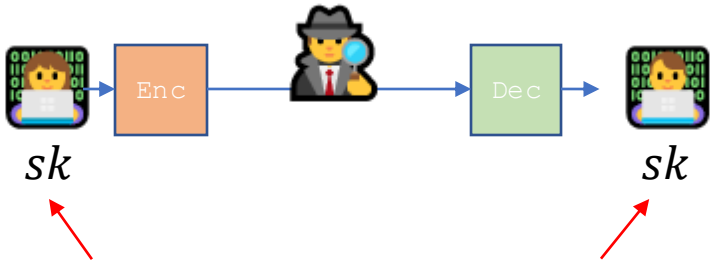


sk = secret key (shared)

At Rest

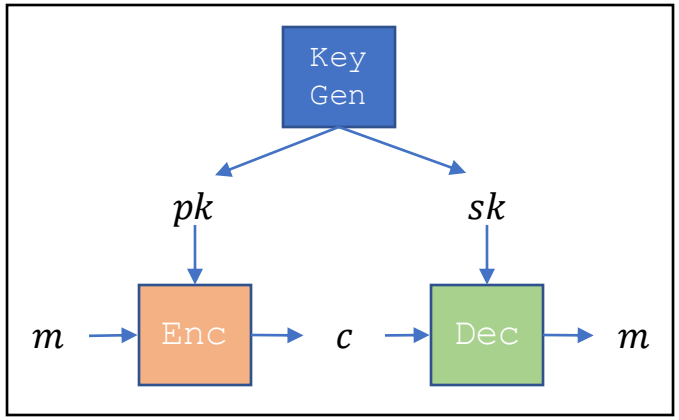


In Transit

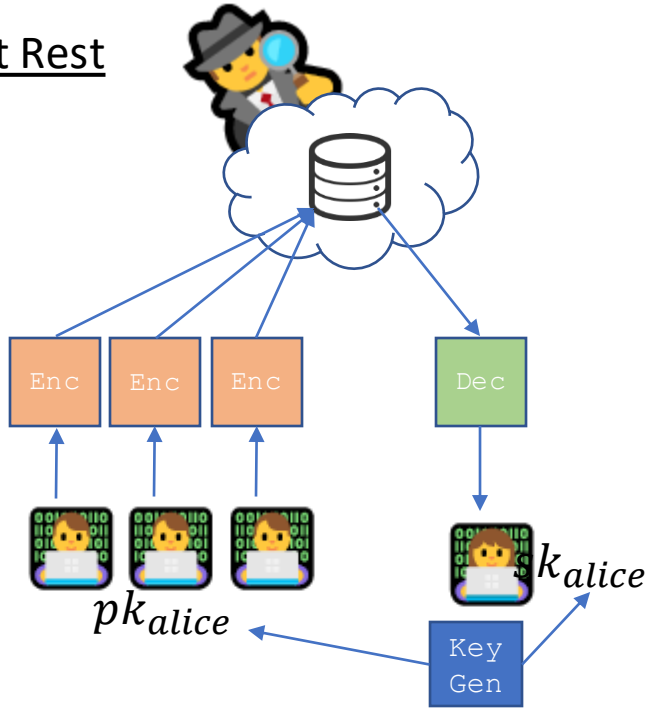


Still need to exchange sk somehow

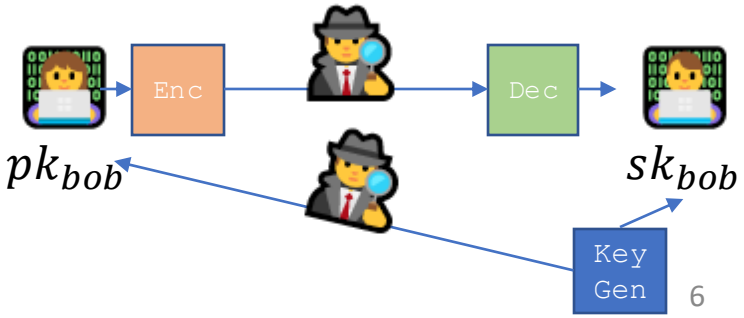
Cryptographic tools – Asymmetric encryption



At Rest



In Transit



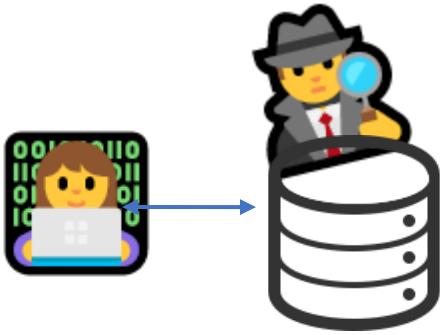
Cryptography in Data Security & Privacy

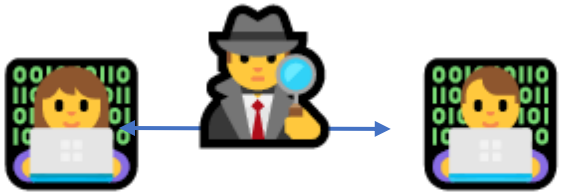
Cryptography is used to enforce **data security**:

- **Confidentiality**
  - ✓ Encryption
- **Integrity** & **Non-repudiation**
  - ✓ Hash-and-sign

Is it enough ?

…at rest and ***in transit***



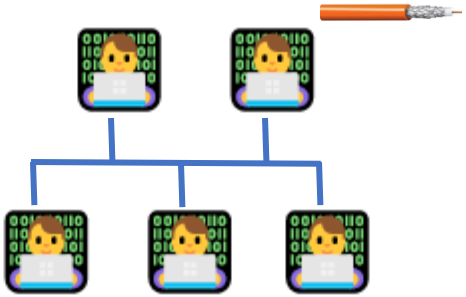


Wireless Network Security - Outline

- Brief reminder of security and crypto principles
- Challenges in Wireless Networks Security
- Cellular Network Security
- WiFi 802.11 Network Security
- Wireless Pairing & Bluetooth

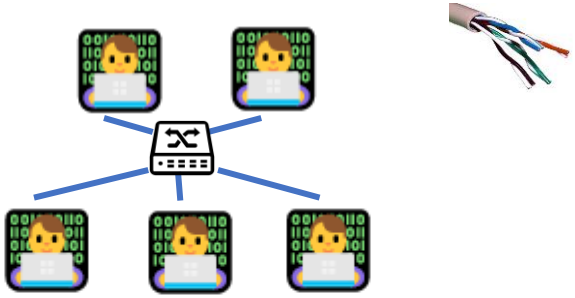
Wired and Wireless Topologies

1982: 10BASE5 Ethernet



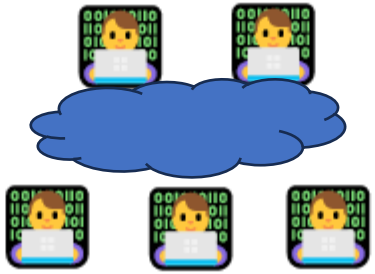
Shared bus for all (CSMA/CD)

Today: 10BASE-T... Ethernet



Partitioned collision domains, no broadcast

Today: Wireless



Air Interface: Shared bus for all

[1]: https://nassaunationalcable.com/cdn/shop/products/Belden9880_1024x1024@2x.png?v=1762488061

Why is security more of a concern in wireless?

1. No inherent physical protection
   - Physical connections between devices are replaced by **logical associations**
   - Sending and receiving messages **do not need physical access** to the network infrastructure (cables, hubs, routers, etc.)

2. Broadcast communications
   - Wireless usually means radio, which has a **broadcast nature**
   - **Transmissions can be overheard by anyone in range**
   - Anyone can generate transmissions,
     - Which will be received by other devices in range
     - Which will interfere with other nearby transmissions and may prevent their correct reception (jamming)

- Eavesdropping is **easy**
- Injecting bogus messages into the network is **easy**
- Replaying previously recorded messages is **easy**
- Illegitimate access to the network and its services is **easy**
- Denial of service is **easily** achieved by jamming

Security Requirements

1. **Confidentiality**
   - Messages sent over wireless links must be encrypted
2. **Authenticity**
   - Origin of messages received over wireless links must be verified
3. **Replay detection**
   - Freshness of messages received over wireless links must be checked
4. **Integrity**
   - Modifying messages on-the-fly (during radio transmission) is **not so easy**, but **possible** …
   - Integrity of messages received over wireless links must be verified
5. **Access control**
   - Access to the network services should be provided only to legitimate entities
   - Access control should be **permanent**
     - It is **not enough** to check the legitimacy of an entity **only when it joins the network** and its logical associations are established, because **logical associations can be hijacked**
6. **Protection against jamming**

Wireless Network Security - Outline

- Brief reminder of security and crypto principles
- Challenges in Wireless Networks Security
- Cellular Network Security
- WiFi 802.11 Network Security
- Wireless Pairing & Bluetooth

Cellular Network Security

Standards that we will (briefly) explore :

- **GSM** (2G), **G**lobal **S**ystem for **M**obile communications
  - introduced in 1991, now deprecated but very useful to explain things
  - It is called 2G, but is actually the first **digital** cellular system (1G designates the analog versions)
- **UMTS** (3G), **U**niversal **M**obile **T**elecommunications **S**ystems
  - introduced in 2001
- **LTE** (4G), **L**ong **T**erm **E**volution
  - introduced in 2009
- **5G**
  - Introduced in 2020

- Objective: get a **sense** of the security measures used by these standards
- It is not meant to be exhaustive

GSM Security

- Main security requirement :
  - Subscriber authentication (for the sake of billing)
    - Challenge-response protocol
    - **Long-term secret key** shared between the subscriber and the home network operator
    - **Supports roaming** **without revealing long-term key to the visited networks (privacy-friendly)**
  - **Missing:** Network authentication

- Other security services provided by GSM :
  - Confidentiality (communications + signaling)
    → over the wireless channel
    → wired network in clear-text
  - Protection of the subscriber's identity from eavesdroppers on the wireless channel
    - Usage of short-term temporary identifiers

GSM Security – SIM

- User devices have Subscriber Identity Module (**SIM**)
- Allows to **authenticate** the device on the network
- It contains:
  - a secret key K_i shared with the home network
  - the user identity i, the **IMSI** (International Mobile Subscriber Identity)
- Security consideration:
  - It is protected (encrypted) with a Personal Identification Number (**PIN**…)
  - If possible, devices don't send the IMSI for privacy reasons (tracking)
    … but agree on a **TSMI** (Temporary Mobile Subscriber Identity)
  - It should be **tamper-resistant**...
    … yet some cloning / extraction attacks succeeded

C.f.: https://www.etsi.org/deliver/etsi_ts/155200_155299/155205/18.00.00_60/ts_155205v180000p.pdf

GSM Security – Cryptography

- Devices use three cryptographic algorithms: **A3**, **A5** and **A8**
  - A3 for **authentication**
    → Produces a 32 bits challenge
  - A8 for **key generation**
    → Produces a 64 bits encryption key

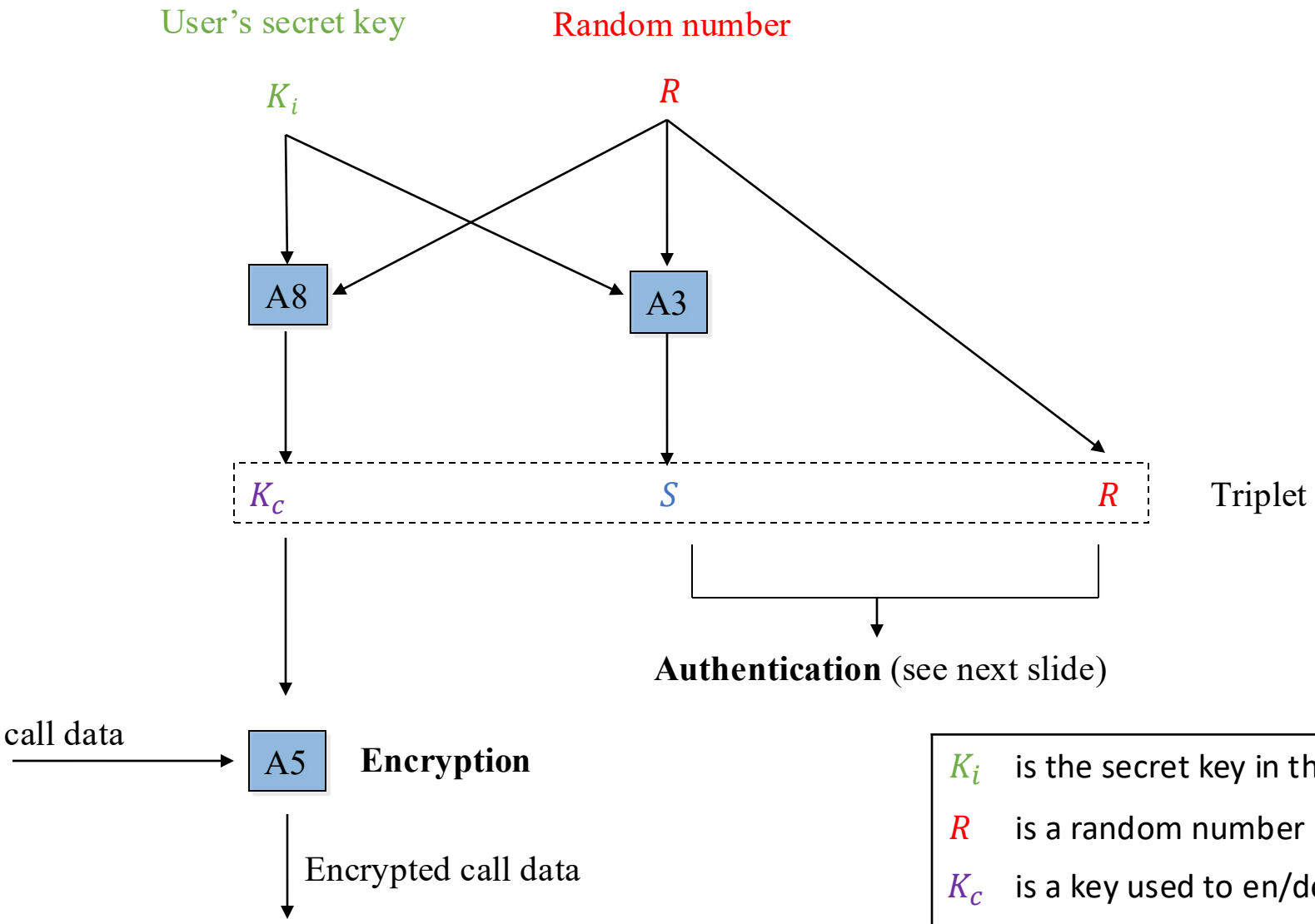
  Pseudorandom functions

  - A5 for **encryption**

  Stream cipher

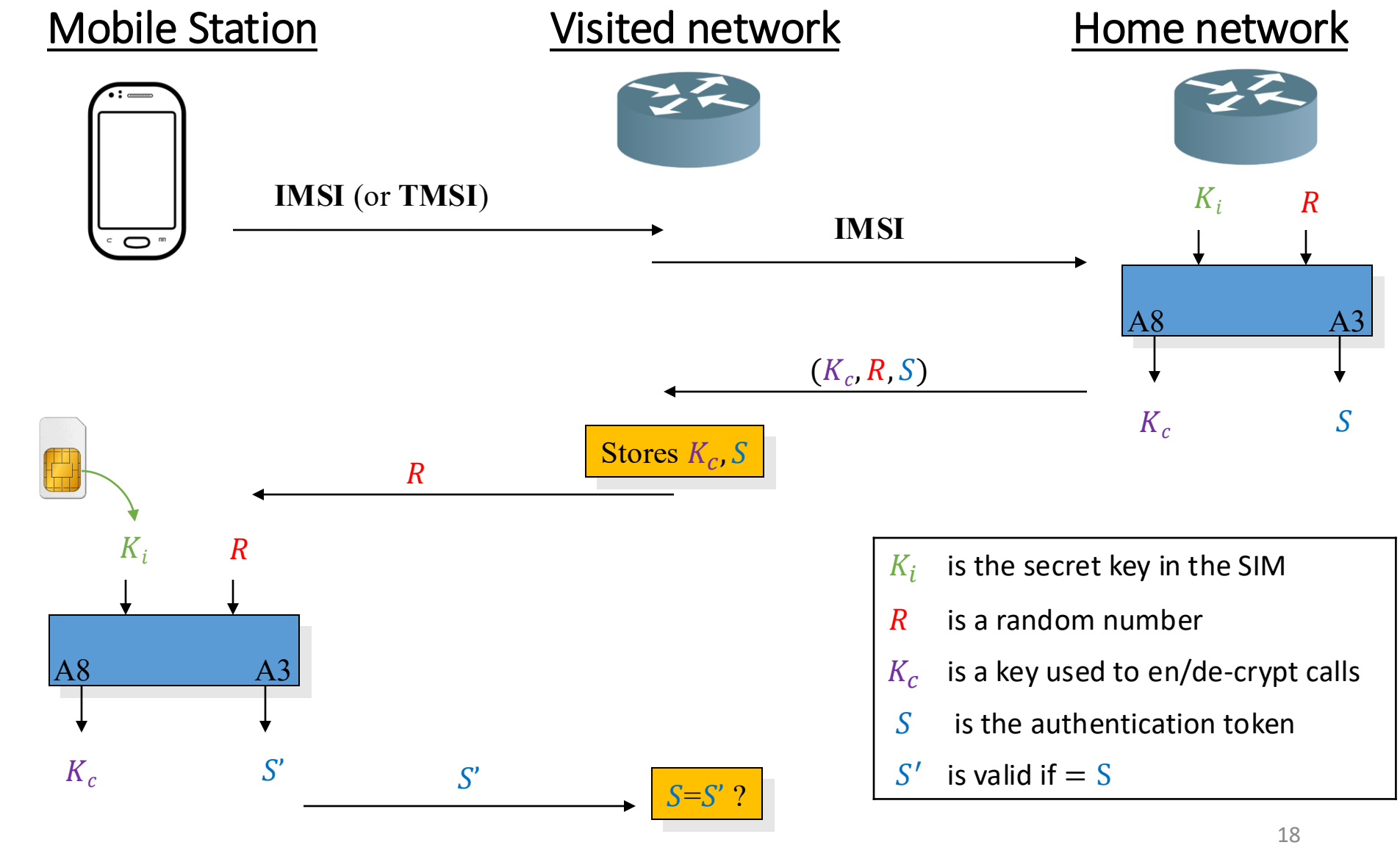
- Left unspecified by the standard: operator may choose
- Often: A5/A8 implemented together as a hash function

GSM Cryptography



K_i	is the secret key in the SIM
R	is a random number (RAND)
K_c	is a key used to en/de-crypt calls
S	is the authentication token

GSM Authentication



GSM Weaknesses – "By design"

1. Unilateral authentication:
   - The network authentifies the user, but not the opposite
   - A fake base station can **pretend to be the legitimate network**
   - E.g it can collect identities (IMSI) to spy on users (movement profiles)
   - E.g it can mount a MitM and decrypt the traffic

2. No Security within the wired network:
   - Call data is decrypted at the **visiting network**
   - No end-to-end confidentiality
   - Signaling messages are not protected

3. A5 secret key can be extracted from recorded traffic
   - **Missing salt in encryption** enables attackers to pre-compute **Rainbow Tables** [1]

[1]: https://srlabs.de/blog/decrypting-gsm

GSM Weaknesses – Attacks

3. Over-the-air cracking:
   - Cryptographic weakness in the 1st generation of A3/A8
     (that allowed an adversary to decrypt calls)
   - Extraction of IMSI (to spy on user) via packet injection

4. SIM Cloning:
   - Allows to receive calls instead of someone
   - … or make «free» calls
   - Yet the network will probably detect / ban the two identical SIMs
   - It is thus a rather short-lived attack

GSM Weaknesses (cont'd)

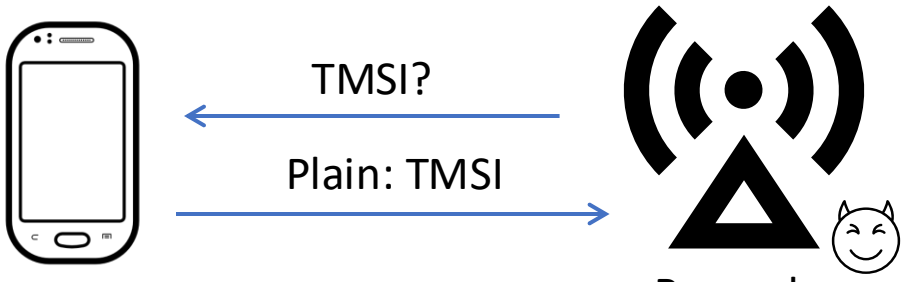
5. Security by obscurity for the crypto protocols
   - Closed-source design
   - Was reverse-engineered
   - Finally made public (but when it was already implemented and widespread)
   - A very bad decision for cryptographic protocols!

Note:

the Snowden-leaked documents show that the NSA is able to «process encrypted A5 [GSM] data» [1]

[1] Timberg, Craig; Soltani, Ashkan (13 December 2013). "By cracking cellphone code, NSA has ability to decode private conversations», Washington Post, Dec. 2013

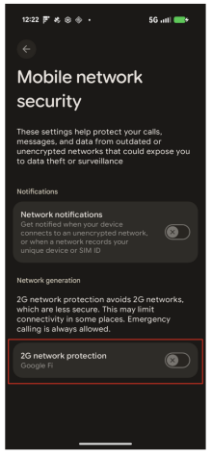
Attacks Today: IMSI Catching in 2G



Attack due to missing mutual authentication:

- Rogue base station overshadows the authentic operator's signals
- Collection of unique and permanent identifiers: TMSI, IMEI

➔ Enables tracking and collection of movement profiles



➔ **Recommendation: Disable 2G (GSM) if possible [1]**

[1]: https://source.android.com/docs/security/features/cellular-security/toggle-2g

UMTS Security improvements

- Network to Device authentication:

  Protection against the **false base station attack**

- Increased key length:

  allows for stronger encryption and integrity verification algorithms

- Security mechanisms within and between networks

  Through authentication and encryption on the wire

- Introduced authentication sequence number

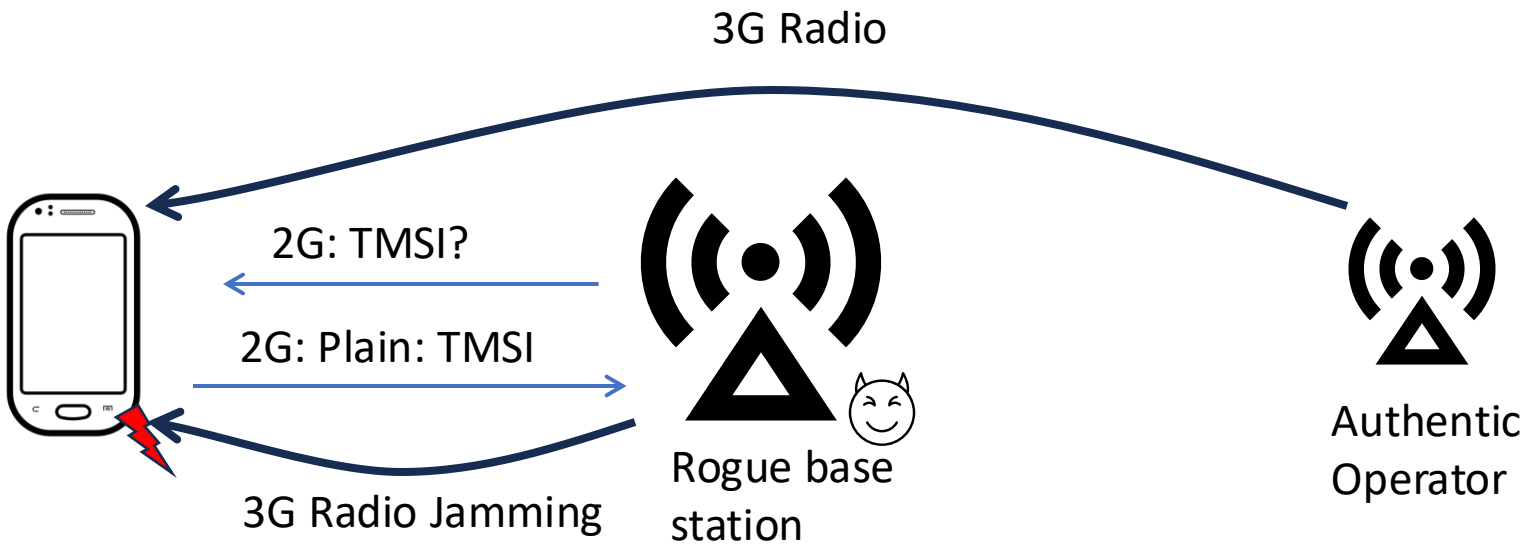
  Protection against re-use of compromised authentication vector

- Addition of a *mandatory cipher mode*, with stronger security, that allows the mobile to verify that encryption is enabled.

(Brief) UMTS (3G) security assessment

- Some improvements with respect to GSM:
  - It fixed the obvious flaws
  - Cryptographic algorithms were publicly discussed
    - Allows for review, ultimately brings stronger security
  - Integrity of the signaling messages is protected
- Quite conservative solution
- Privacy/anonymity of the user not completely protected
  - E.g. still no end-to-end encryption

Attacking 3G: Downgrade to 2G



- Mutual authentication of 3G prohibits IMSI catchers
- However: Attacker can downgrade to 2G by jamming the 3G band
- Reduce attack to the (concurrently running) 2G stack

LTE (4G) Security Improvements

- LTE is more secure than its precedent standards
  - Integrity protection mechanisms are required
  - Unfortunately: many of the new security mechanisms are
    - optional
    - disabled by default

- LTE is completely IP-based
  - End of circuit-based communications
  - Leverages on all the knowledge of IP networks …
  - … but a flaw / 0-day exploit may now affect both networks (Internet and cellular) at the same time

LTE security principles [1]

1. Permanent security association with home network
   - Permanent key stored both in devices and in the home network.
   - It is used to securely derive ephemeral keys
2. New key hierarchy
   - A new local master key enables cryptographic network separation
   - This master key is less exposed (it remains in the core network)
3. Reciprocal authentication mechanisms
   - The mobile phone can also verify the authenticity of the network
   - Additional shared cryptographic material between the terminal and the network allows for the confidentiality and integrity-protection of the transmitted data

[1] Anastasios N. Bikos, Nicolas Sklavos, "LTE/SAE Security Issues on 4G Wireless Networks", IEEE Security & Privacy vol. 11, p. 55-62, 2013

LTE security principles (cont'd)

4. Trusted environment and secure execution in the devices
   - Isolated, even from the OS
   - Device authentication executed within the trusted environment
   - Support for autonomous validation, in which the network is assured of the integrity of authenticated devices. (i.e. the computations were done in an uncompromised trusted environment)
5. DoS protection of the network
   - Restriction in the number of connections
   - Only validated network elements should be allowed in the core network
6. User privacy
   - IMSI should be kept secret both inside the device (from the Apps) and over the air
   - Confidentiality of signaling and user data must be retained
7. Authorization
   - Authorization is required for the connection to core networks and for software integrity

LTE Security in Practice

- LTE more secure than predecessor **as a standard**
- but in practice:
  - Implementation mistakes
  - Configuration flaws
  - Specification/protocol deficiencies

- Researchers have found several flaws in currently deployed LTE networks that can result in:
  - Privacy leaks
  - Denial of service attacks

Practical attacks against Privacy and Availability in 4G/LTE [1]

- Vulnerabilities in 4G **specifications** and networks
  - User Equipment (UE) measurement reports
    - Requests are not authenticated, reports are not encrypted
  - Tracking area update (TAU) procedure
    - Reject messages are not integrity protected
  - Attach procedure
    - Network capabilities are not protected against bidding down attacks
  - Radio Capabilities can be tampered – Bidding Down attack
  - User Plane Data is not integrity protected
- **Implementation** flaws (all vendors)
  - Radio Link Failure (RLF) reports
    - Requests not authenticated, reports are not encrypted

[1] Shaik A, Borgaonkar R, Asokan N, Niemi V, Seifert JP, Practical attacks against privacy and availability in 4G/LTE mobile communication systems, NDSS, 2016. https://arxiv.org/pdf/1510.07563.pdf

Practical attacks against Privacy and Availability in 4G/LTE (cont'd)

- Location leaks
  - Track user locations and movements to much higher levels of granularity than was previously thought possible
  - Social applications used for silent tracking

- DoS attack enables downgrade attacks
  - Deny access to LTE network to LTE-capable device
  - Users limited to less secure 2G/3G networks
  - Silent and persistent attack from the user's point of view

LTEInspector: A Systematic Approach for Adversarial Testing of LTE [1]

- Systematic framework for adversarially analyzing LTE specification to find security and privacy problems
  - Abstract model based on LTE standards
  - Relies on a symbolic model checker and a cryptographic protocol verifier
  - Analyzes three critical procedures: attach, detach and paging
- Findings
  - Uncovered 10 new attacks
  - Identified 9 prior attacks: IMSI-catching DoS, linkability, MitM in 3G and 2G.

[1] Hussain SR, Chowdhury O, Mehnaz S, Bertino E. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE, NDSS, 2018

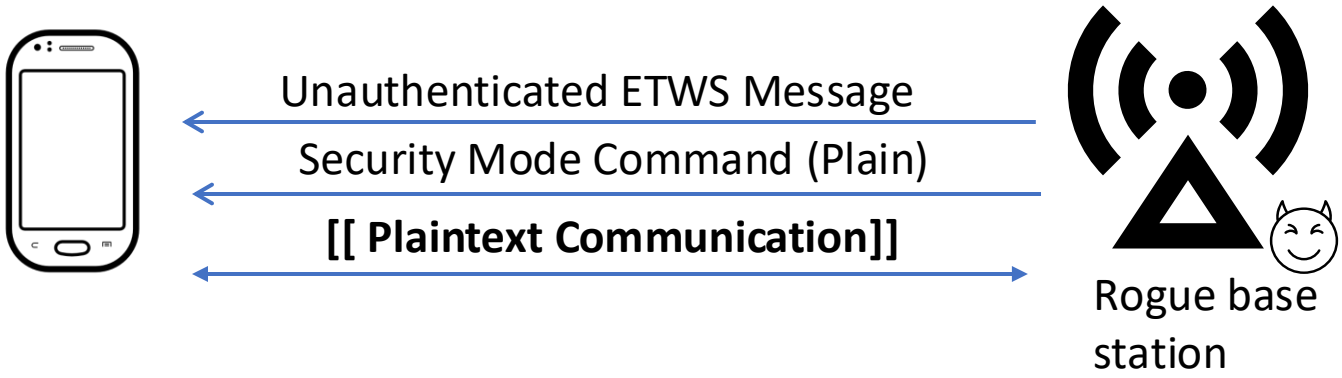
LTEInspector: New attacks Uncovered

Attack	Procedures	Responsible	Notable Impacts
Auth Sync. Failure	Attach	3GPP	DoS
Traceability	Attach	carriers	Coarse-grained location tracking
Numb using auth_reject	Attach	3GPP, smartphones	DoS
Authentication relay	Attach	3GPP	Location spoofing
Paging Channel Hijacking	Paging	3GPP	DoS
Stealthy Kicking-off	Paging	3GPP	DoS, coarse-grained location tracking
Panic	Paging	3GPP	Artificial chaos for terrorist activity
Energy Depletion	Paging	3GPP	Battery depletion/DoS
Linkability	Paging	3GPP	Coarse-grained location tracking
Targeted/Non-targeted Detach	Detach	3GPP	DoS

3GPP is the main standardization body of cellular networks

LTEInspector: Panic Attack

- Adversary can inject fake emergency paging messages to a large number of user equipments
- "**Panic situation**": Drops Authentication and Confidentiality requirements UEs



ETWS: Emergency and Tsunami Warning System

5G Security Features

- **Integrity Protection** for User Plane Function (IP traffic)
- **Radio Capabilities** are sent over a **Secured Channel**
  - Prevents Bidding Down Attacks
- **Additional Authentication Primitives**
  - EAP = Extensible Authentication Protocol
  - Enables interop with e.g. WiFi subscriber management
- Introduce **Subscriber Unique Concealed Identifier** (SUCI) to cryptographically secure the IMSI (SUPI)
- Decoupling the IMSI for paging procedures
  - Instead of IMSI / SUPI a Globally Unique Temporary Identifier (GUTI) is used for paging
  - Mitigation of tracking with ephemeral identifiers

Papers on Analysis of 5G Security and Privacy

- Ijaz Ahmad et al. Overview of 5G Security Challenges and Solutions, IEEE Communications Standards Magazine, March 2018
- R. P. Jover, V. Marojevic. Security and Protocol Exploit Analysis of the 5G Specifications, IEEE Access, Feb. 2019
- D. Basin et al. A Formal Analysis of 5G Authentication. ACM CCS 2018

Protocol-Focused Analysis

Hermes (SEC'24) Synthesizing Formal Models via NLP

- **Approach:**
  - Synthesizing Protocol State machines with ML (consituency parsing, domain-specific language design)
  - **(a)** model-checking on specification to verify security properties
  - **(b)** check if **COTS** baseband respect the model
- **Evaluation:**
  - Perform MC on generated FSM to find deviations, that were not designed
  - Analysis of LTE and 5G - NAS, RRC

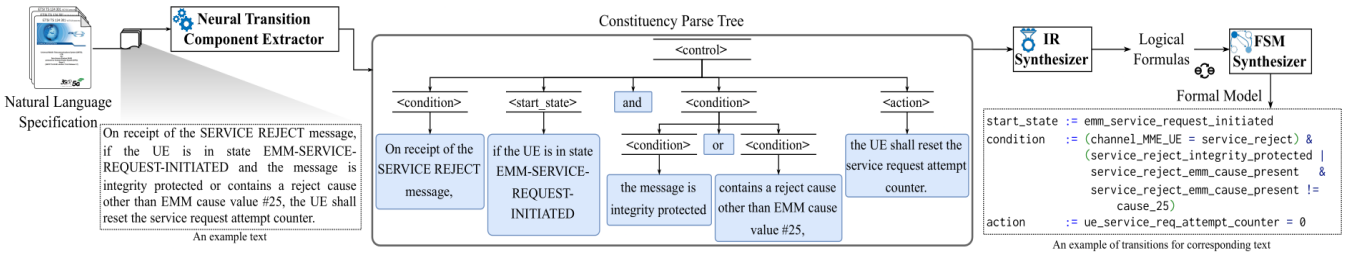
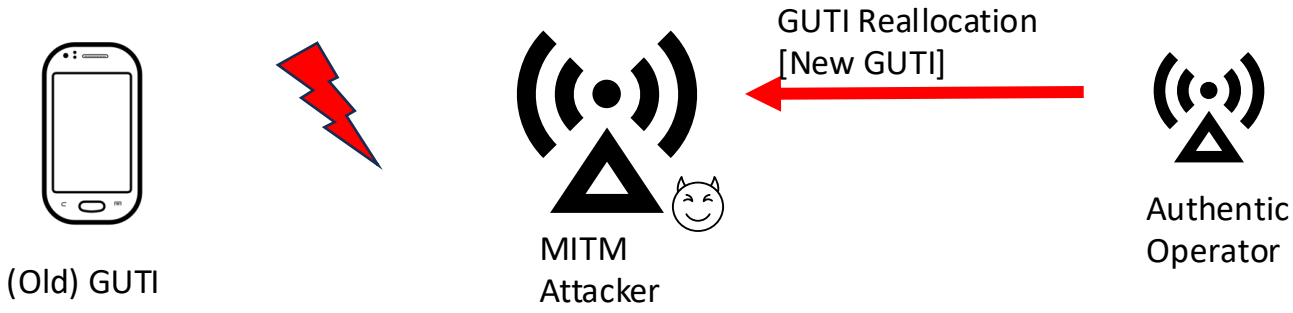


Figure 3: Overview of Hermes.

Protocol-Focused Analysis

"Demystifying Privacy in 5G Stand Alone Networks" (MobiCOM'24)

- Surveys and compares 5G NSA and SA security for real operator networks
- Attack on privacy because of unauthenticated GUTI Reallocation
  - Drop "GUTI Reallocation" message -> identity tracking via "permanent" GUTI



Implementation-Focused Analysis

"BaseComp" (SEC '23) Static Analysis between Specification ↔ Baseband

Approach: Analyze ***integrity protection function***

1. **Compare** the specification graph with the firmware's probabilistically identified graph
2. Perform **symbolic execution** on the extracted function and compare to the specification

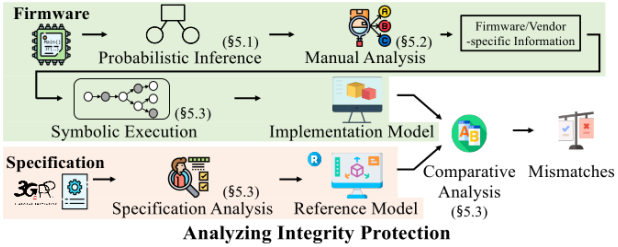
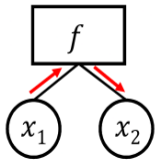


Figure 4: Workflow of BASECOMP.

Evaluation:

1. Finds logic bugs, since the behavior is analyzed
2. Requires a reference model for integrity protection → built manually
3. Manual analysis required (e.g. for symbolic execution)



		$x_1 \xrightarrow{Pos} x_2$	$x_1 \xrightarrow{Neg} x_2$	$x_1 \xrightarrow{Pref} x_2$
x_1	x_2	f_{Pos}	f_{Neg}	f_{Pref}
0	0	0.5	0.5	0.5
0	1	0.5	0.5	p_{Pref}
1	0	p_{Pos}	$1 - p_{Neg}$	$1 - p_{Pref}$
1	1	$1 - p_{Pos}$	p_{Neg}	0.5

Hardware-In-The-Loop Testing

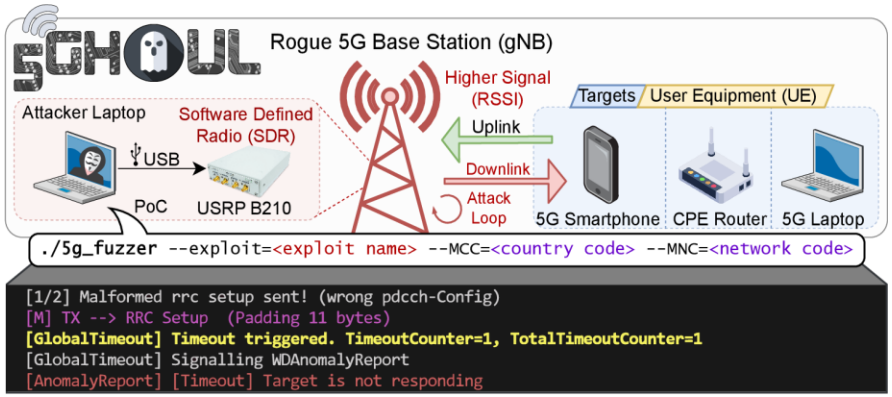
5Ghoul (2024)

Approach:

- Hardcoding the protocol fields for the tested UE (extracted from specification)
- Instrumenting open-source gNB to send messages from a protocol-based fuzzer to the target UE
- UE reply is verified in the open-source gNB implementation

Evaluation:

- Requires to have the gNB running during fuzzing, especially verifying message flows
- Manual (and error-prone) work to implement the protocol fields for fuzzing

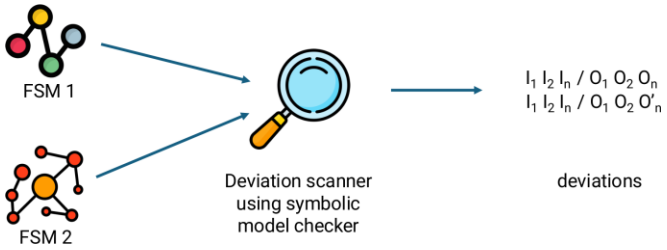


Hardware-In-The-Loop Testing

5GBaseChecker (SEC'24)

Approach:

- **Automata learning** and **differential testing:** construct a state machine (FSM) for a vendor
- FSMs should be used to compare between vendors, regardless of implementation
  - **Reason**: black-box testing: no insight into the baseband given
  - **Advantage**: Flexible use against different vendors without additional effort



Steps for Analysis:

1. **Passive learning** (from traces → bootstrap the FSM)
2. Model refinement (**active learning** to validate and extend)
3. Probe the state machines to find differences between the FSMs
4. (Manual step) Read in the specification to decide what the correct behavior is

Cellular Security Landscape

3G: Public Security Algorithms
- Security improvements on radio link
- Lacking security in core network

➔ *Hardening the radio interface*

5G: Hardening and Integration
- Introduce SUCI for concealment
- Introduce EAP for Auth

➔ *Focus on Implementation Flaws*

2G: First Digital Cellular System
- Insufficient Security
- Lacking mutual auth
- "Easy" USIM cloning

➔ *"Security by obscurity"*

4G: IP-based communication
- Lacking UPF authentication
- ETWS security downgrade

➔ *Unify network layer*

Wireless Network Security - Outline

- Brief reminder of security and crypto principles
- Challenges in Wireless Networks Security
- Cellular Network Security
- WiFi 802.11 Network Security
- Wireless Pairing & Bluetooth

IEEE 802.11 “WiFi” Security

- **WEP**, **W**ired **E**quivalent **P**rivacy
  - introduced in 1997
  - part of the original IEEE 802.11 standard
- **WPA**, **W**i-Fi **P**rotected **A**ccess
  - introduced in 2003
  - part of the **draft** of IEEE 802.11**i** standard
- **WPA2**
  - introduced in 2004
  - part of the **final** IEEE 802.11**i** standard
- **WPA3**
  - introduced in 2018
- WPA4
  - TBD

Note: IEEE 802.11**i** is the standard (part of IEEE 802.11) related to security

WEP

Stands for **W**ired **E**quivalent **P**rivacy

- Goals:
  - make WiFi networks *at least as secure as a wired LAN* (that has no particular protection mechanisms)
  - WEP was never intended to achieve strong security
- Services:
  - access control to the network
  - message confidentiality
  - message integrity

WEP (cont'd)

- before **association**, the station needs to **authenticate** itself to the AP

- authentication is based on a simple challenge-response protocol:

  STA → AP: authenticate request
  AP → STA: authenticate challenge (r) // r is 128 bits long
  STA → AP: authenticate response ($e_K(r)$) // signature of r
  AP → STA: authenticate success/failure

- if **authentication** fails, no association is possible

- once **authenticated**, the station can send an association request and join the network

WEP Weaknesses

WEP is **insecure** and should not be used.

1. authentication is one-way only
   - AP is not authenticated to the device (c.f. 2G in cellular)
   - device is at risk to associate to a rogue AP

2. the same shared secret key is used for authentication and encryption
   - weaknesses in any of the two protocols can be used to break the key

3. no session key is established during authentication
   - access control is not continuous
   - once a device has authenticated and associated to the AP, an attacker can send messages using the MAC address of the station
   - correctly encrypted messages cannot be produced by the attacker, but replay of messages is possible

WEP Weaknesses (cont'd)

4. Weak crypto standards

   (this is the main WEP weakness)

   - 64-bit security that uses 40-bit key + 24-bit Initialization Vector
   - It comes from the US export restriction on cryptography
   - Has been extended to 128 bits, but still with an IV of 24 bits

   - Encryption uses RC4, a **stream** cipher
   - Crypto reminder on stream ciphers:
     - never reuse the IV between sessions !!!
     - or you can have ***related-key attacks*** *(*next slide)
   - IV changes for every message
   - Yet, on a busy network, the IV of 24 bits is not long enough

     $2^{24} \times 1500$ Bytes ≈ 25 GB → forced to reuse after that
   - By collecting enough messages, a ***key-recovery attack*** is possible [1]

[1] Fluhrer, S., Mantin, I., and A. Shamir, Weaknesses in the Key Scheduling of RC4

Related-Key attack - details

- The client encrypts some plaintext P_1 with RC4, with a key K and some IV:

$$C_1 = P_1 \oplus \text{RC4}(K, IV)$$

- The key remains the same, and if the IV is reused, the encryption of another plaintext P_2 yields:

$$C_2 = P_2 \oplus \text{RC4}(K, IV)$$

- The attacker sees the ciphers C_1, C_2 . He cannot directly decrypt, but he can get the XOR of the two plaintexts :

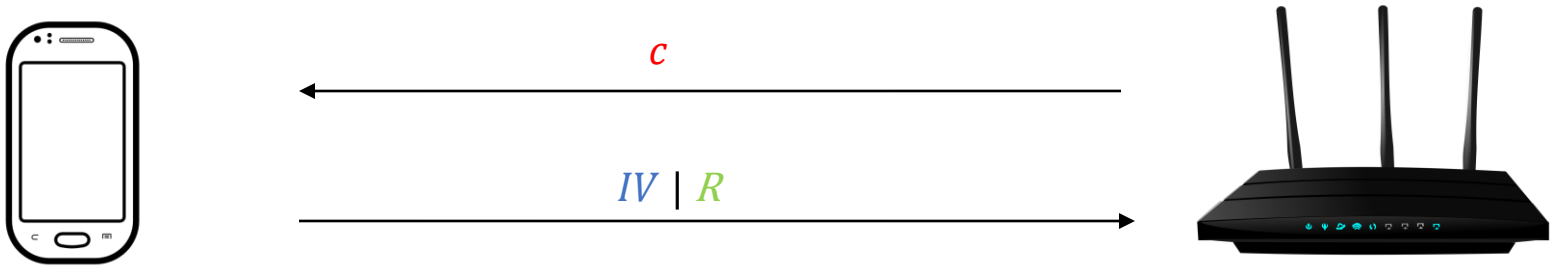
$$C_1 \oplus C_2 = P_1 \oplus P_2$$ <- this is catastrophic

Which yields information about the plaintexts (e.g. it allows for *known-plaintext attacks*).

WEP Weaknesses (cont'd)

5. Broken authentication protocol

- c is a random challenge, IV is chosen by the client. R is $c \oplus \mathrm{RC4}(IV, \text{Wireless Network Key})$, and proves that the client knows the Wireless Network Key.



Attack:

- An attacker eavesdrops, and sees c, IV, and R in plaintext
- He computes $R \oplus c$ which is $\mathrm{RC4}(IV, \text{Wireless Network Key})$, the proof-of-knowledge of the Wireless Network Key
- Then, he authenticates himself easily :

AP → attacker: c'

attacker → AP: IV | $c' \oplus R \oplus c$

=> Valid authentication, can associate without knowing the Wireless Network Key

Cracking WEP

- Linux + an external, monitoring-capable antenna
- Most popular antenna for «playing around»:
  ALFA AWUS036h

Five steps:

1. Put the interface in monitoring
2. Choose and eavesdrop on a WEP WiFi LAN for encrypted packets
3. … wait (for IV repetitions), or…
4. Spam the network with DEAUTH packets that forces devices to reconnect (this generates legitimate traffic on the network)
5. Let the automated tool recover the key

=> Takes less than a minute on a modern computer

WEP – Lesson learnt

Engineering security protocols is difficult:

- One can combine otherwise strong building blocks in a **wrong way** and obtain an **insecure system** at the end
  - Example:
    - stream ciphers alone are OK
    - challenge-response protocols for entity authentication are OK
    - but they could be combined in an insecure way (Weakness n°5)
- Don't do it alone
  - functional properties can be tested
  - **security is a non-functional property**

    → extremely difficult to tell if a system is secure or not
- Using an expert in the design phase pays out

  → Fixing the system after deployment will be much more expensive
- Experts cannot guarantee that a system is 100% secure, but know the pitfalls
- E.g., they know the details of crypto algorithms (e.g. RC4 weak keys)

IEEE 802.11 “WiFi” Security

- **WEP**, **W**ired **E**quivalent **P**rivacy
  - introduced in 1997
  - part of the original IEEE 802.11 standard
- **WPA**, **W**i-Fi **P**rotected **A**ccess
  - introduced in 2003
  - part of the **draft** of IEEE 802.11**i** standard
- **WPA2**
  - introduced in 2004
  - Is part of the **final** IEEE 802.11**i** standard
- **WPA3**
  - introduced in 2018
- WPA4
  - TBD

Note: IEEE 802.11**i** is the standard (part of IEEE 802.11) related to security

WPA/WPA2

Main novelties in WPA/WPA2 w.r.t to WEP:

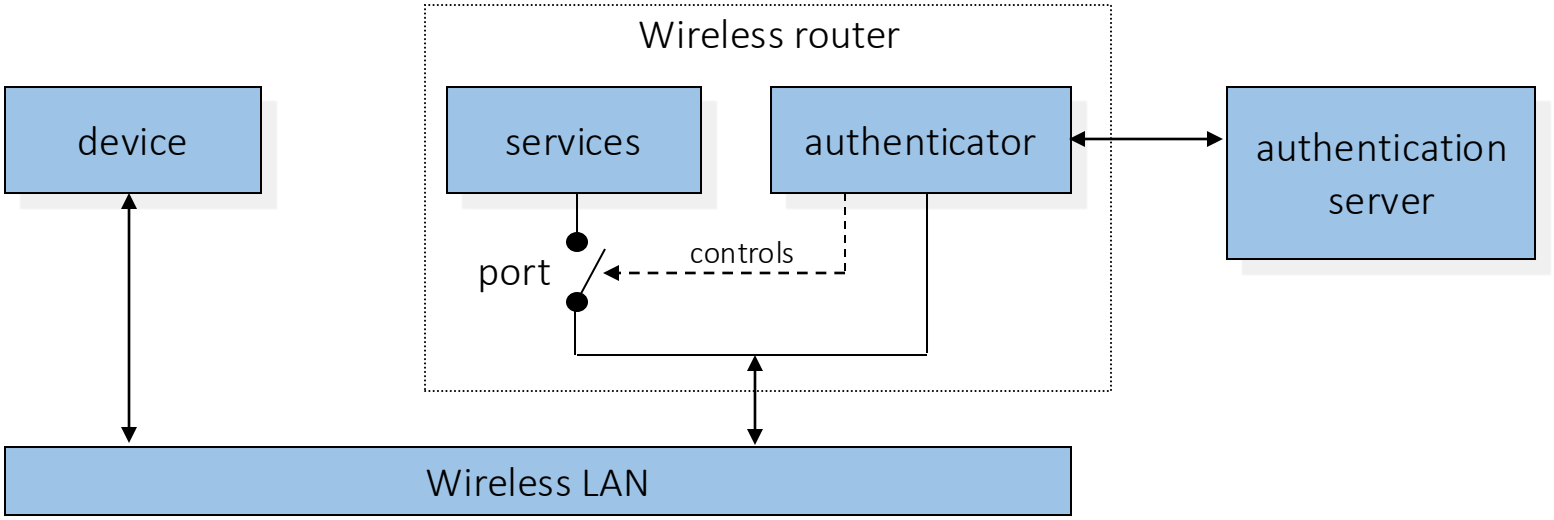
- **Encryption** is replaced by either:
  - **TKIP:** Now insecure + deprecated,
    → (Ugly) temporary solution replacing WEP while using the same hardware
  - or **AES:** Strong
    → But required changing **all** network cards…
    → This is the standard now.

- Flexible **authentication:** EAP (Extensible Authentication Protocol)
  - Multiple authentication mechanisms available
  - Successful authentication results in a per-session **shared session key**
  - Can interoperate with 5G for authentication

- **Integrity protection** is improved, notably by sequence counters and Message Authentication Codes

- Different functions (encryption, integrity) **use different keys** derived from the session key using a one-way function

WPA/WPA2 authentication model



- the authenticator controls access to the services (controls the state of a port)
- the authentication server authorizes access to the services

- the device **authenticates itself** to the authentication server
- if the authentication is successful, the authentication server instructs the authenticator to **switch the port on**
- the authentication server informs the device that access is allowed, and sends a **session key** shared between the mobile device and the authentication server

WPA/WPA2 authentication

Three main supported modes :

1. **PSK**, Pre-Shared Key
   - All users share a (usually human-readable) key
   - Mostly deployed in homes, small companies
   - The authentication server is very simple, and part of the wireless router
   - Simple, but not flexible

2. **Entreprise**
   - The authentication server is an external RADIUS server
   - Every user/device has a different key
   - Complex setup, flexible

3. **WPS (Wi-Fi Protected Setup)**
   - This is a way to bootstrap/set up PSK, but with a very secure key
   - Aims to prevent bruteforce attack on PSK with weak keys (e.g., «password123»)

WPA/WPA2 weaknesses

1. Weak passwords (this is not specific to WPA/WPA2)
   - Use strong passwords…
   - But also don't use a SSID that is common, as there exist downloadable rainbow tables for them [Pairwise Master Key with PBKDF2(`passphrase || SSID`)]

2. TKIP has been broken
   - First attacks in 2008 [1], latest (and strongest) in 2013 [2].
   - Attacker can inject data (usable payload, or malicious Javascript on webpages), hijack TCP connections, decrypt (some) messages, without knowing/learning the network's key

3. WPS PIN Recovery [3]
   - The 8-digit PIN is "burned" in the device, and usually written on it
   - The protocol uses two *different* messages to transmit the PIN (split in two parts)
   - The adversary sees if the authentication failed after the first *or* the second message => The PIN can be bruteforced easily in ~ 2 hours

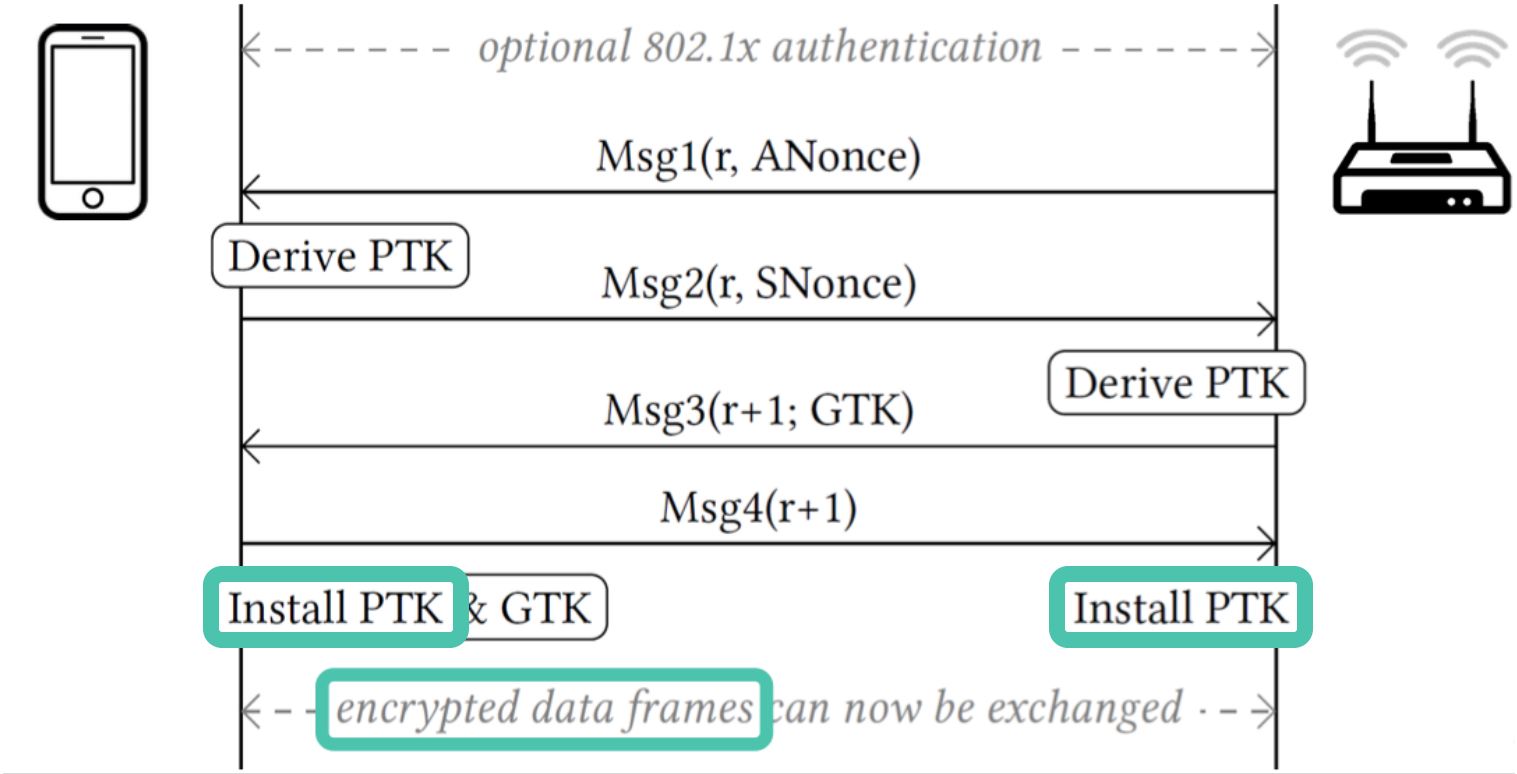
[1] http://dl.aircrack-ng.org/breakingwepandwpa.pdf
[2] https://lirias.kuleuven.be/bitstream/123456789/401042/1/wpatkip.pdf
[3] Stefan Viehböck, Brute Forcing Wi-Fi Protected Setup

WPA2 4-Way Handshake

- Used to connect to any protected WiFi network
- Two main purposes:
  - Mutual authentication
  - Negotiate fresh pairwise temporal key (PTK)
- **Appeared** to be secure:
  - No attacks in over a decade (apart from password guessing)
  - Proven that negotiated key (PTK) is secret [1]
  - Encryption protocol formally proven secure [2]

[1] C. He, M. Sundararajan, A. Datta, A. Derek, and J. Mitchell, A Modular Correctness Proof of IEEE 802.11i and TLS, CCS, 2005.
[2] J. Jonsson, On the security of CTR+ CBC-MAC, SAC, 2002.

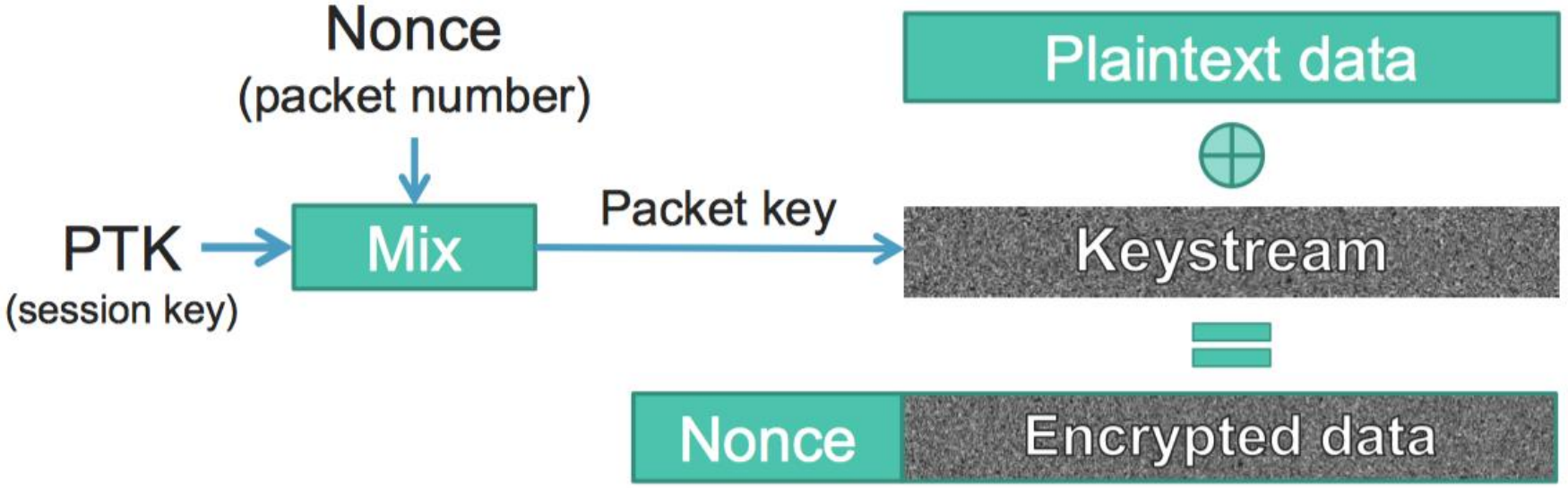
WPA2 4-Way Handshake (simplified)



PTK = Combine (shared secret, ANonce, SNonce)

GTK = Group Temporal Key (broadcast & multicast)

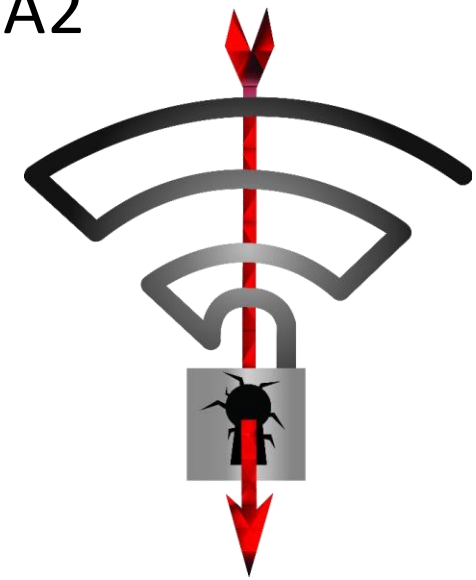
WPA2 Frame Encryption (simplified)



→Nonce reuse implies keystream reuse (in all WPA2 ciphers)
Reuse occurs after 2^48 packets (over 281 trillion packets)

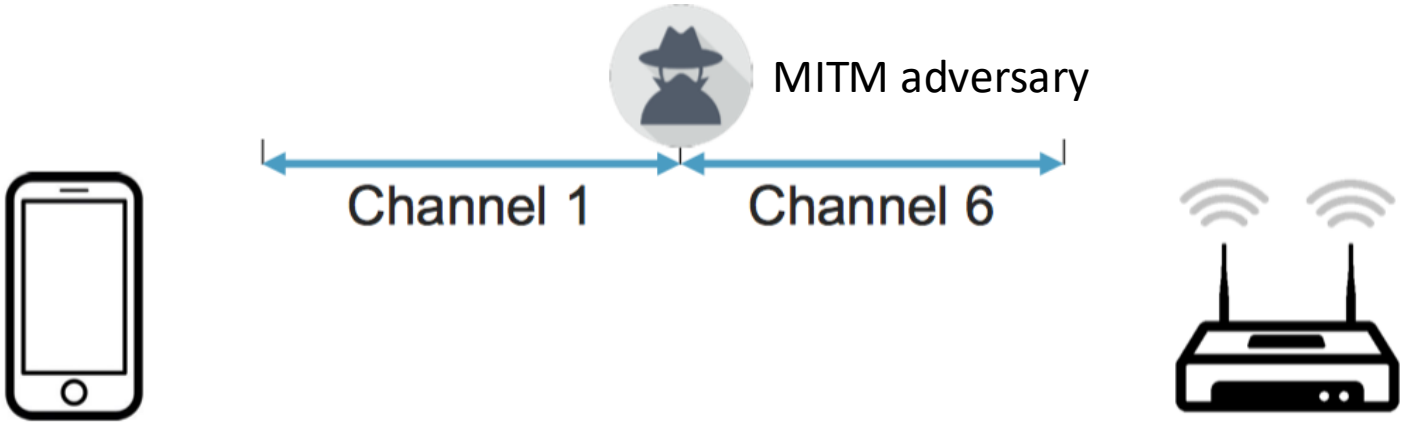
WPA2 Key Reinstallation Attacks [1]

- Exploit of WPA2 4-way handshake
- Weaknesses in the WiFi standard itself
  - Any correct implementation of WPA2 is likely affected
- https://www.krackattacks.com/



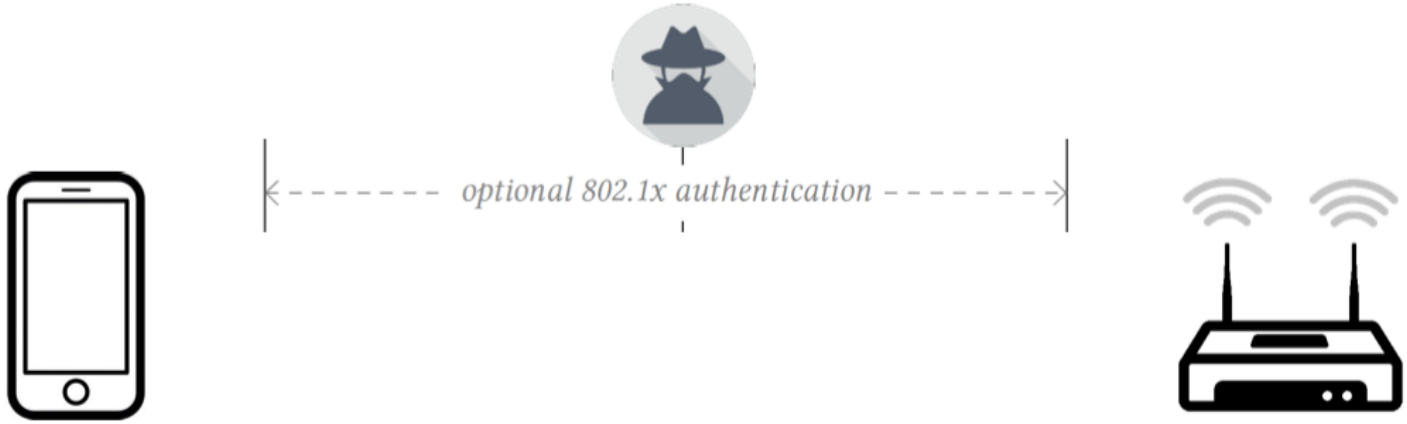
[1] Vanhoef, M. and Piessens, F., Key reinstallation attacks: Forcing nonce reuse in WPA2, CCS, 2017

Key Reinstallation Attack (1)



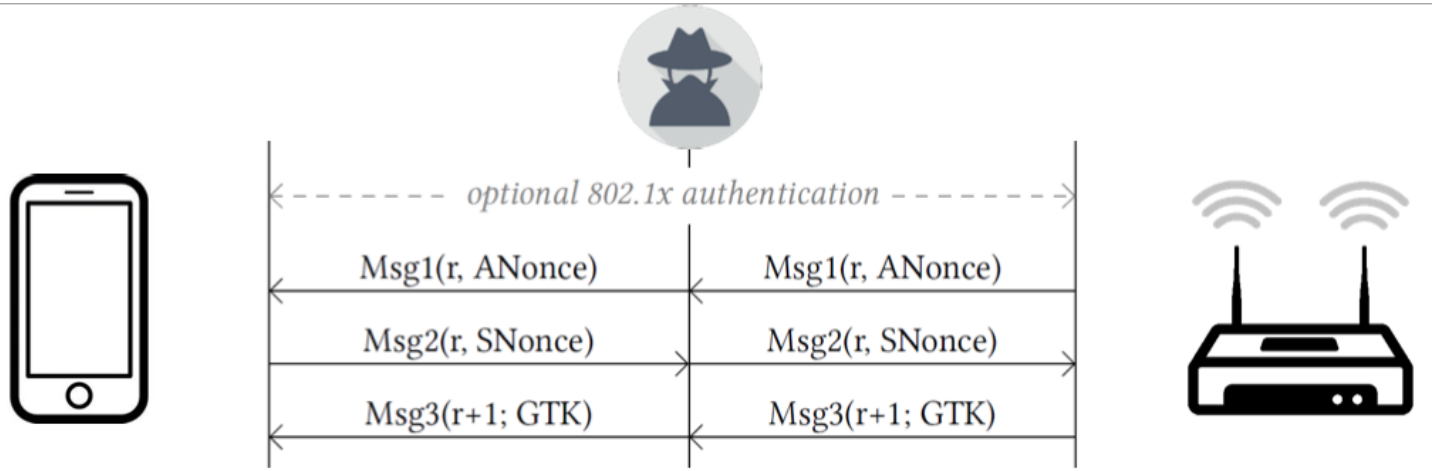
- Adversary gains MITM position to manipulate messages exchanged between client and AP
  - E.g., clone access point on a different channel
  - Delay or block messages
- Adversary cannot decrypt messages, just manipulate messages

Key Reinstallation Attack (2)



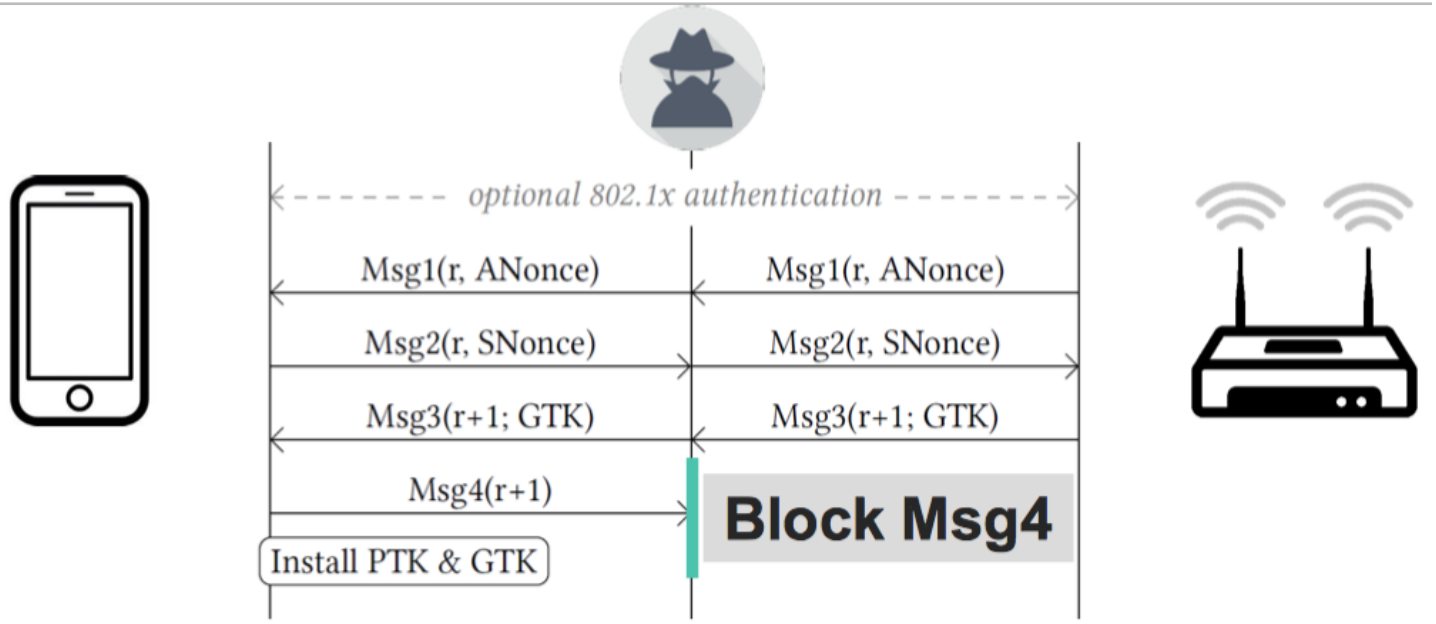
- If 802.1x authentication is used, adversary just forwards the messages

Key Reinstallation Attack (3)



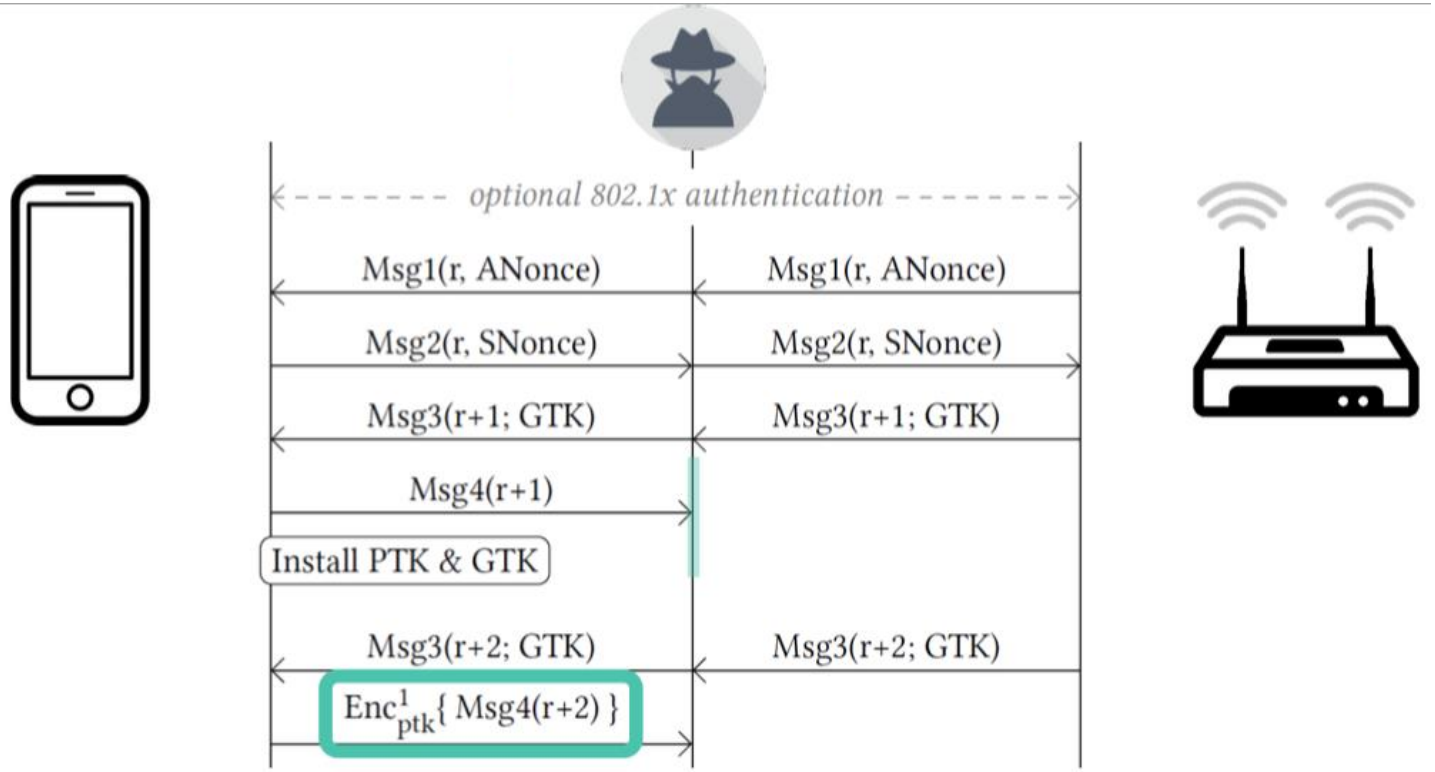
- Adversary forwards the first three messages of the exchange

Key Reinstallation Attack (4)



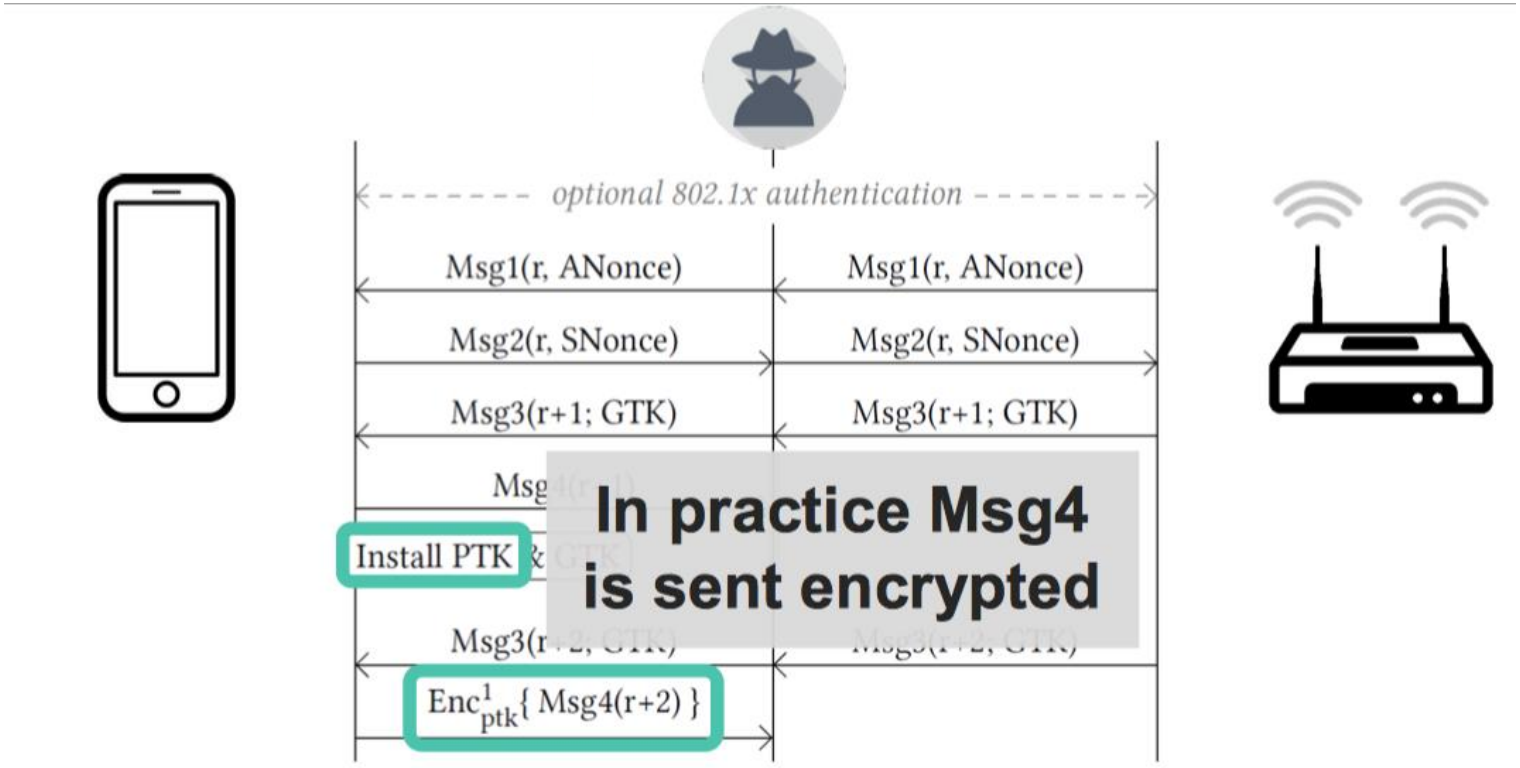
- Adversary **blocks message 4** from the client
- Client assumes that the handshake has completed and proceeds to **install the key PTK**; client is ready to encrypt data

Key Reinstallation Attack (5)



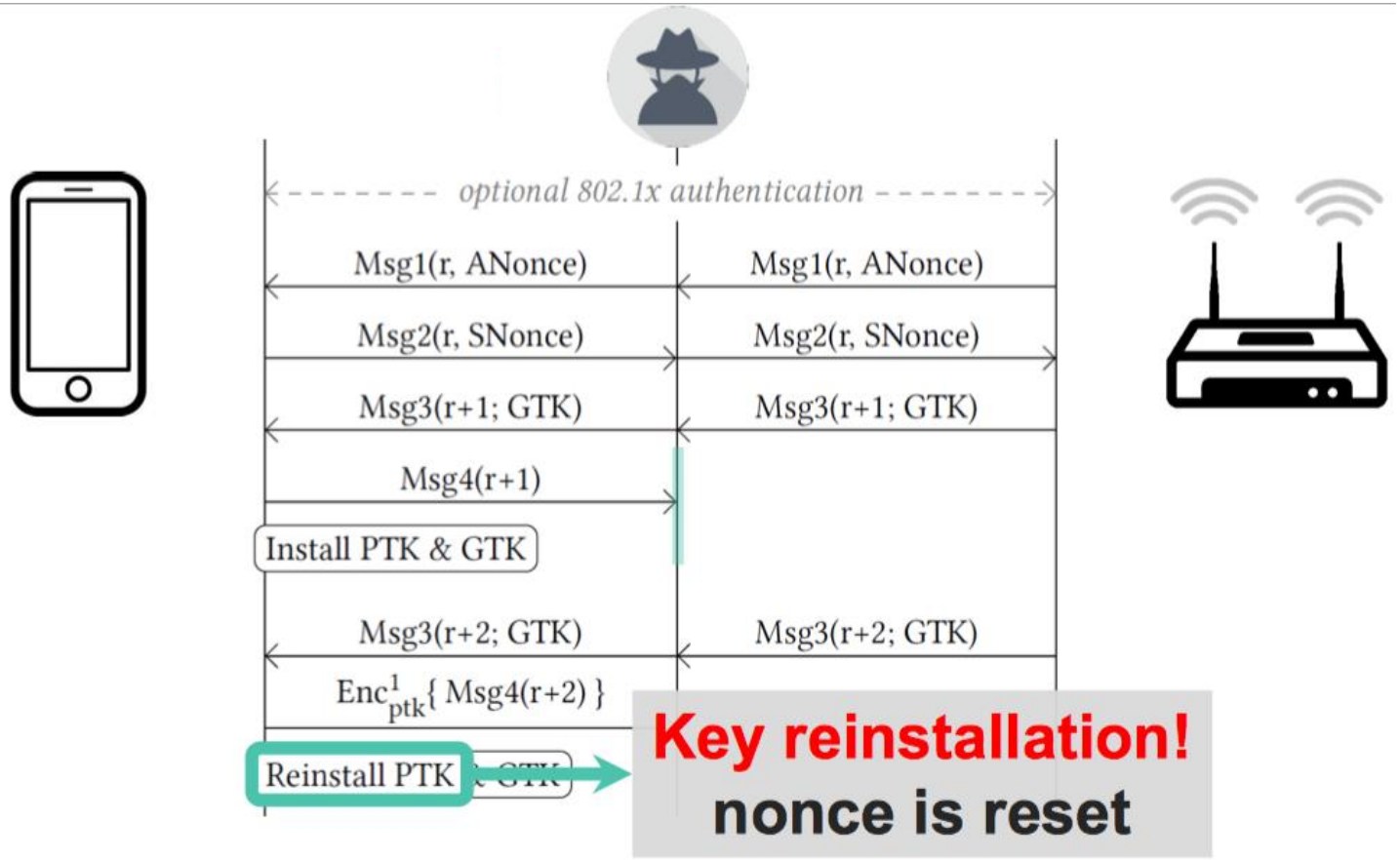
- Access point assumes that message 3 or client's message 4 got dropped, proceeds to retransmit message 3
- Client assumes previous message 4 got lost, proceeds to send a **new message 4**

Key Reinstallation Attack (6)



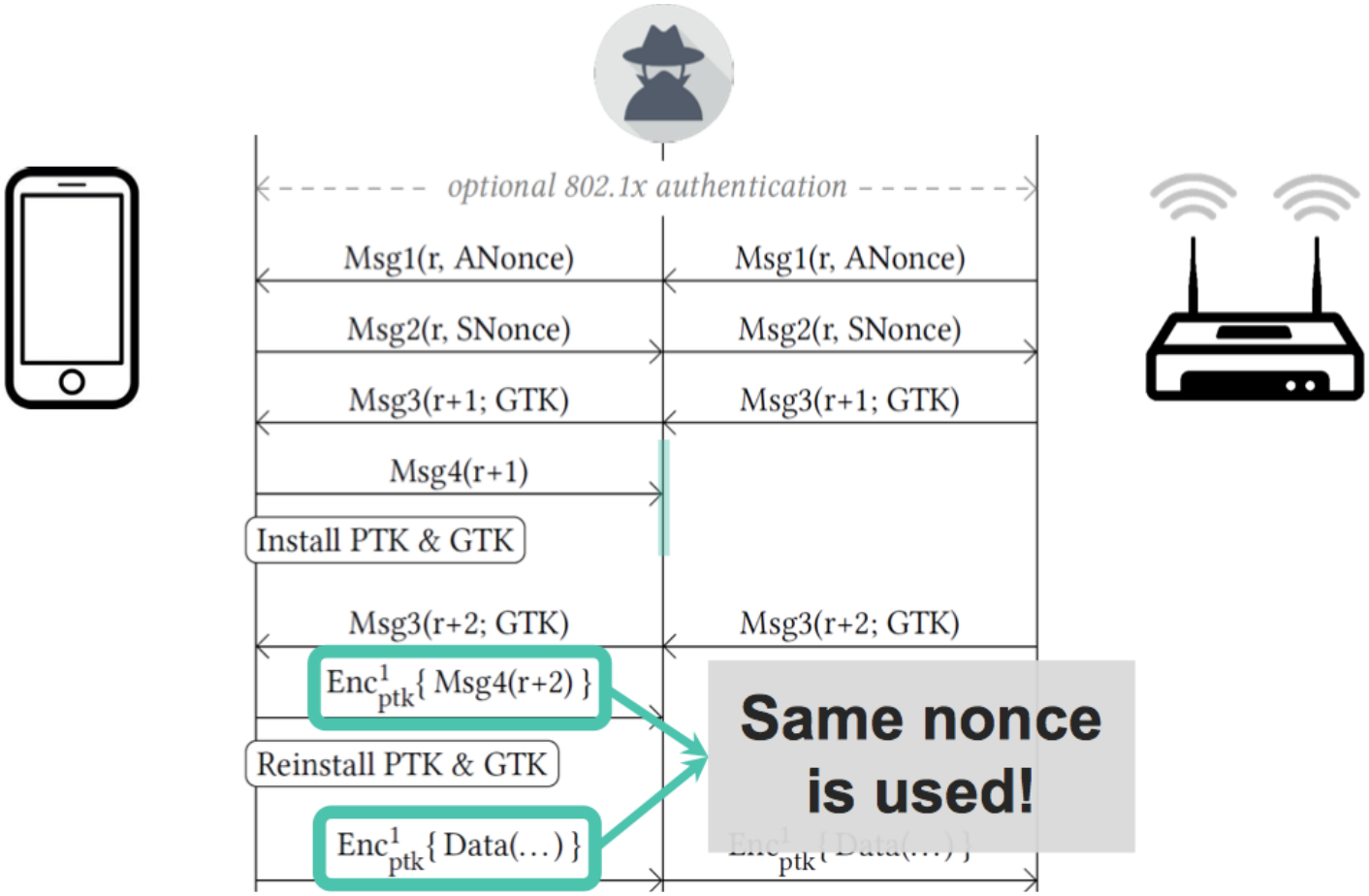
- Standards state that normally the **new message 4 should not be encrypted** with PTK, **but in practice it is!**

Key Reinstallation Attack (7)



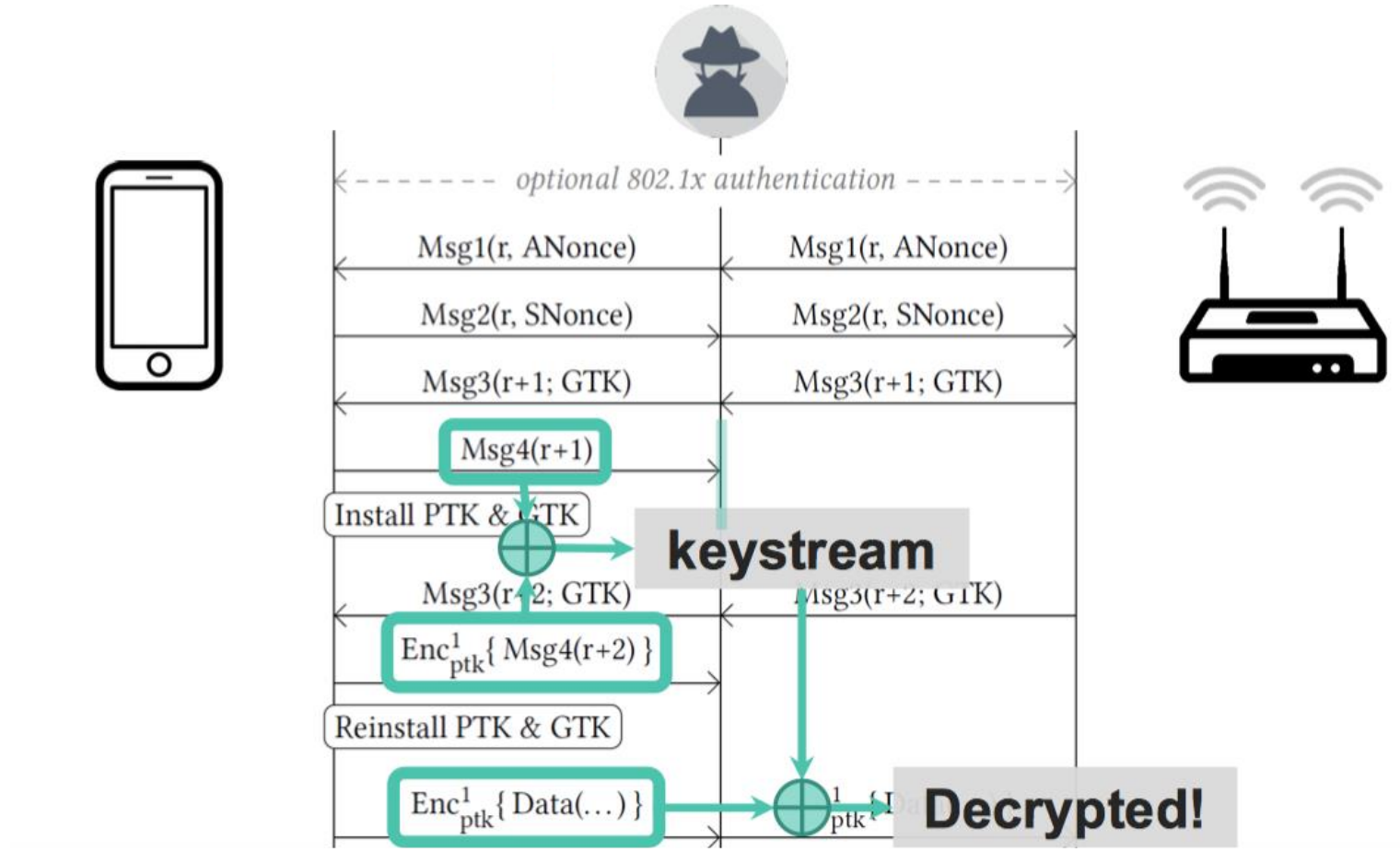
- The standards stipulates that Msg3 **triggers** key (re)installation
- Same key **PTK is reinstalled**, nonce is reset to 0

Key Reinstallation Attack (8)



- New messages will be encrypted with the same nonce previously used

Key Reinstallation Attack (9)



- Adversary can obtain--n bytes of the keystream (>80 bytes) and **start decrypting messages**

Key Reinstallation Attack: Impact

- Nonce re-use = decrypt messages sent from client
- Replay frames towards victim

Evolution of wireless security in WiFi

- **WEP**: Flawed Algorithms / Flawed Protocol
  - Stations can read each other's traffic (no PTK)
- **WPA**: Flawed Algorithms / Flexible Protocol
  - Introducing PTK but Algorithms still too weak
  - Only GTK-encrypted messages are visible for all stations
- **WPA2**: Secure Algorithms / Flexible Protocol / Impl. Flaws
  - KRACK: Exploiting inconsistencies in specification and implementation.
- **WPA3**: Forward Secrecy in case of leaked secrets
  - Stronger Encryption, Rate Limiting

Parallel to Cellular security:
➔ Shifting from Protocol via Algorithms to Implementations

Wireless Network Security - Outline

- Brief reminder of security and crypto principles
- Challenges in Wireless Networks Security
- Cellular Network Security
- WiFi 802.11 Network Security
- Wireless Pairing & Bluetooth

Device Pairing : Problem

- In some settings, there is no central authority distributing the keys
  - Wireless ad-hoc networks
  - Bluetooth devices
  - Internet of Things (in some cases)

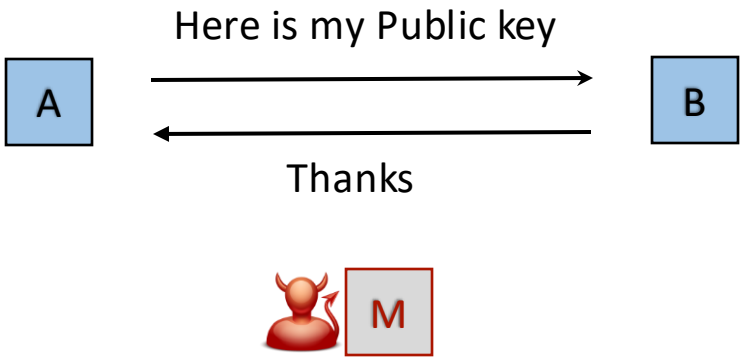
- Devices have to pair autonomously: **set up keys for**
  - **Encryption**
  - **Message Authentication**

- Key establishment is easy over an **authenticated** channel
  - Public-key cryptography (e.g. Diffie-Hellman)
  - Symmetric key techniques *(under special assumptions)*

→ But it is rarely the case in practice

Device Pairing : Problem

- Given a pair of wireless devices, how do they establish a secret key in the presence of an adversary either
  - passive
  - or active – MITM attack ? (MITM: « man-in-the-middle »)



- We assume the devices have no preloaded keys / credentials as for WEP/WPA/WPA2-PSK
- (e.g., two mobile phones, a phone and a printer, ...)

Diffie-Hellman Protocol

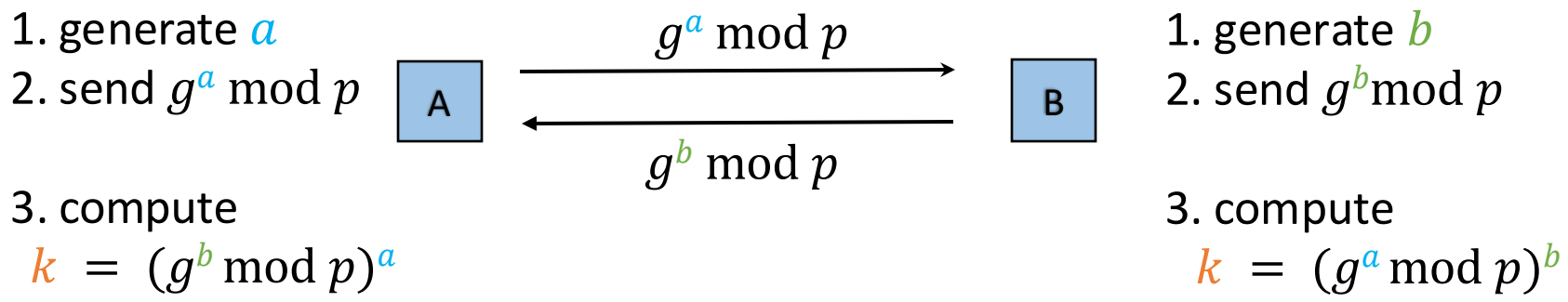
The DH protocol enables secret key establishment in cleartext :

Based on the hardness of the *Computational Diffie-Hellman problem*:

- Given a prime p, a generator g of Z_p^* and elements $g^a \bmod p$ and $g^b \bmod p$, it is computationally difficult to find $g^{ab} \bmod p$.

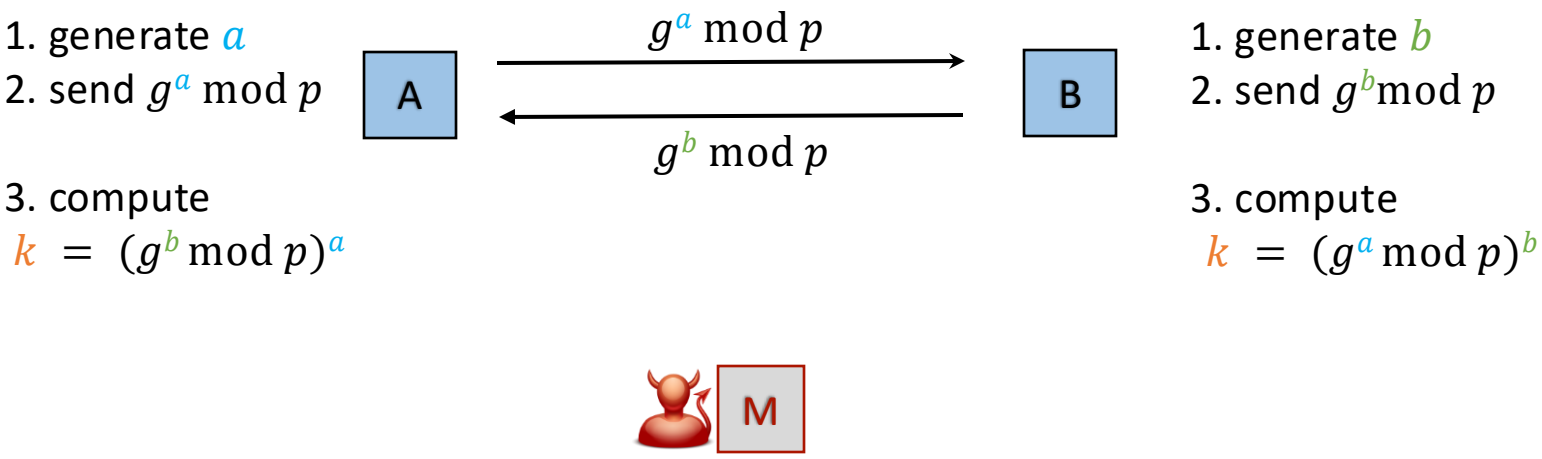
Itself based on the hardness of the Discrete Logarithm problem :

- Given $g^x \bmod p$ it is computationally difficuly to find x.



Diffie-Hellman Protocol (cont'd)

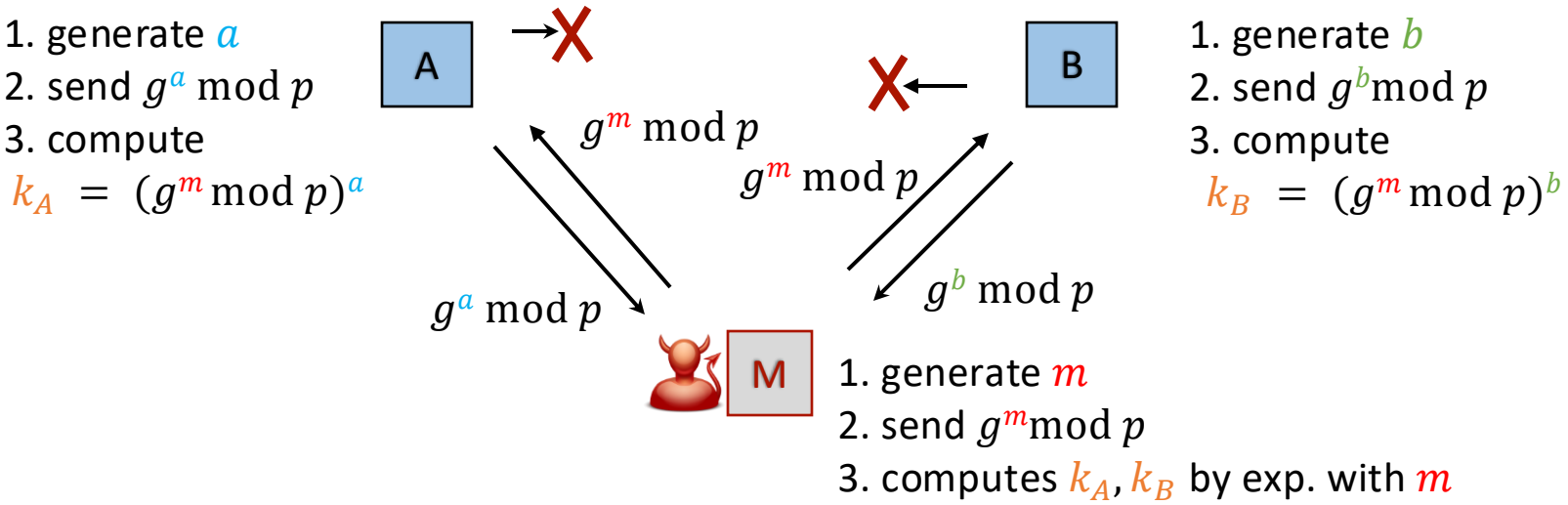
The DH protocol enables secret key establishment in cleartext :



- DH fully resists passive attackers (eavesdropping only).
- DH is not secure against active attackers (MITM attacks).

Diffie-Hellman Protocol (cont'd)

DH is not secure against active attackers (MITM attacks) :



- Hence, messages need to be **authenticated** (which violates our requirement that devices have no preloaded keys)
- **Or**, there need to exist a procedure to verify if the keys are identical

Device Pairing Examples

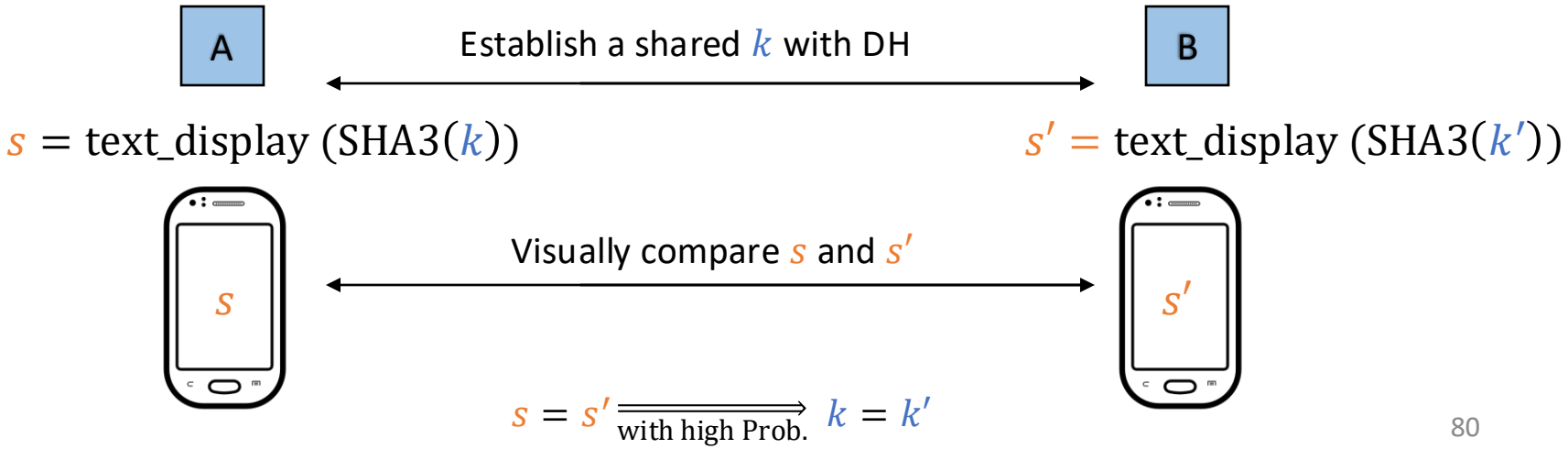
- [1993] Maher: Short String Comparison
- [2005] McCune et al: Seeing is Believing
- [2005] Goodrich et al: Loud and Clear
- [2006] Capkun, Cagalj, Hubaux: Integrity Regions

Device Pairing: Short String Comparison

Maher, 93, US patent, Gehrmann et al 01,03,04, (MANA I, II, III)

Steps:

- Establish key k using DH
- If an attacker is performing a MITM, one user will have $k' \neq k$
- Hash the key and display it on both devices
- Compare the displayed values (160 bits = 20 characters)

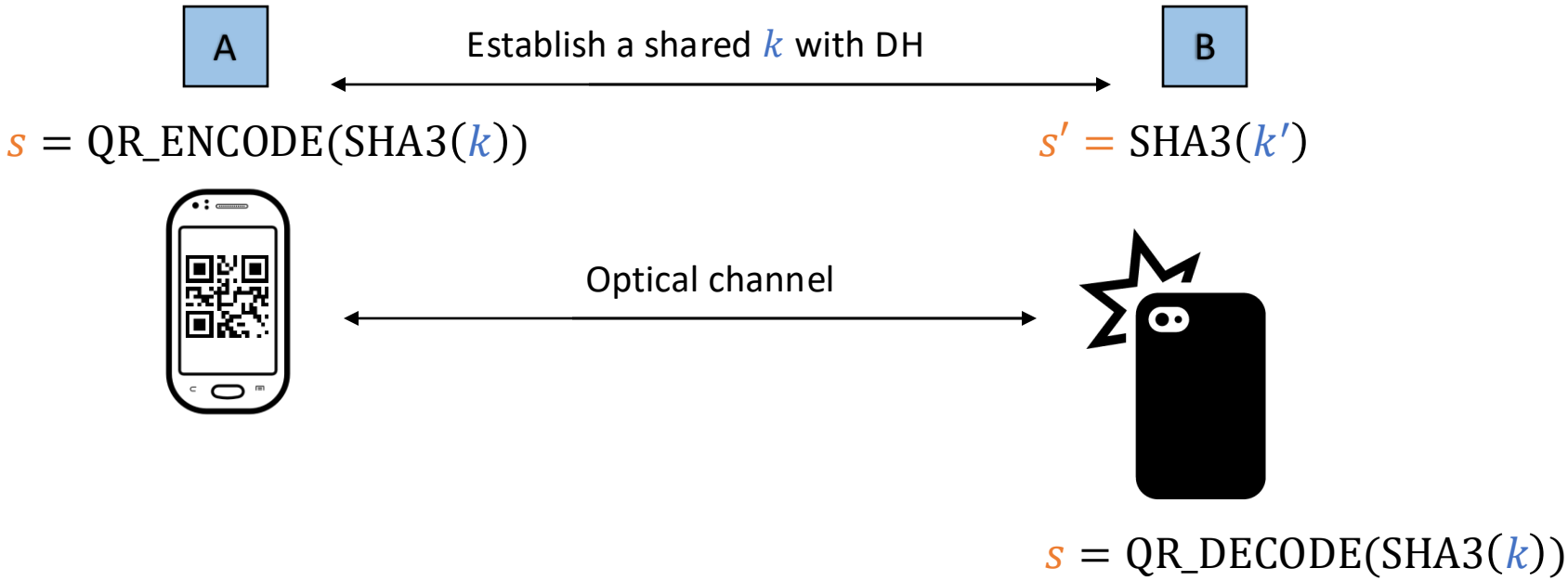


Device Pairing: Seeing is Believing

McCune et al. 05

Idea: Send the public witness of the key over an optical channel.

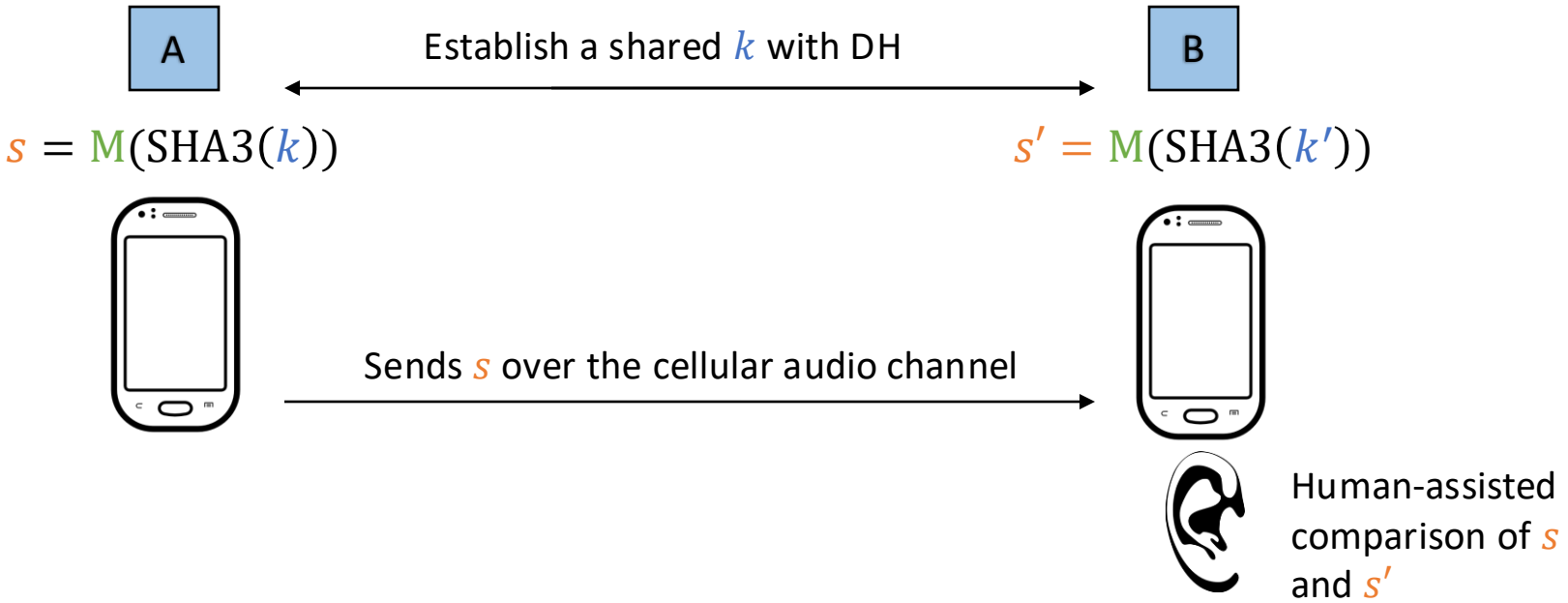
It uses the optical channel (in which it is impossible to perform a MitM attack steathily) to validate the key exchange



Device Pairing: Loud and Clear

Goodrich et al. 05

Idea : Human-assisted string comparison using voice communication



M is a public, pre-defined mapping from bytes to audio (English words)

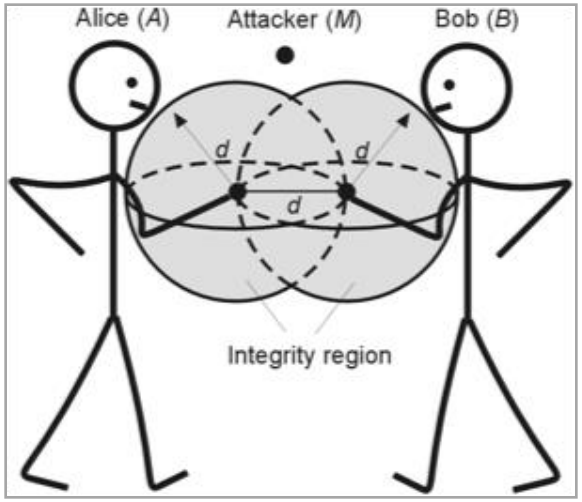
Similar: on-line authentication (e.g., for Secure VOIP applications)
http://zfoneproject.com/

Device Pairing: Integrity Regions

Capkun, Cagalj, Hubaux 06

Idea:

- Establish key k using DH
- Authenticate k using distance-bounding protocols



- Distance-bounding protocols are based on a **proximity proof** :

  → proves that devices are at most at distance *X* from each others
- Assumption: "close-by devices are considered friendly"

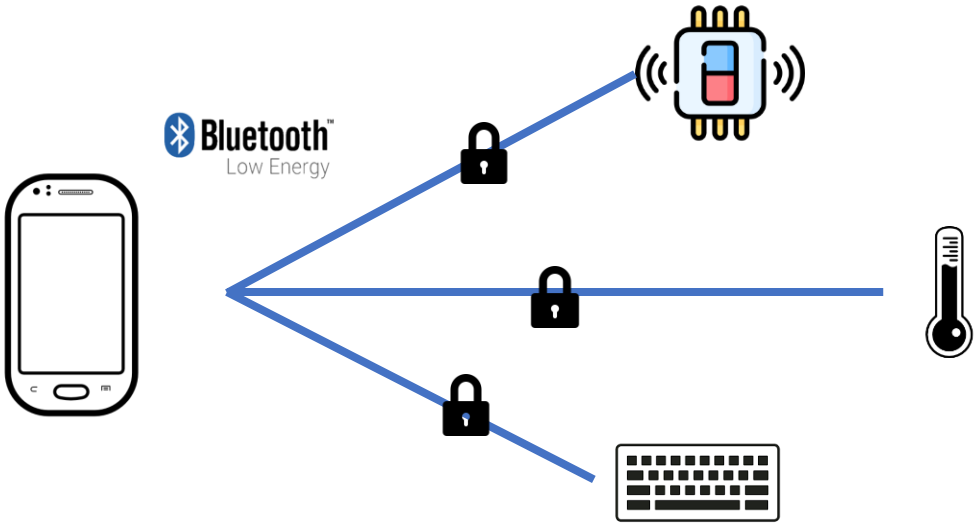
These ideas were later used to launch a start-up company: https://www.3db-access.com

Device Pairing: Conclusion

- Diffie-Hellman can be "protected" against MITM attacks
  - MITM cannot be avoided, but can be detected
  - … **without previously established keys**/certificates.
  - … but assuming a side **verification channel**

- Lots of approaches exploiting different side channels:
  - physical contact
  - string comparison (voice communication)
  - image comparison (hash visualization)
  - distance bounding (physical presence verification)

Case Study: Bluetooth LE



- **Pairing**: establishing an encrypted connection
- **Bonding**: Store information of pairing process to not repeat pairing on next connection

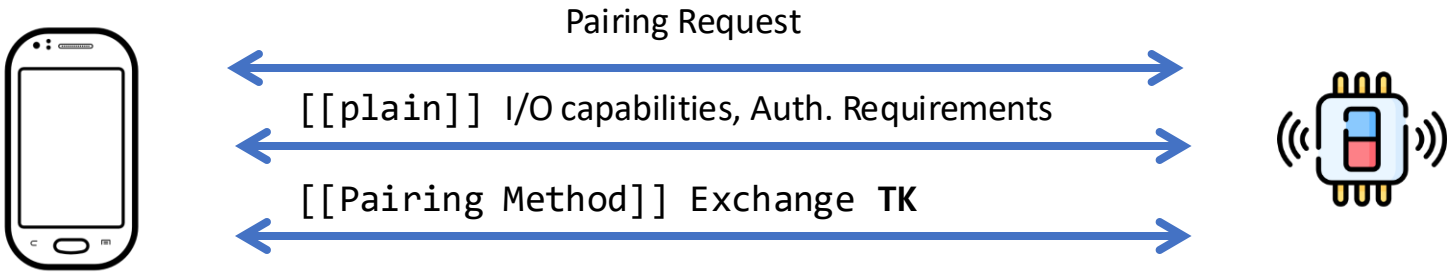
BLE Shared Keys

- **TK (Temporary Key)**
  - Used in pairing procedure
  - Determined by the pairing algorithm
  - It is used to calculate the STK
- **STK (Short Term Key)**
  - Used as encryption key on first connection pairing
  - Consists of **`[TK, Srand, Mrand]:`**

$$\boldsymbol{STK} = \mathrm{E}_{\mathrm{tk}}(\boldsymbol{Srand} \,||\boldsymbol{Mrand})$$

  - ***Srand/Mrand:*** random number from slave/master
- **LTK (Long Term Key)**
  - Used once the initial pairing procedure is complet

BLE Pairing Methods



BLE 4.1

- **Just Works:** Set TK to 0
  - Can be brute-forced or sniffed
  - Used for low-power no I/O devices
- **Out-of-Band:** Use a different wireless technology
  - E.g. NFC for proximity, mitigating a MITM attacker
- **Passkey Entry:** 6 digit number exchanged by devices
  - E.g. LCD shows 123456 and the other device's keypad is used to confirm the numbers
  - Can protect against passive eavesdropping
- **Numeric Comparison (4.2): Like JustWorks with extra verification**
  - **Both devices independently generate 6-digit confirmation numbers**
  - **User manually verifies the digits matching to assure no MITM attack**