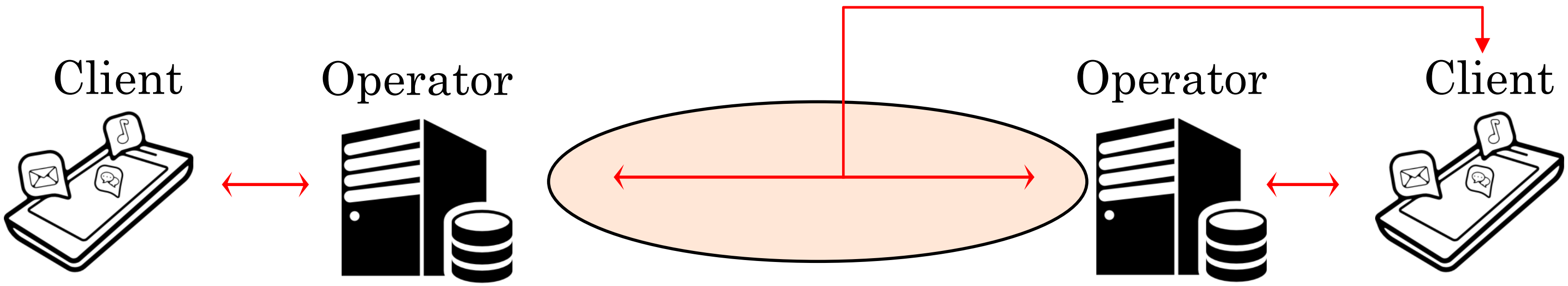
# Towards 5G AKE: Lessons learned from 3G/4G networks


Cristina ONETE (UR1)

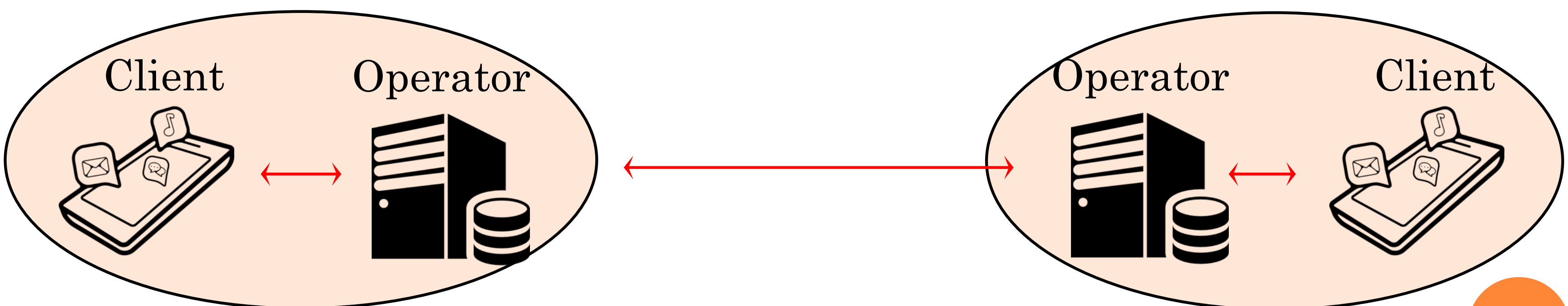
With P.-A. Fouque (UR1),
B. Richard (Orange)
G. Macario-Rat (Orange)
S. Alt (DGA)

# Transit from previous talk

- Previous talk:




- This talk:




**Main focus of my talk: C-Op channel & AKE**

# Contents

- Authenticated Key Exchange
- The AKA protocol
  - Its structure
  - Security problems
  - Privacy problems
  - Fixing AKA
- From 3G/4G to 5G networks

# Authenticated Key Exchange

# Secure Channels

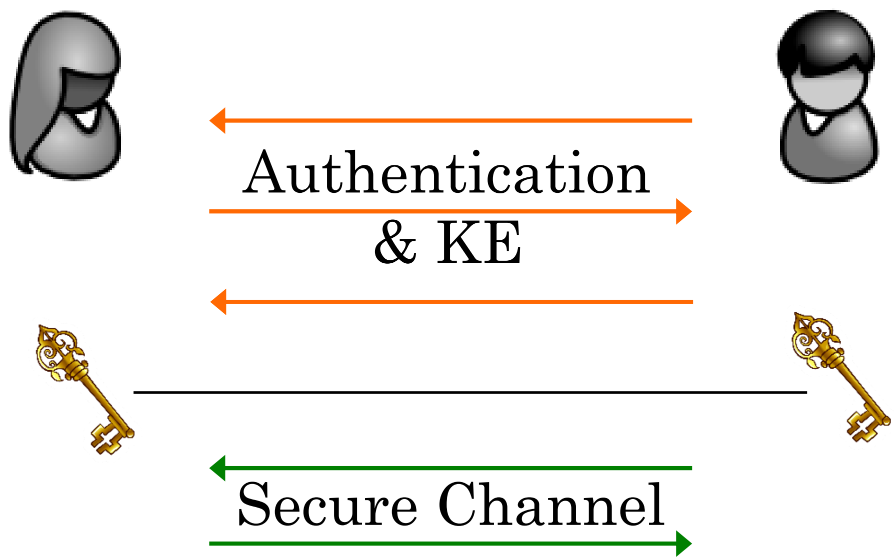
- Goal:

**Secure communication over insecure channels**

- Insecure channels:
  - The Internet (HTTP://)
  - Mobile networks (2G, 3G, 4G…)
  - Bluetooth
  - Radio Frequency Channels

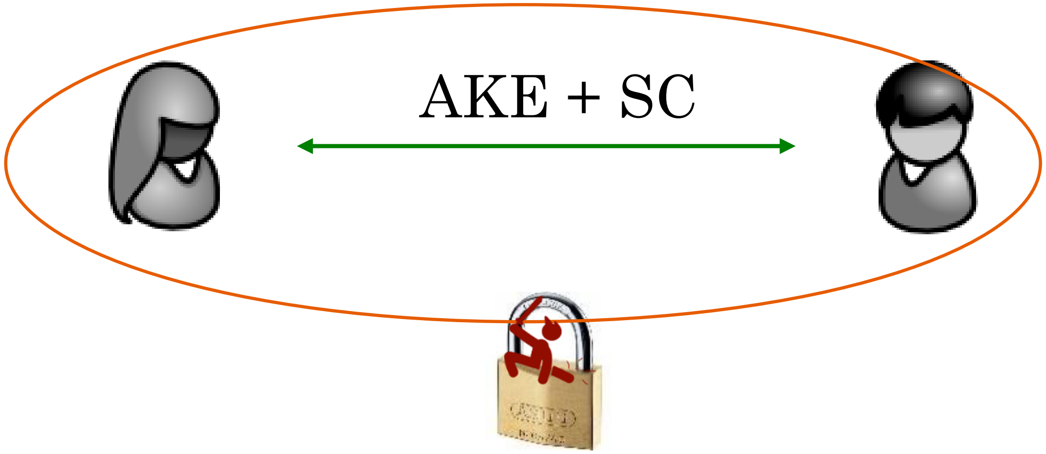
- "Secure" channels:
  - Messages exchanged over this channel could be intercepted, but not read by active $3^{rd}$ parties (Man-in-the-Middle attackers)

# Typical 2-Party AKE



# Security of AKE

- Meet the adversary:
  - A Man-in the Middle, aims to break channel security
  - Can interact in multiple sessions of many parties
  - Can corrupt parties to learn long-term keys
  - Can reveal computed session keys
  - Forward-secrecy: if the adversary corrupts a user, it cannot break the security of past sessions



# Real-world AKE

- In practice, ensures:
  - Secure Internet browsing (TLS/SSL)
  - Mobile services (AKA)
  - Payments
  - Personal identification (ID cards/passports)

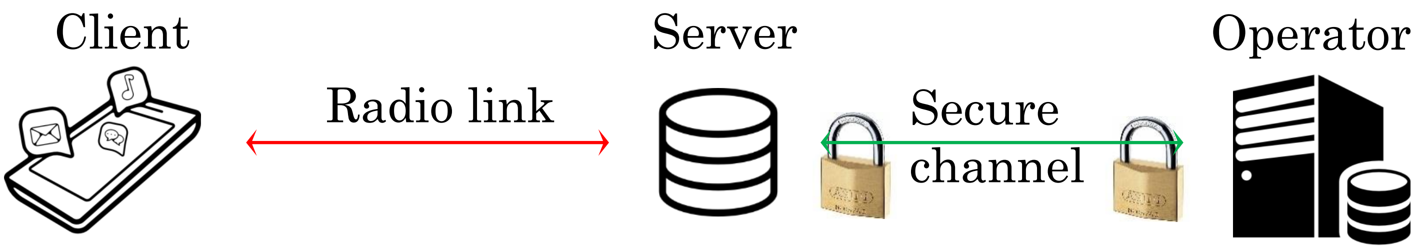
- Security of protocol only proved for 2-party use
  - Yet sometimes, handshakes are proxied, by semi-trusted third parties

**Is the resulting protocol still secure?**

# The case of AKA

# AKA AND 3G/4G NETWORKS

- Communication as a service for mobile users




- Service provided by servers:
  - Local service: usually affiliated with client's operator
  - Roaming: server affiliated with partnering operator
  - Requirement: secure Client-Server channel, with server only semi-trusted

# The AKA protocol

- Standardized in the 1990s by 3GPP
- 3 party design: server proxies between client and operator
- Symmetric-key & stateful

- 3-tiered trust:
  - Operator is trusted with all data: client key $sk_C$, operator key $sk_{op}$, and client-specific state $Sqn_{Op,C}$
  - Client trusted with almost everything: client key $sk_C$, a function of the operator key $sk_{op}$, client state $Sqn_C$
  - Server trusted with nothing: only manages identity management

- Additional concern: client privacy

# THE BASIC 2-PARTY PROTOCOL

$sk_C, sk_{op}, Sqn_C$
UID

$\xrightarrow{\quad UID \quad}$

$sk_C, sk_{op}, Sqn_{Op,C}$
UID

Update $Sqn_{Op,C}$, generate fresh $R$, do:

$MAC_{OP} = F_1(sk_C, sk_{OP}, R, Sqn, AMF)$
$MAC_C = F_2\,(sk_C, sk_{OP}, R)$
$AK = F_5\,(sk_C, sk_{OP}, R)$
$Autn = (Sqn \text{ XOR } AK) \mid AMF \mid MAC_{OP}$
$CK = F_3\,(sk_C, sk_{OP}, R)$
$IK = F_4\,(sk_C, sk_{OP}, R)$

$\xleftarrow{\quad R \mid Autn \quad}$

Compute $AK$, get $Sqn_{Op,C}$ check $MAC_{OP}$
If $Sqn \in \{st_C, \ldots, st_C + \delta\}$ compute:
$CK, IK, Rsp = F_2\,(sk_C, sk_{OP}, R)$

$\xrightarrow{\quad Rsp \quad}$

Else: *Resynch*

Check: $Rsp == MAC_C$

# Resynch Procedure

$\mathrm{sk_C, sk_{op}, Sqn_C}$
IMSI

$\mathrm{sk_C, sk_{op}, Sqn_{Op,C}}$
IMSI

If $MAC_{OP}$ verifies, but $Sqn$ out of range

Compute:

$AK^* = F_5^* \, (sk_C, sk_{OP}, R)$

$MAC_C^* = F_1^*(sk_C, sk_{OP}, st_C, AMF, R)$

$(st_C \text{ XOR } AK^*) \; || \; MAC_C^*$

Compute: $AK^*$, get $st_C$
Check: out of range
Check: $MAC_C^*$
Set $st_{OP}^C := st_C$

Start from there.

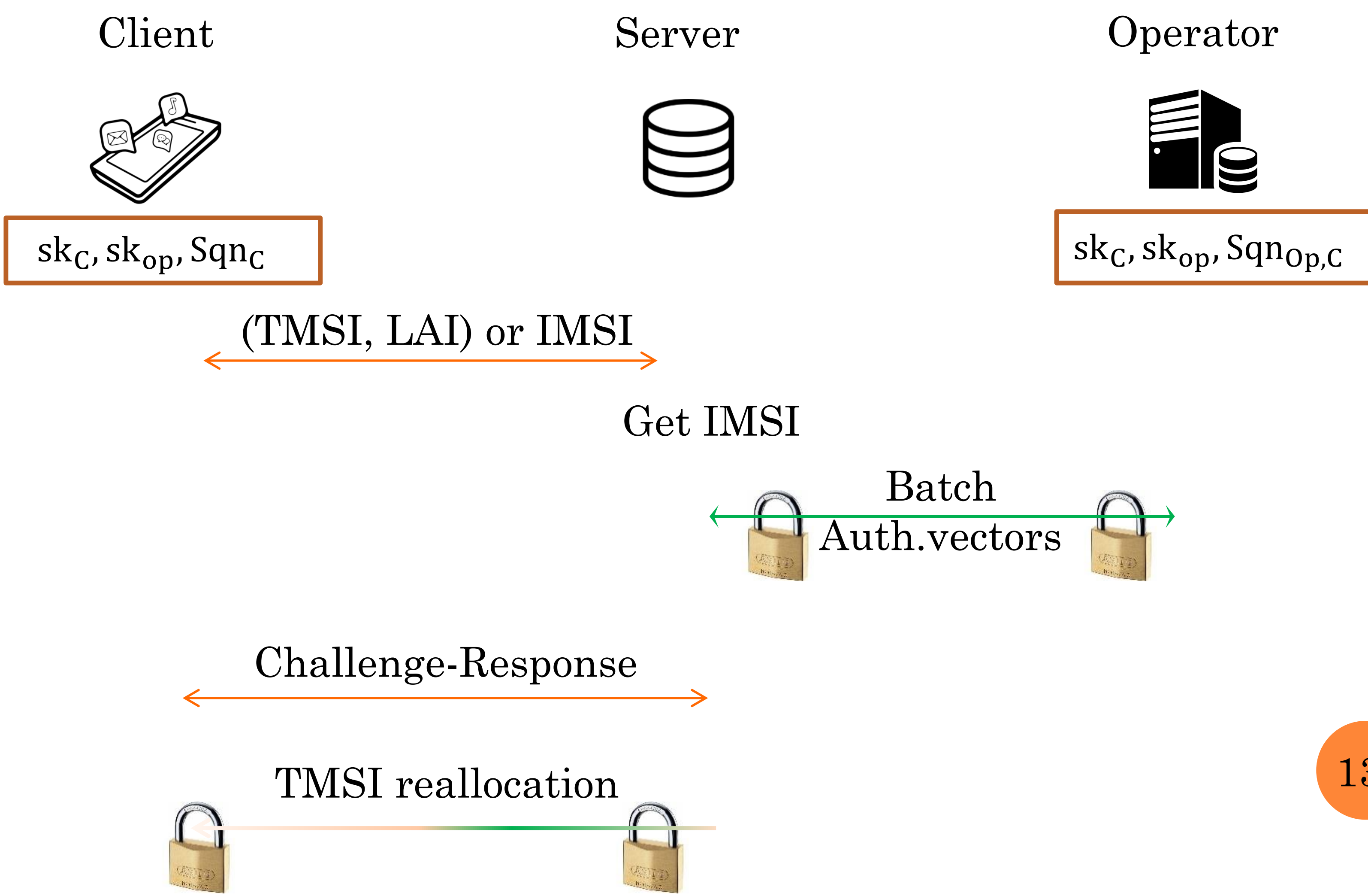
# INTRODUCING THE THIRD PARTY

- The server is not trusted to know $sk_C, sk_{op}, Sqn_C, Sqn_{Op,C}$
- However, it is the server that provides service to the client
  - Only legitimate clients may receive the service
  - Client will only receive service from legitimate servers

**How can authentication work without client secrets?**

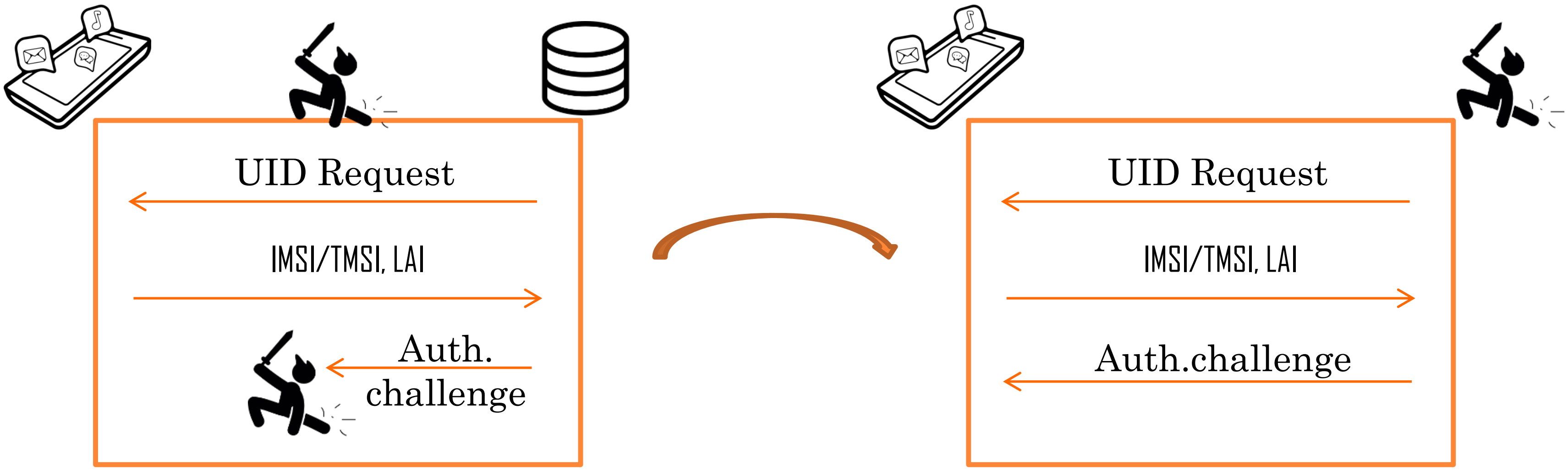
- Server used as proxy, does only identity management
  - Client identifiers: IMSI/TMSI also stored by client and server
  - Area identifier: LAI, unique per server/area
  - IMSI known by all, (TMSI, LAI) tuple handled by server
  - In 4G, TMSI and LAI replaced by GUTI

# AKA PROTOCOL STRUCTURE



# Security Weaknesses of AKA

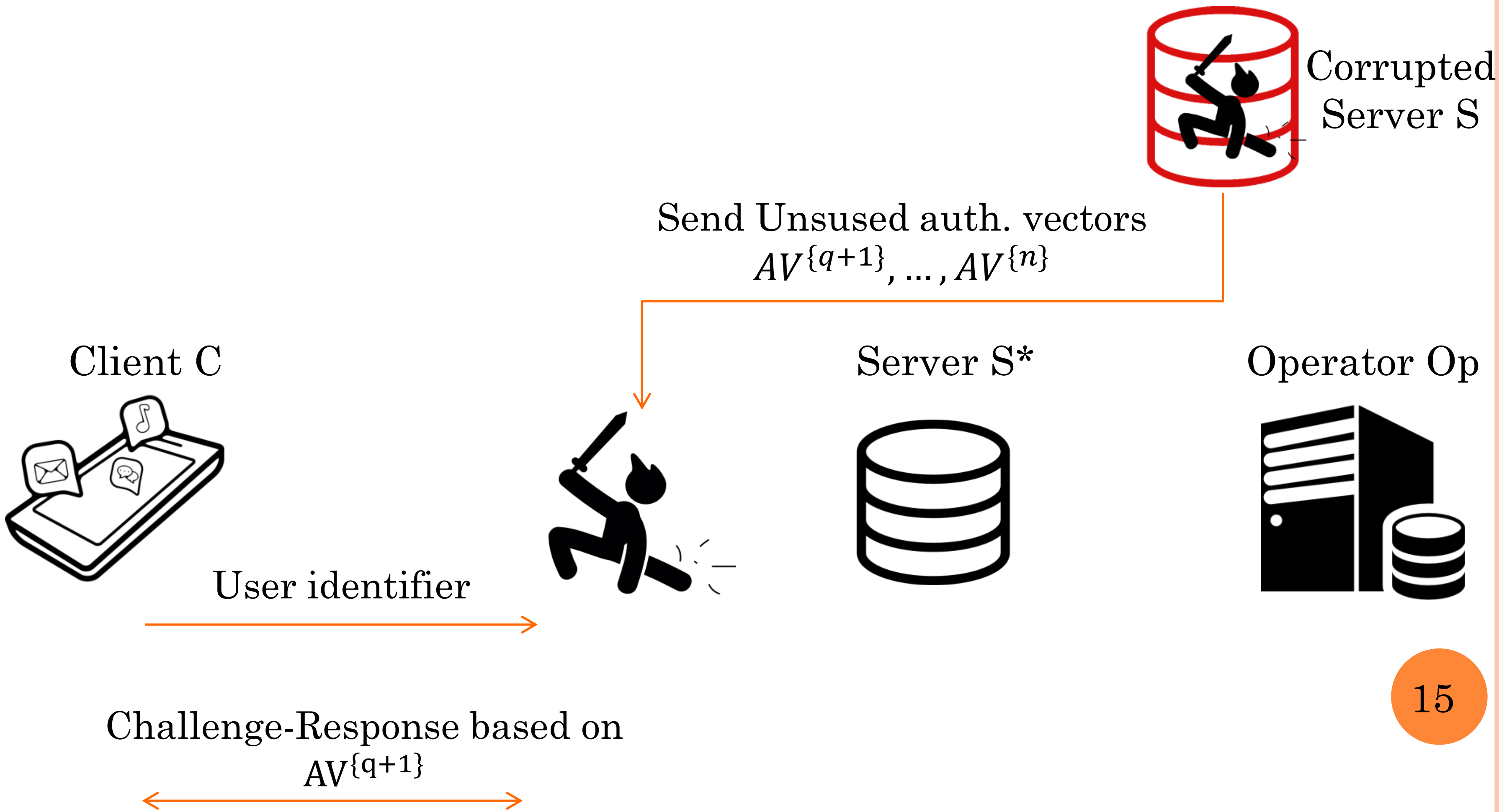
- Server impersonation by offline relays




- Main causes:
  - No authentication of UID
  - No nonce on client side

# Security Weaknesses of AKA (cont'd)

- Impersonation/key-distinguishing attack [ZF03]:



# Security of AKA

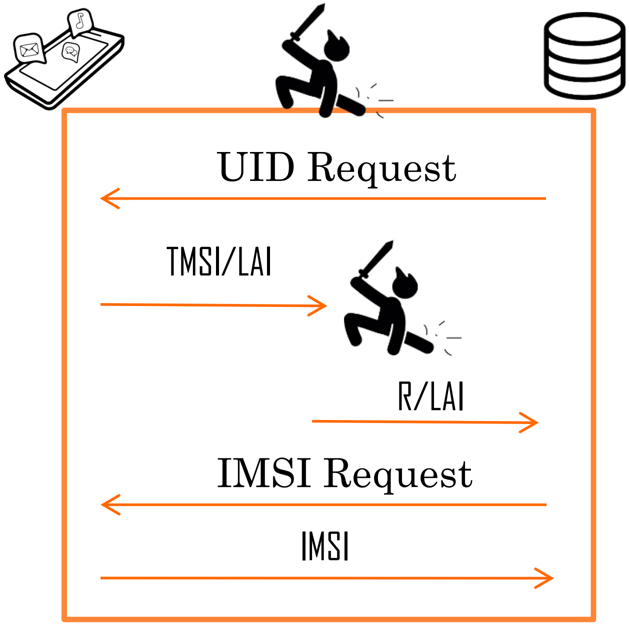
- What AKA guarantees:
  - C-imp. security: even for server corruptions & offline relays
  - S-imp. security: no server corruptions, no offline relays
  - Key-indistinguishability: no server corruptions
  - State confidentiality
  - Soundness

- Where AKA security fails:
  - Server corruption attacks reveal session keys
    - Thus even sessions in "safe" areas are vulnerable
  - IMSI/TMSI insecurity leads to offline relays

# Privacy of AKA

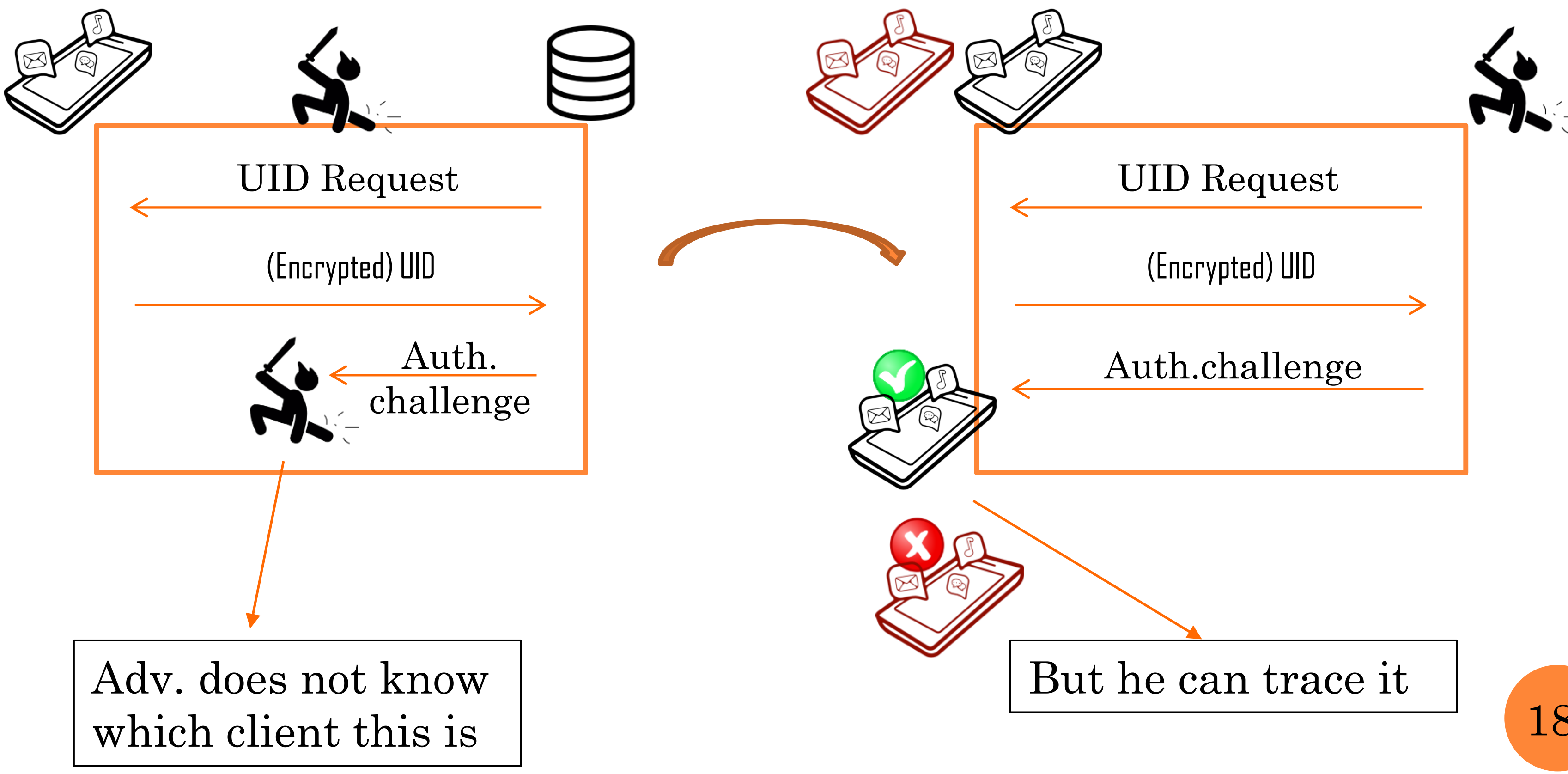
- 3GPP requirements:
  - ID-Hiding: nobody can trace the client's IMSI
  - Location-hiding: nobody can trace the client's LAI
  - Untraceable: nobody can link services to clients

- "IMSI catcher" [S07] attackers break the first two:
  - First get the LAI
  - Then force IMSI revelation
  - [BVR15]: encrypt TMSI with PKE
    But this still allows traceability

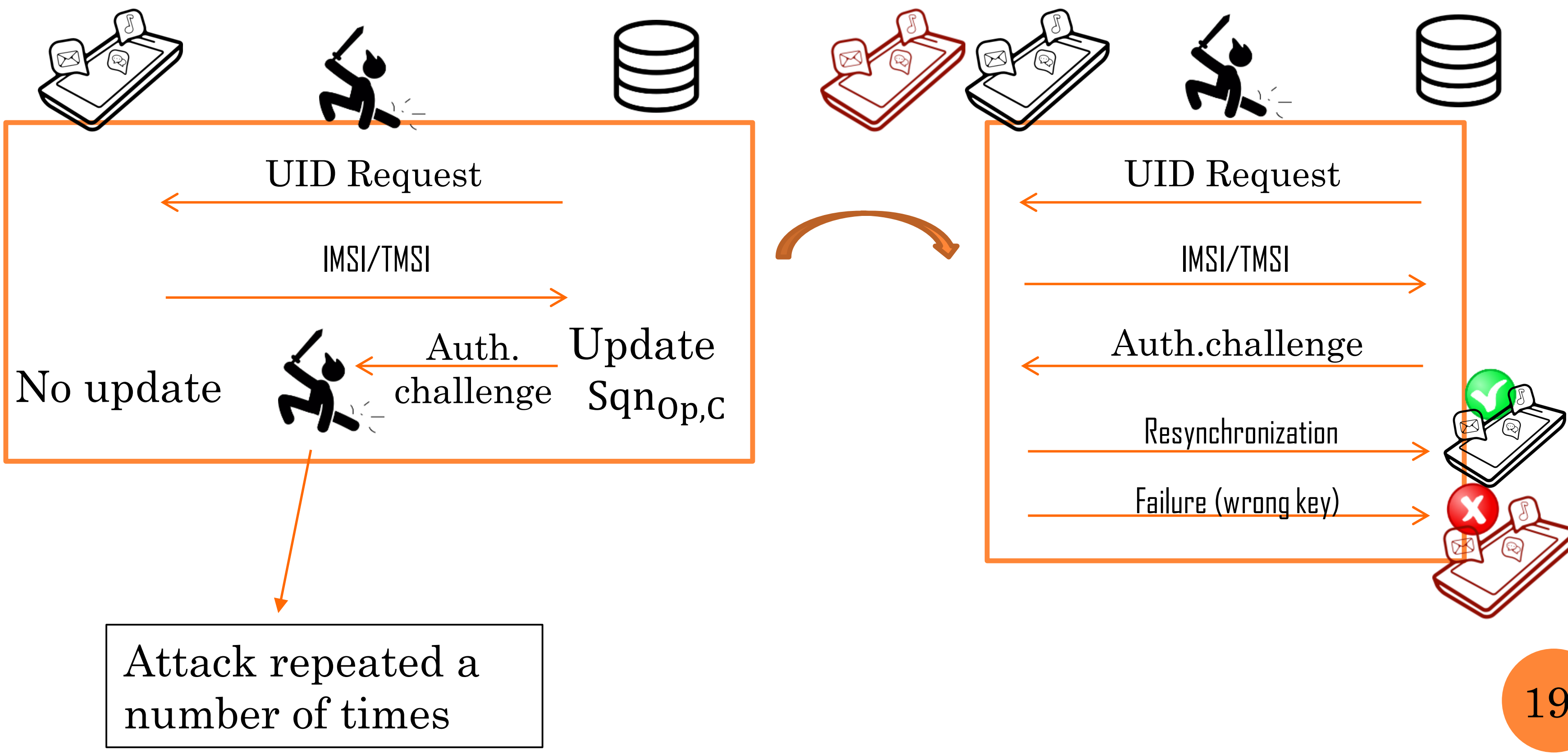


# TRACEABILITY ATTACKS

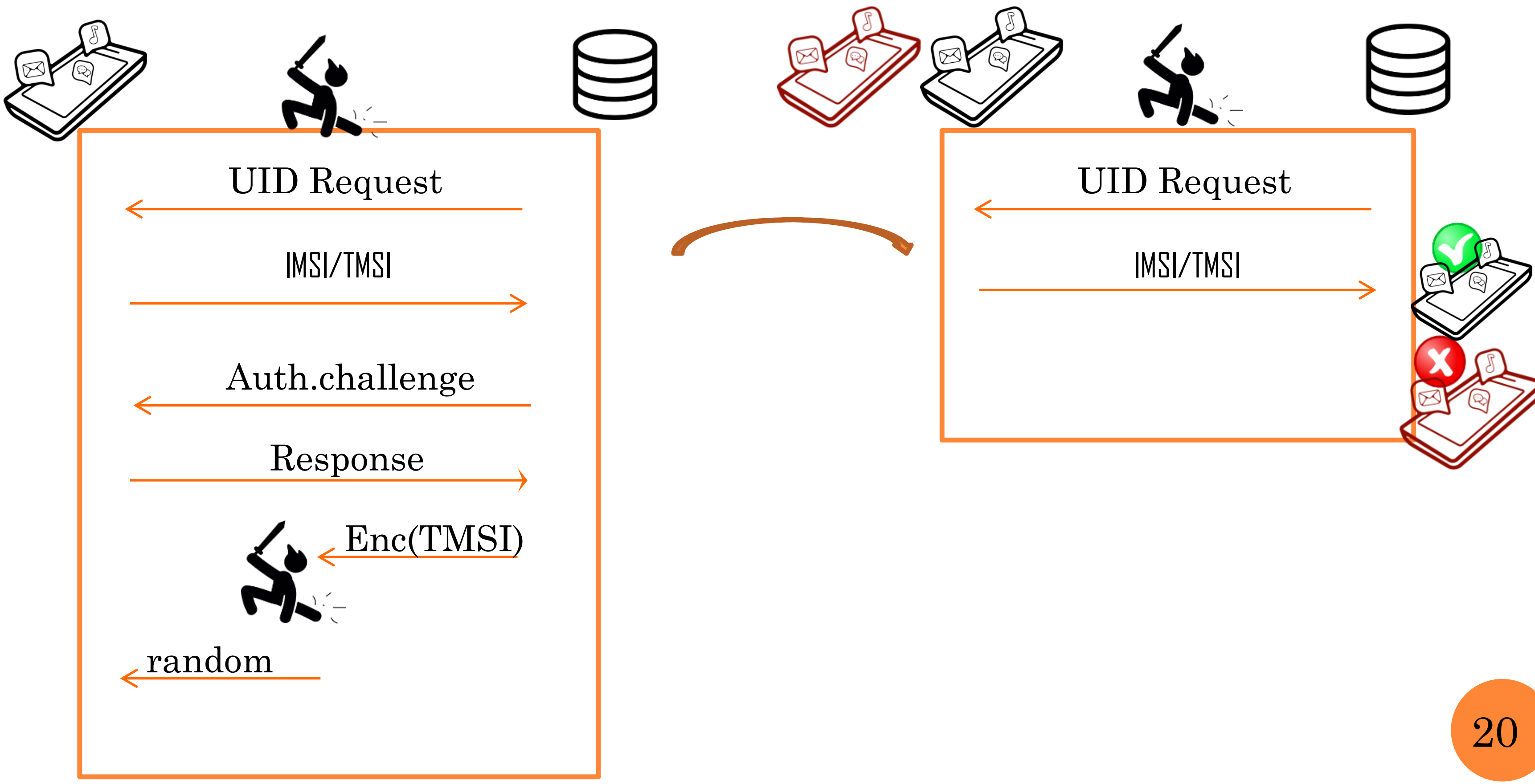
- Distinguishing between two clients allows traceability



# Traceability by resynchronization



# Traceability by TMSI-Reallocation



# Our counter-proposals

- Easy fix: security even with server corruptions
  - Add server identifier to all cryptographic functions
  - Even if a server is corrupted, the adversary cannot use its identity in a different area

- Harder fix: better privacy
  - Encrypt TMSI in smarter way:
    - Use symmetric encryption inside PKE scheme
    - Use Operator as soon as an attack is detected
  - Remove need for resynchronization
  - Add authentication at TMSI reallocation
  - Optimality: impossibility result for better privacy

# Research project: Security & Privacy in 5G

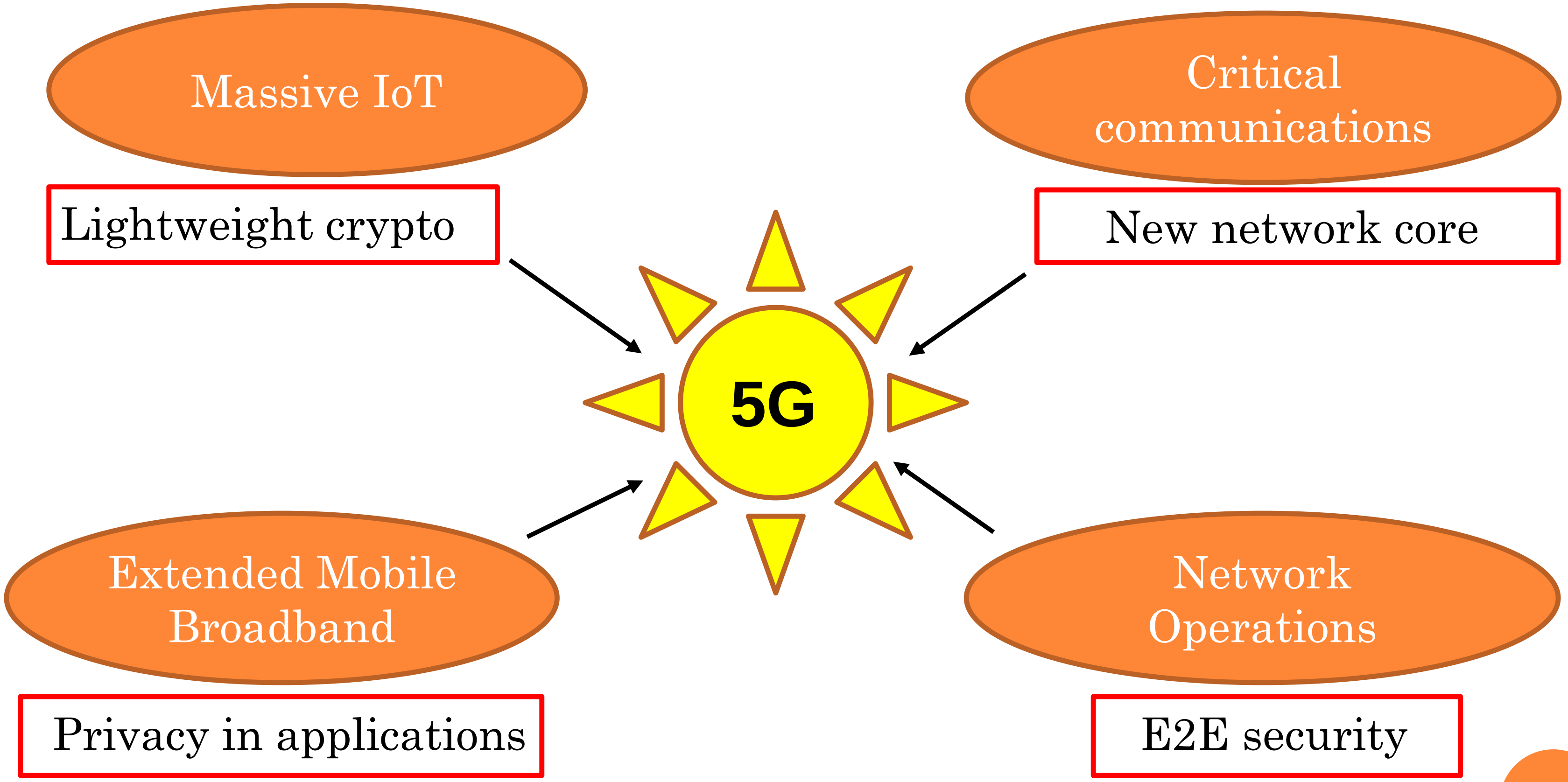
# 3G/4G PROBLEMS ARE FUNDAMENTAL

- 3G/4G AKA provides some limited security
  - And we can fix it to get better privacy

**But should we use it for 5G networks?**

- Some AKA problems:
  - Currently all computation done at the operator's end
  - Legal interceptions: operators reveal long-term keys
  - Strong deviation in practical implementations
  - Application-layer primitives problematic
  - No concept of "E2E": everything goes through Op

# Towards 5G: A fundamental leap



# Challenges for 5G

- A fundamental leap (akin to TLS 1.3 vs 1.2)
- Many new applications
- A transformation for 5G AKE :
  - Protocol that allows for unfederated E2E security
  - Usability/Privacy tradeoff:
    - allow operators to give away less data for LI
  - Different handshakes for different situations:
    - Roaming/domestic , Client-Server/P2P
  - Better application-layer primitives:
    - Including lightweight primitives for data-stream transfer

**Efficiency, compliance to standards, ease of use**