Mobile Security – LTE (cont.)

Network Security - Lecture 8

Ruxandra F. Olimid

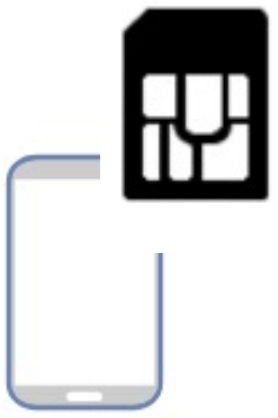
Faculty of Mathematics and Computer Science, University of Bucharest

*slides adapted from the course TTM4137 thought at NTNU

Outline

- UE Identification
- EPS AKA
- Key hierarchy (again)
- Cryptographical aspects
- AS / NAS Protection

UE Identification



- Similar to identification in GSM and UMTS
  - **IMSI**
  - **IMEI , IMEI SV**

- **GUTI (**Global Unique Temporary UE Identity**),** allocated to provide user identity confidentiality
  - Similar to TMSI in GSM

- **C-RNTI** (Cell Radio Network Temporary Identifier) with security role in handover preparation

UE Identification

- MME assigns a GUTI to the UE in `Attach Accept` or `Tracking Area Update Accept` messages
- MME can also assign GUTI in a separate `GUTI Reallocation` procedure

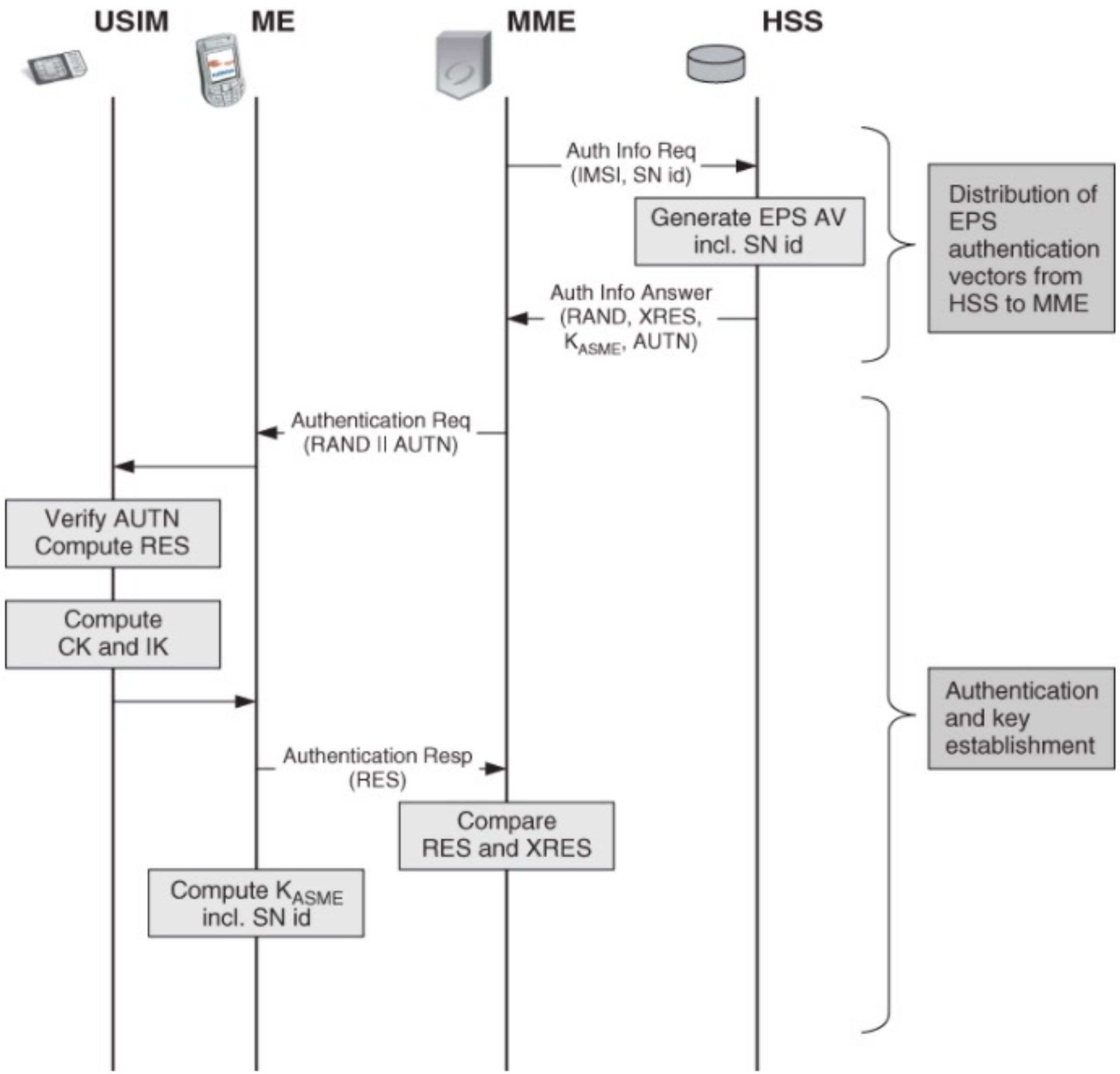
GUTI (Global Unique Temporary UE Identity)

<table>
<tr><td colspan="4">GUMMEI (Globally Unique MME Identifier)</td><td rowspan="3">M-TMSI
(MME Temporary Subscriber Identifier)</td></tr>
<tr><td rowspan="2">MCC
(Mobile Country Code)</td><td rowspan="2">MNC
(Mobile Network Code)</td><td colspan="2">MMEI
(MME Identifier)</td></tr>
<tr><td>MMEGI
(MME Group ID)</td><td>MMEC
(MME Code)</td></tr>
</table>

Identifies the MME that allocated the GUTI

Identifies the UE within the MME

EPS AKA



SN id: Serving Network Identity
AV: Authentication Vector
AUTN: Authentication Token
RES: Response
XRES: Expencted Response
CK: Ciphering Key
IK: Integrity Key
ASME: Access Security
Management Entity

[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

EPS AKA – Network side

- The recommendation is to send a single AV at a time (not more)…
  … because the need to request fresh AV is reduced due to the existence of the K_{ASME}, which is not exposed as the CK and IK were exposed in UMTS

- Precomputed AV are not longer used when the UE moves to another network...
  ... because the SN id is input to the KDF

- Each AV is used only once

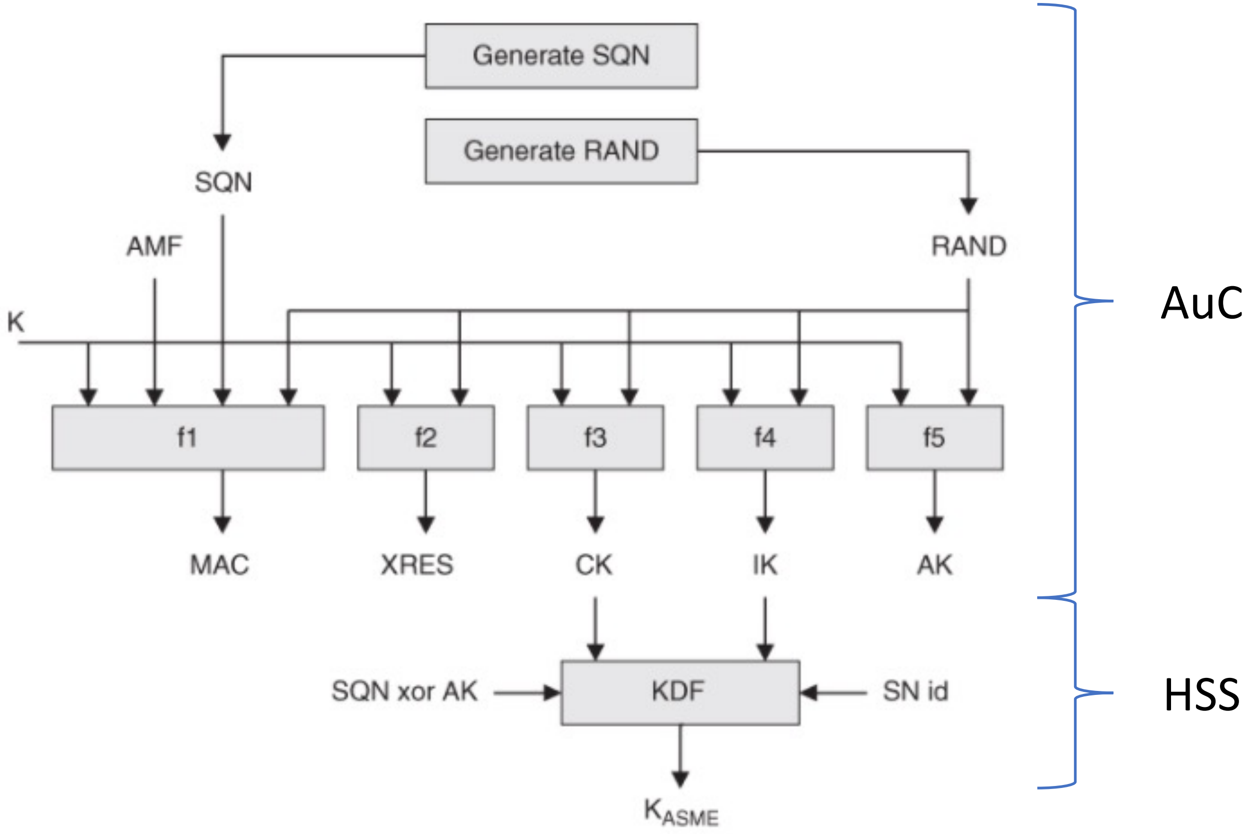
- CK and IK do not leave the HSS

- Operator specific: if AK=0, then AK XOR SQN = SQN (if the operator decides no need for concealment of SQN is required)

EPS AKA – Network side

UMTS AV:
(RAND, XRES, CK, IK, AUTN)

EPS AV:
(RAND, XRES, K_{ASME}, AUTN)



AUTN := SQN xor AK II AMF II MAC

UMTS AV := RAND II XRES II CK II IK II AUTN

EPS AV := RAND II XRES II K_{ASME}II AUTN

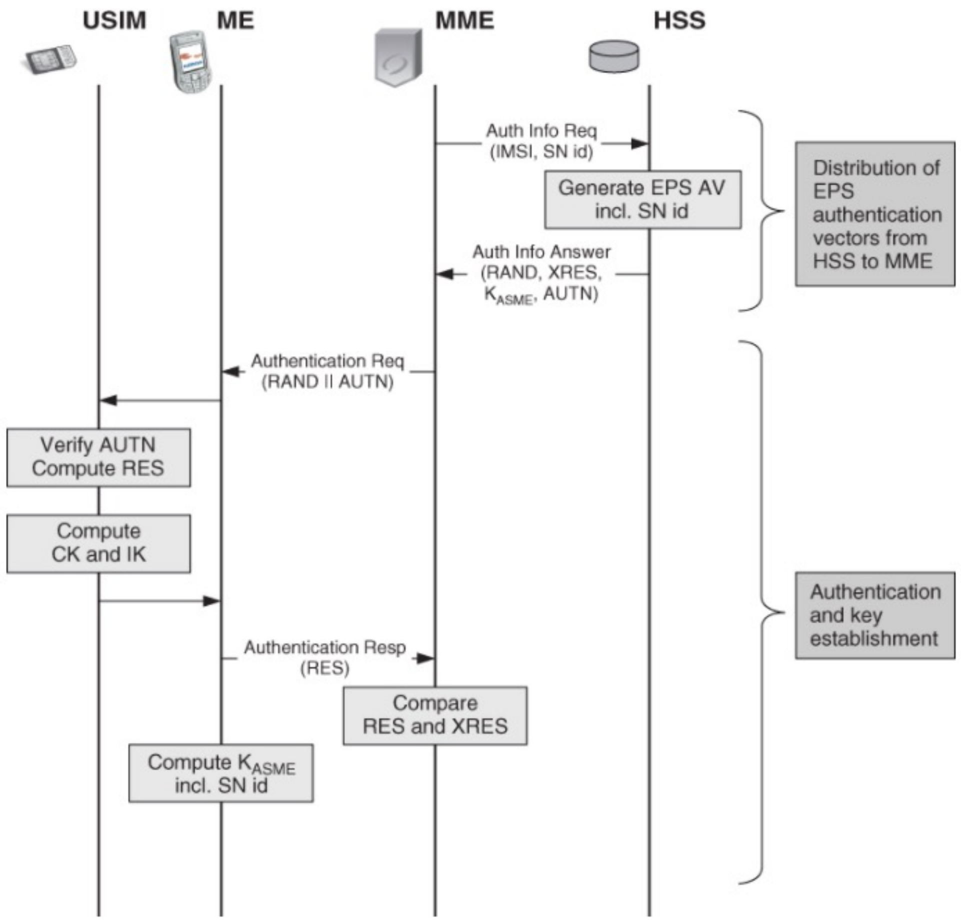
AMF: Authentication Management Field
AK: Anonymity Key

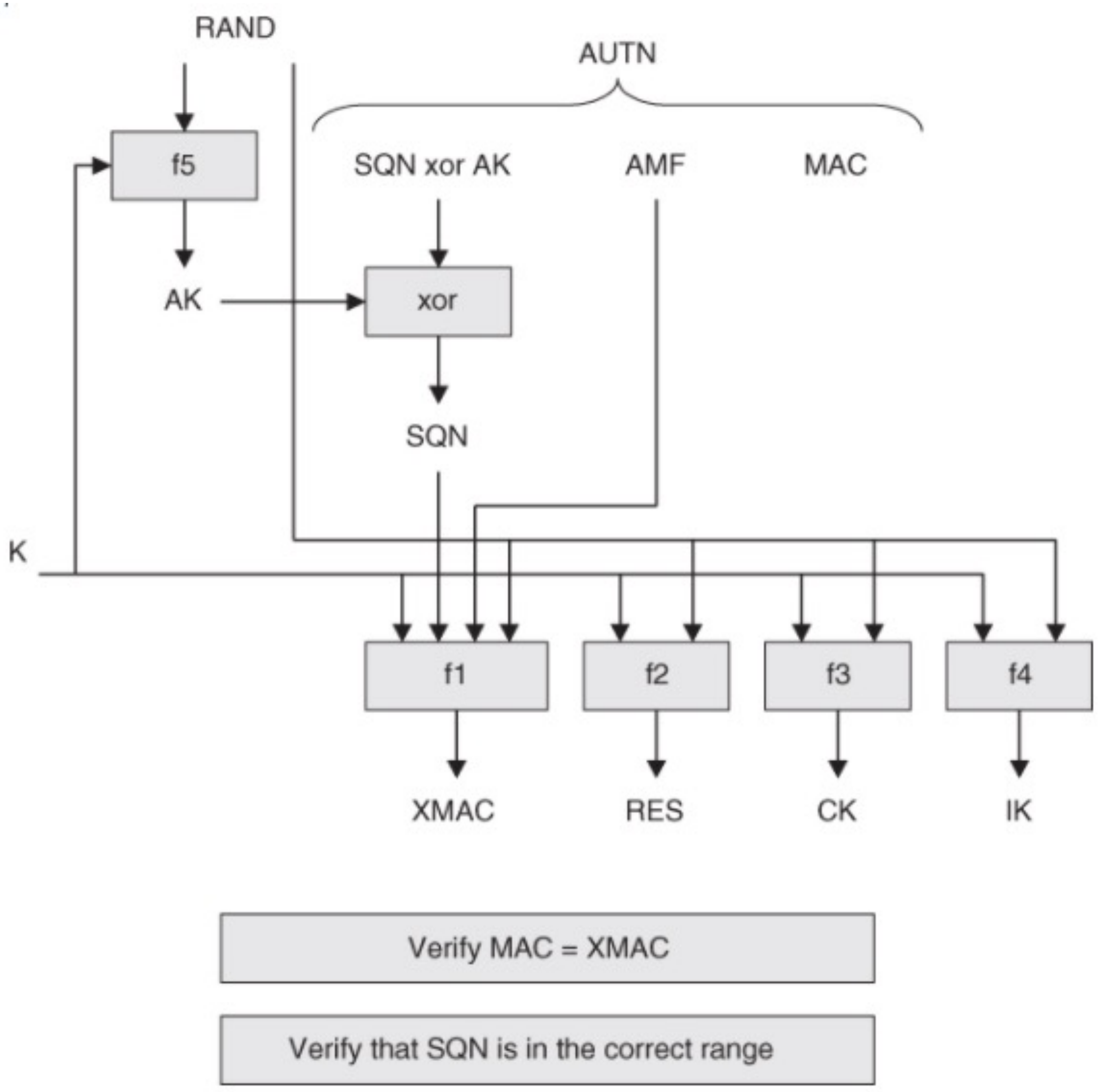
[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

EPS AKA – Network side

- Both UMTS and EPS authentication vectors are generated
- The AuC generates the AVs in exactly the same way as for UMTS
- The HSS derives the K_{ASME} from CK and IK
- The AuC generates fresh SQN and unpredictable random RAND
- AMF (Authentication Management Field):
  - Indicates the algorithm used to generate a particular auth vector when several exist
  - Sets threshold values for key lifetimes
  - First bit is set to 1 to mark that the AV is for EPS use (this should be checked in the MME)

EPS AKA – User side





[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

EPS AKA – User side

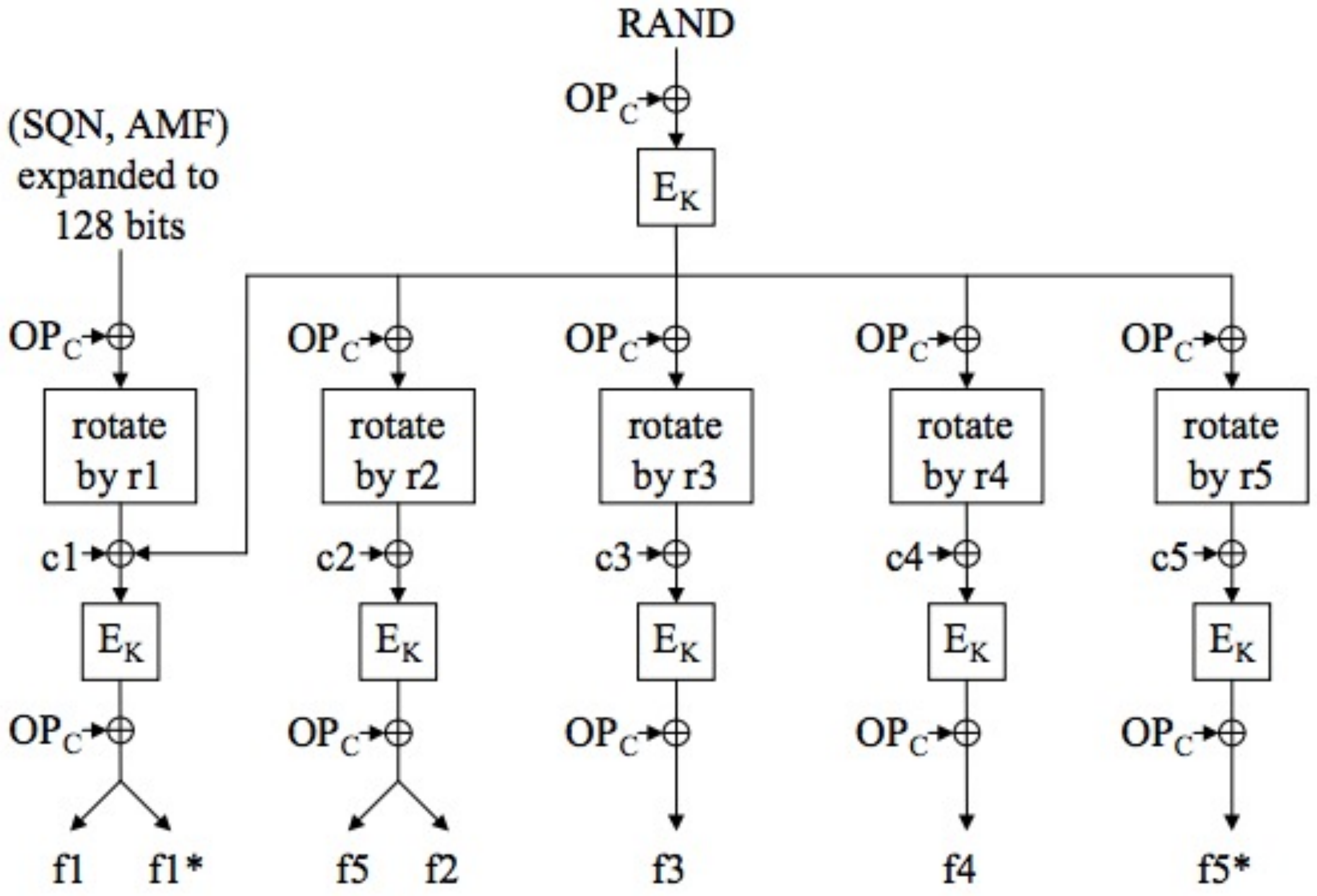
- SQN verification has not been standardized (generation and verification takes place in the home network, so it can be operator specific)
- Requirements for SQN:
  - No SQN should be used twice: USIM should not accept 2 AUTN with the same SQN after AUTN was verified
  - Allow, in a given threshold, out of order SQN numbers
    (might not accept a SQN if the jump from the last one is too big)
  - Reject too old time-based SQN
- Verification is performed in the USIM

An example: MILENAGE

OPc: Operator variable derived (128 bits)
r1…r5: fixed rotations constants
c1…c5: fixed addition constants
E_K: encryption with they K

Note: f1*, f5* used in case sync.failure at auth. (see Sect.7.2.3, Auth.failures) in the book

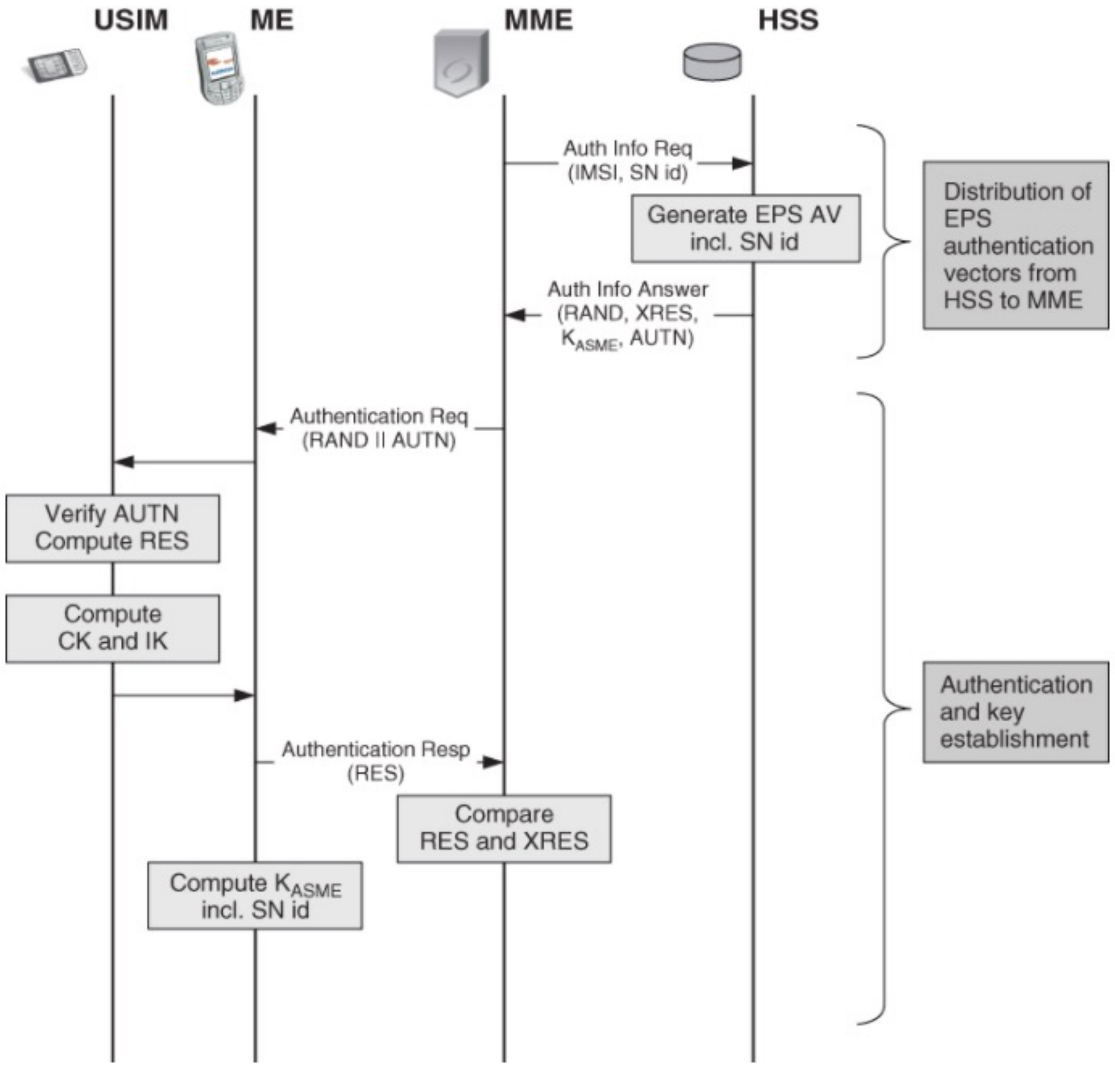
f0	the random challenge generating function;
f1	the network authentication function;
f1*	the re-synchronisation message authentication function;
f2	the user authentication function;
f3	the cipher key derivation function;
f4	the integrity key derivation function;
f5	the anonymity key derivation function.
f5*	the anonymity key derivation function for the re-synchronisation message.



Definition of f1, f1*, f2, f3, f4, f5 and f5*

[Source: ETSI TS 135 205 V13.0.0 (2016-01)]

EPS AKA – User side



- If USIM supports GSM, then it converts (CK, IK) to a GSM key K_c and sends it the the ME

Handover and Roaming

- When the UE changes MME, it identifies itself by GUTI in the `Attach Request` and `Tracking Area Update Request`

- The MME is unaware of the GUTI, so it has 2 possibilities:
  - *Request the IMSI* – breaks confidentiality!
  - *Ask the old MME to translate the GUTI to IMSI*

- Data exchanged between the old and the new MME in 2 scenarios:

  - *Old and new MME are in the **same** network (*Handover*)*
    - Transfer the EPS security context*
    - The old MME transfers the remaining AVs (if any)
  - *Old and new MME are in **networks of different operators** (*Roaming*)*
    - The current security context* is allowed, depending on the security of the networks (EPS to EPS only)
    - The old MME does not transfer the remaining AVs (if any), because they are not good in the new network

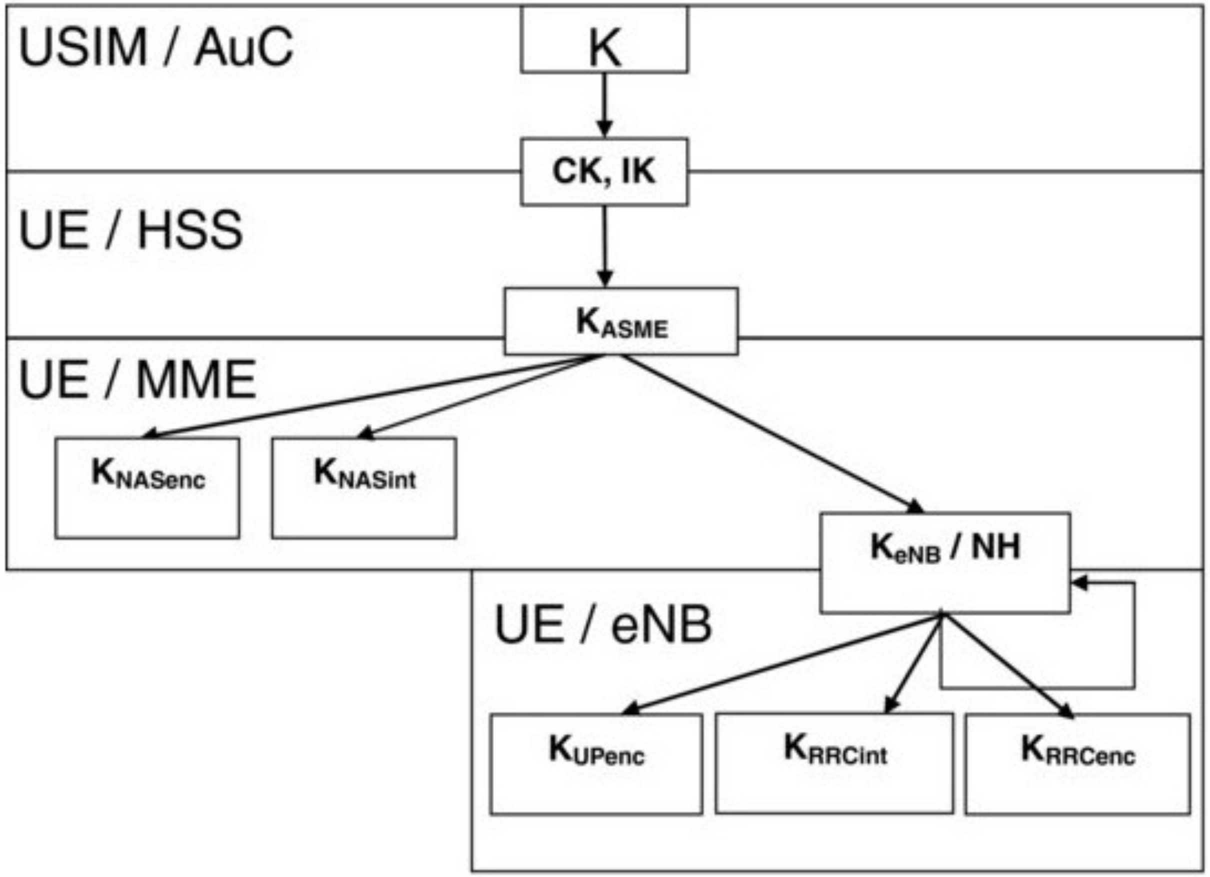
Security Context

- A security context is a set of parameters agreed by 2 parties when they engage in a secured communication

- Contains: algorithm identifiers, cryptographic keys, etc.

Key hierarchy (remember!)

Key	Length	Info
K	128 bits	Key shared between the subscriber and the network operator, stored in the USIM and AuC; permanent key of the subscriber
CK, IK	128 bits	Ciphering key CK and integrity key IK are for UMTS interconnection
K_{ASME}	256 bits	A local master key of the subscriber from which all other keys will be derived; Shared between the UE and the MME
K_{NASenc}, K_{NASint}	128 / 256 bits	Ciphering key K_{NASenc} and integrity key K_{NASint} for NAS protection
K_{eNB} /NH	256 bits	Intermediate key stored in the eNodeB NH (Next Hop) is used in handover
K_{RRCenc}, K_{RRCint}	128 / 256 bits	Ciphering key K_{RRCenc} and integrity key K_{RRCint} for AS protection
K_{UPenc}	128 / 256 bits	Ciphering key K_{UPenc} for user data



[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

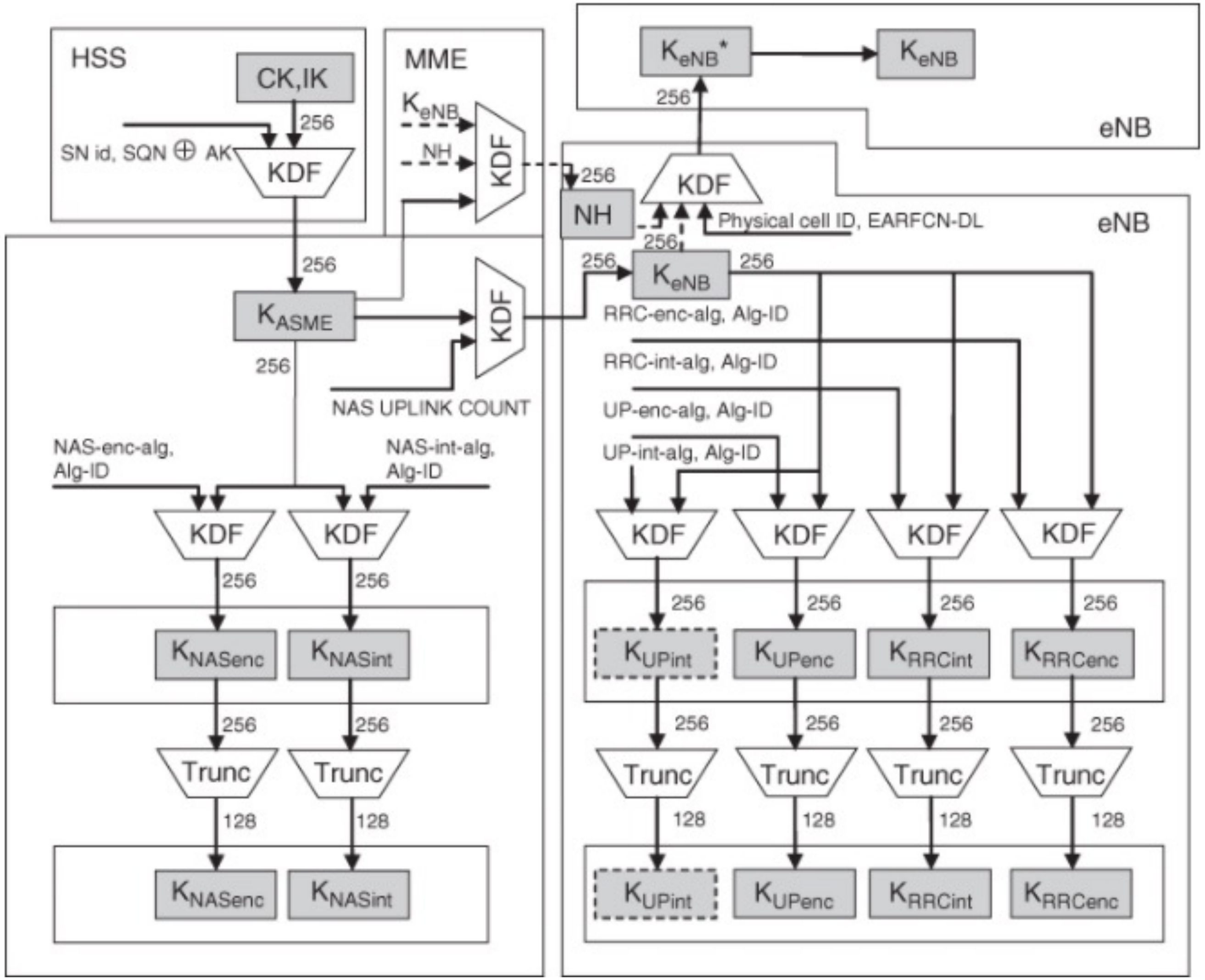
Key hierarchy

- K_{ASME} is derived in the ME (not the USIM!) and the HSS => its derivation it must be standardized; others not necessarily
- KDF used to derive keys in the hierarchy must be one-way; **why?**
- KDF based on **HMAC-SHA-256**
- Encryption and integrity keys (K_{NASenc}, K_{NASint}, K_{RRCint}, K_{RRCenc}, K_{UPenc}) are on 256 bits and truncated to 128 last significant bits (EPS accepts both 256 and 128 bits keys)
- Keys are derived in hierarchical manner, with additional parameters as input (e.g.: SN id, SQN xor AK, etc.) – the params are all **assumed to be known by a potential attacker** because they are sent in clear or easy computable from unencrypted communication

Key hierarchy

- A principle that brings **advantages**:
  - Cryptographic key separation:
    - Each key is used to one context only (e.g.: encryption of signalling traffic)
    - Prevents **expanding of leakage**: leakage of keys in one context do not help finding the key in another context
    - **Related key attacks**: the attacker can ask the exchange of the key in a way that he predetermines the relation between the old and new keys
  - Key freshness:
    - Keys can be renewed without affecting other keys (e.g.: renew of K_{eNB} does not require renewal of the K_{ASME} , X2 hadover)
    - Renewal of keys takes place more often
- … and **disadvantages**: added complexity

Key hierarchy



Question: Can K_{NASenc}, K_{NASint} be refreshed without refreshing the K_{ASME} ? How?

Just by changing the other param, NAS-enc/int-alg Alg_ID

Cryptography

- Algorithm agility / flexibility: the cryptographic **algorithms should be replaced** without much difficulty
  - Allows removal of out-dated algorithms
  - The number of algorithms should be keep small (for synchronization and management reasons), but more than 1…
  - … because if one algorithms fails (is broken), others will be used
- Algorithms diversity**: the design** of the algorithms **should differ** from each other as much as possible
  - *Why? Where did you encounter this principle before (in crypto)?*
- Emergency scenarios

Emergency

- Null algorithm: provides no cryptographic protection
  - Must exist for emergency cases
  - **Problematic** from security perspective because it can be triggered in cases where protection should be enabled
- Turn-off principle: the **cryptographic protection should be by default on**, and only by request (on special scenarios) should be turned off
- EEA0 (EPS Encryption Algorithm): the identity function (i.e. ciphertext equals the cleartext)
- EIA0 (EPS Integrity Algorithm) **: a 32-bit string of 0's** is appended to the message
  - Reason: **keep the protected and non-protected scenarios as similar as possible** (e.g.: same length)

Confidentiality

- Same structure for NAS and AS protection
- Out-of-the shelf algorithms (easier than to invite submission and go through a selection process) …
- … keeping in mind **reusability** from 3G (compatibility reasons)

- 128-EEA1: **SNOW 3G** adapted to the EPS security architecture
  - 128 bits keys
- 128-EEA2: **AES** over **KASUMI**
  - 128 bits keys
  - Counter mode

Integrity

- Same principles as for confidentiality
- Usage of the same main cryptographic blocks (re-usability)

- 128-EIA1: **UIA 2 (SNOW 3G)** adapted to the EPS security architecture
  - 128 bits keys
- 128-EIA2: **Cipher- based MAC (AES)**
  - 128 bits keys

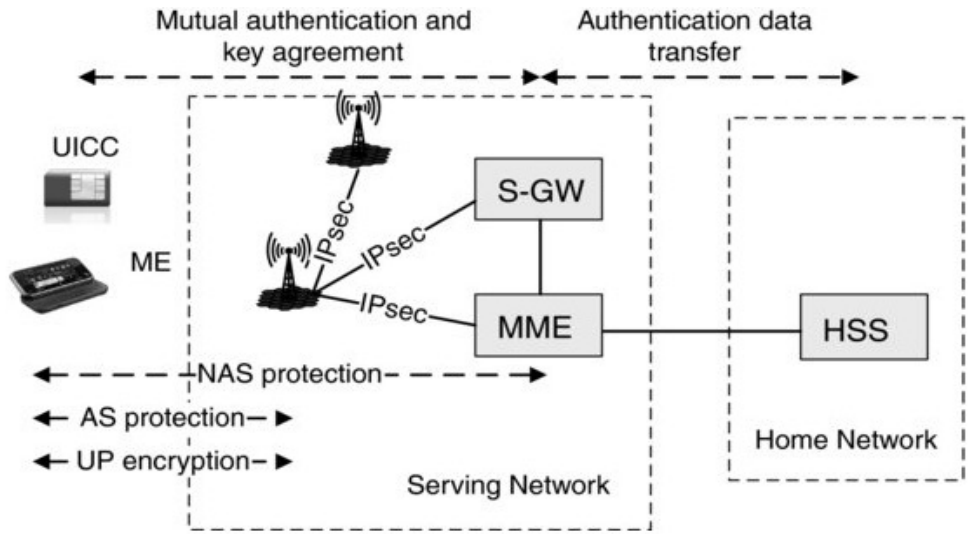
- The key length in the naming implies that other key lengths (e.g.:192, 256) can be used in case of improved security

Key derivation

- One-way: an adversary cannot use one key to derive a key located upper in the hierarchy
- Independence: 2 keys derived from the same key should be independent
- **SHA-256** used in the **HMAC** mode

Algorithm negotiation

- Algorithms are **negotiated separately** for AS (between UE and eNodeB) and NAS (between UE and MME)
- Negotiation is based on the UE capabilities and a list of allowed cryptographic algorithms in the eNodeB, respectively MME in priority order
- eNodeB and MME are responsible for selecting the AS level, respectively the NAS level algorithms, after UE sends its capabilities in the attachment procedure
- Selection is indicated in AS Security Mode Command, respectively NAS Security Mode Commands

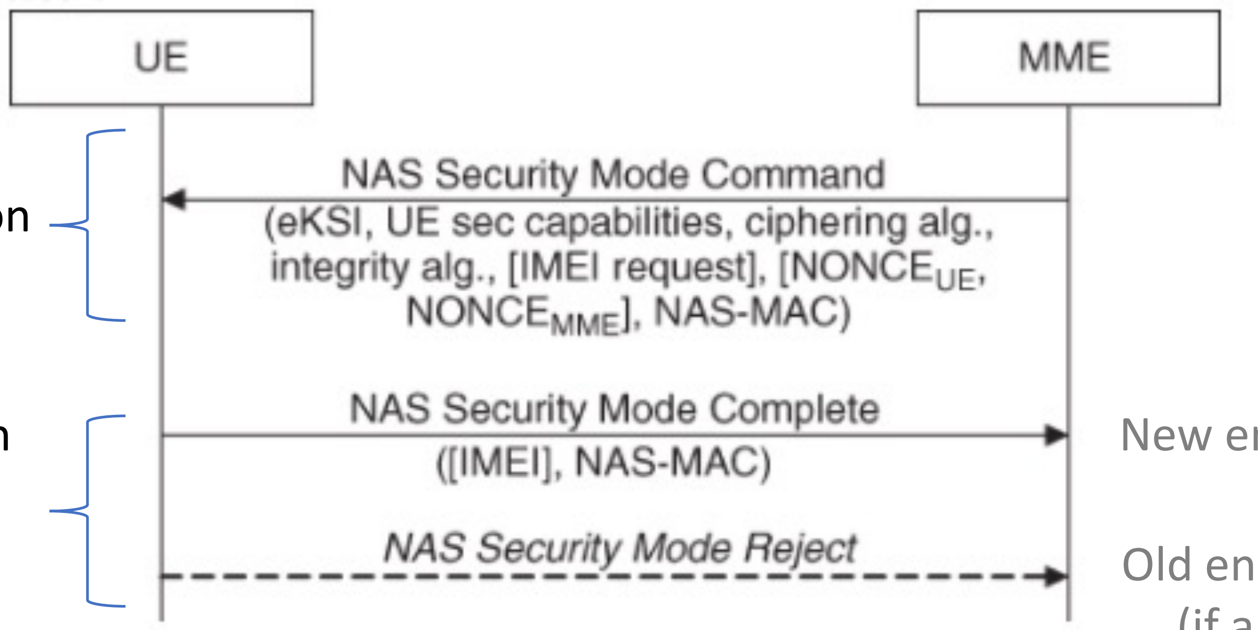


[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

NAS signalling protection

Integrity protection

Integrity protection
&
Encryption



New enc.key

Old enc.key
(if any)

[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

eKSI: key set identifier that identifies the key K_{ASME}
NONCE$_{UE}$, NONCE$_{MME}$: used for mobility

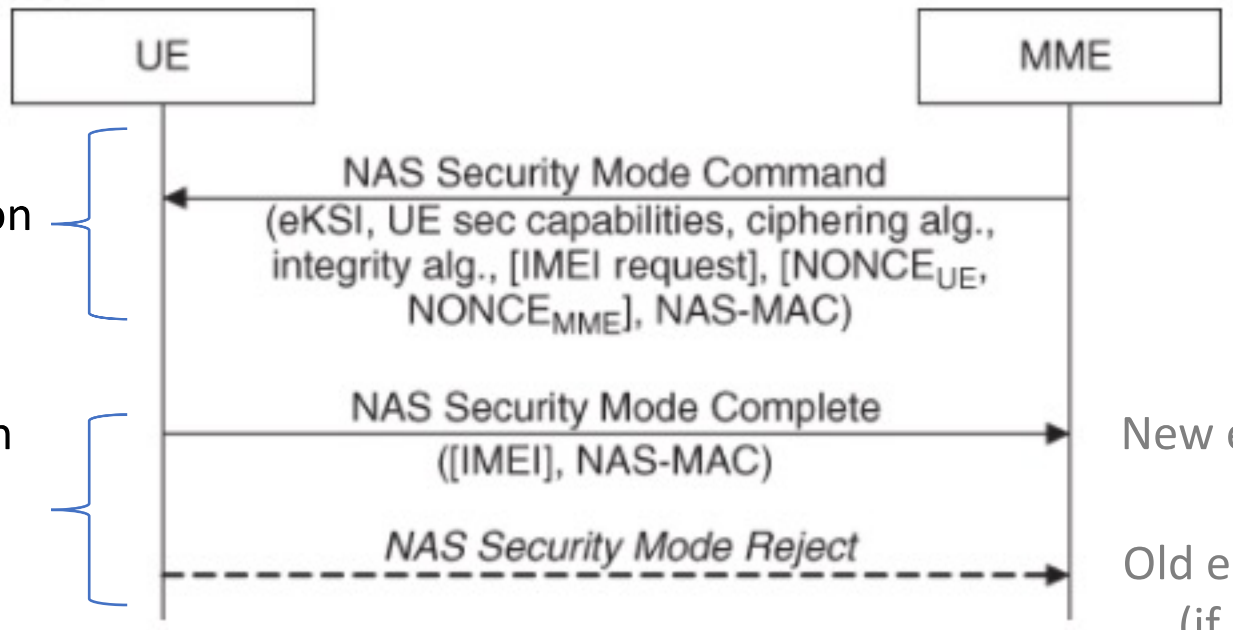
Question: Why is the NAS Security Mode Command not encrypted?

The UE does not know what algorithm and key to use for decryption

NAS signalling protection

Integrity protection

Integrity protection
&
Encryption



New enc.key

Old enc.key
(if any)

[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

eKSI: key set identifier that identifies the key K_{ASME}
NONCE$_{UE}$, NONCE$_{MME}$: used for mobility

Question: Why is the NAS Security Mode Complete encrypted?

To not expose IMEI

NAS signalling protection

- **Integrity** and **replay protection** are part of the NAS protocol itself

- Integrity algorithm's **input** params:
  - K_{NASint} , 128 bits key
  - COUNT, 32 bits
  - DIRECTION, 1 bit indicating upstream or downstream signalling
  - BEARER, constant value – used for similarity with AS

COUNT = 0x00 || NAS OVERFLOW || NAS SQN

  - NAS OVERFLOW, 16 bits – incremented every time NAS SQN overflows

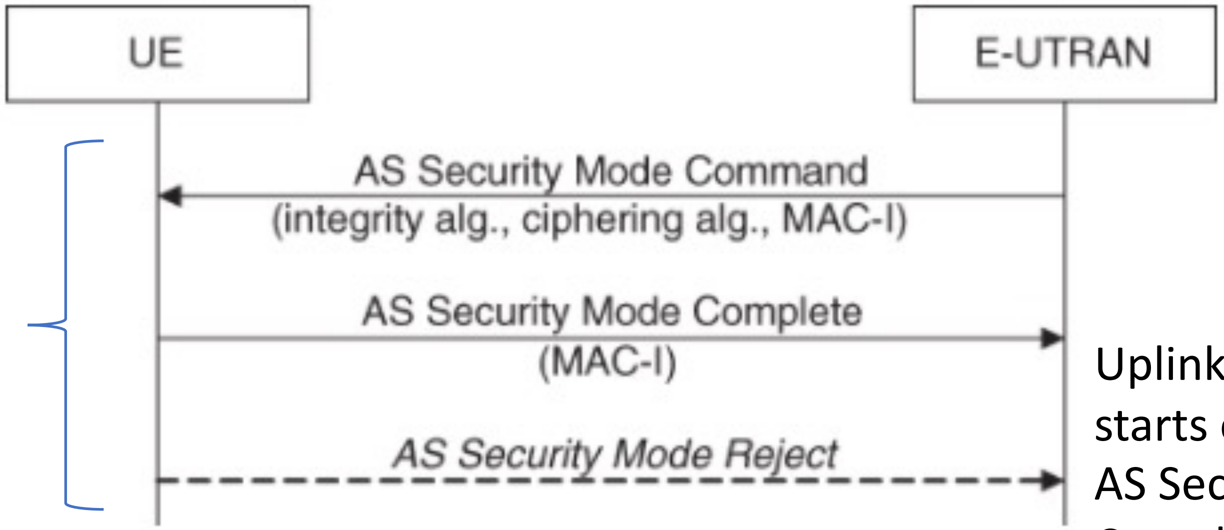
- Integrity algorithm's **output**:
  - NAS-MAC, 32 bits
- For efficiency reasons, `NAS Service Request` message uses 16 bits NAS-MAC (e.g.: when UE responds to paging from the MME)

NAS signalling protection

- **General rule:** messages that are not integrity protected are discarded in the UE and MME once the NAS protection has been activated
- **Exceptions:** emergency calls, etc.
- Ciphering**:** same inputs, except K_{NASenc} instead of K_{NASint} and an additional parameter LENGTH that specifies the length of the keystream to be generated

AS signalling protection

Integrity protection



Uplink enc. starts only after AS Sec.Mode Complete is sent

[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

Question: Why is the AS Security Mode Complete not encrypted?

No need, it contains no private information

AS signalling and User data protection

- Radio Resource Control (RRC): the AS level signalling protocol
- The security is implemented in the PDCP (Packet Data Convergence Protocol) layer, which carries both RRC and user data
- Integrity algorithm's **input** params:
  - K_{RRCint} , 128 bits key
  - COUNT, 32 bits, for each radio bearer (PDCP seg.no).
  - DIRECTION, 1 bit, indicating upstream or downstream
  - BEARER, 5 bits indicating the radio bearer identity, mapped from RRC bearer identity:

Same inputs as for NAS, but a different key and BEARER not constant

Signalling Radio Bearers (SRB):	**SRB0** RRC control messages not protected	**SRB1** RRC control messages protected after sec. activation	**SRB2** NAS messages Always protected
Data Radio Bearers (DRB)	multiple ciphered, but not integrity-protected		

- Integrity algorithm's **output**:
  - MAC-I, 32 bits

AS signalling and User data protection

- Ciphering: same inputs, except K_{RRCenc} instead of K_{RRCint} and an additional parameter LENGTH that specifies the length of the keystream to be generated

NAS vs AS Security Mode Commands (SMC)

AS (Access Stratum)

- Signalling protection **and user data protection**
- Security is implemented in the **PDCP protocol**
- It is **not possible** to change algorithms using `AS Security Mode Command`
- Encryption starts **after** the `AS Security Mode Complete`
- **Several bearers** (there are several AS level connections between UE and eNodeB)

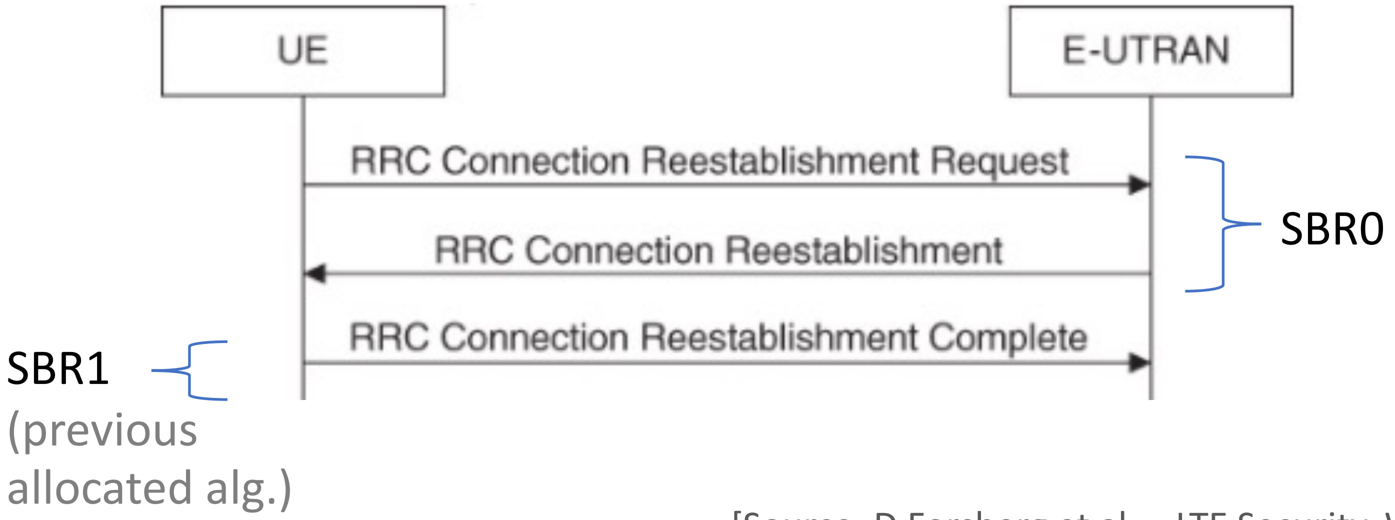
NAS (Non-Access Stratum)

- Signalling protection
- Security is implemented in the **NAS protocol itself**
- It is **possible** to change algorithms using `NAS Security Mode Command`
- Encryption starts **with** the `NAS Security Mode Complete`
- **One bearer** of **constant value** (there is only one NAS level connection between UE and MME)

RRC Connection re-establishment

- Initiated by the UE when there are problems (physical connection, integrity checksum errors, handover errors, etc.)

- Purpose:
    - Resume SRB1 operation
    - Reactivate security without change of security algorithms



[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

RRC Connection re-establishment

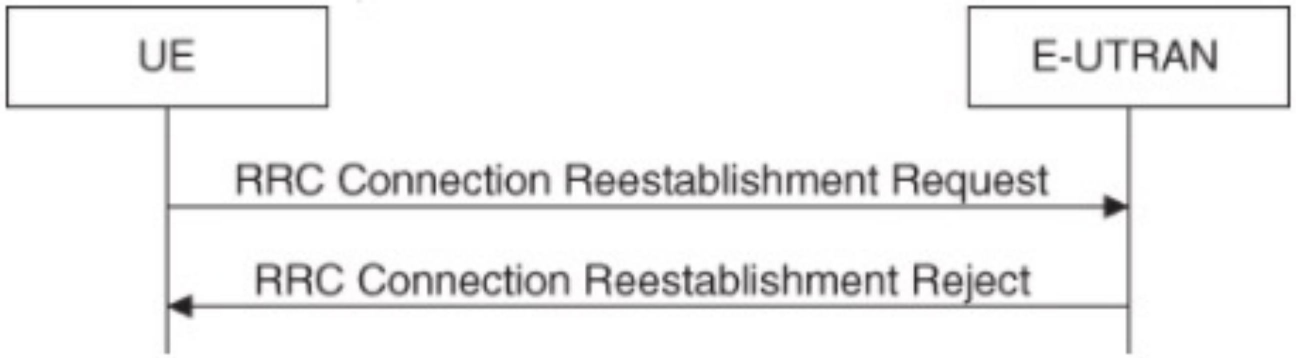
- Parameters:

RRC Connection Reestablishment Request	
ShortMAC-I	16 lsb of MAC-I (calculated with the RRC integrity key used in the source cell or the cell in case of handover, or in the cell that triggers re-establishment)
COUNT **BEARER** **DIRECTION**	All set to binary ones

RRC Connection Reestablishment	
NCC (Next hop Chaining Count)	Used to synchronize the K_{eNB}

RRC Connection re-establishment

- Upon failure, UE moves to *idle state*
- Coming back from idle to *connection state* include new C-RNTI allocation, NAS signalling and fresh key delivery from the MME



[Source: D.Forsberg et al. – LTE Security, Wiley 2012]

To remember!

1. The principles of EPS AKA
2. The advantages of key hierarchy
3. Principles to select and use cryptographic algorithms
4. Implementation in LTE