ETSI TS 133 401 V8.4.0 (2009-07)

Technical Specification

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
3GPP System Architecture Evolution (SAE);
Security architecture
(3GPP TS 33.401 version 8.4.0 Release 8)

Reference

RTS/TSGS-0333401v840

Keywords

GSM, LTE, SECURITY, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document specifies the security architecture, i.e., the security features and the security mechanisms for the Evolved Packet System and the Evolved Packet Core, and the security procedures performed within the evolved Packet System (EPS) including the Evolved Packet Core (EPC) and the Evolved UTRAN (E-UTRAN).

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.401: "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access".

[3] 3GPP TS 23.003: "Numbering, addressing and identification".

[4] 3GPP TS 33.102: "3G security; Security architecture".

[5] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[6] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[7] IETF RFC 4303: "IP Encapsulating Security Payload (ESP)".

[8] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic bootstrapping architecture".

[9] 3GPP TS 24.301: "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3".

[10] IETF RFC 2104 (1997): "HMAC: Keyed-Hashing for Message Authentication".

[11] ISO/IEC 10118-3 (2004): "Information Technology - Security techniques - Hash-functions - Part 3: Dedicated hash-functions".

[12] 3GPP TS 36.323: "Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification"

[13] 3GPP TS 31.102: "Characteristics of the Universal Subscriber Identity Module (USIM) application".

[14] 3GPP TS 35.215: "Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 1: UEA2 and UIA2 specifications"

[15] NIST: "Advanced Encryption Standard (AES) (FIPS PUB 197) "

[16] NIST Special Publication 800-38A (2001): "Recommendation for Block Cipher Modes of Operation".

[17] NIST Special Publication 800-38B (2001): "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication".

[18] Void.

[19] Void.

[20] Void.

[21] 3GPP TS 36.331:"Evolved Universal Terrestrial Radio Access (E-UTRA) Radio Resource Control (RRC); Protocol specification".

[22] 3GPP TS 23.216: "Single Radio Voice Call Continuity (SRVCC); Stage 2".

[23] 3GPP TS 22.101: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service aspects; Service principles".

[24] 3GPP TS 25.331: "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol Specification ".

[25] 3GPP TS 44.060: "3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1], in TS 33.102 [4] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Access Security Management Entity: entity which receives the top-level keys in an access network from the HSS. For E-UTRAN access networks, the role of the ASME is assumed by the MME

Activation of security context: the process of taking into use a security context.

Authentication data: Data that is part of a security context or of authentication vectors.

Chaining of K_{eNB}: derivation of a new K_{eNB} from another K_{eNB} (i.e., at cell handover)

Current EPS security context: The security context which has been activated most recently. Note that a current EPS security context originating from either a mapped or native EPS security context may exist simultaneously with a native non-current EPS security context.

ECM-CONNECTED state: This is as defined in TS 23.401 [2]. The term ECM-CONNECTED state corresponds to the term EMM-CONNECTED mode used in TS 24.301 [9].

ECM-IDLE state: As defined in TS 23.401 [2]. The term ECM-IDLE state corresponds to the term EMM-IDLE mode used in TS 24.301 [9].

EPS-Authentication Vector: $K_{ASME,}$ RAND, AUTN, XRES

EPS security context: A state that is established locally at the UE and a serving network domain. At both ends "EPS security context data" is stored, that consists of the EPS NAS security context, and the EPS AS security context.

NOTE 1: An EPS security context has type 'mapped', 'full native' or 'partial native'. Its state can either be 'current' or 'non-current'. A context can be of one type only and be in one state at a time. The state of a particular context type can change over time. A partial native context can be transformed into a full native. No other type transformations are possible.

EPS AS security context: the cryptographic keys at AS level with their identifiers, the Next Hop parameter NH, the Next Hop Chaining Counter parameter NCC used for next hop access key derivation, the identifiers of the selected AS level cryptographic algorithms and counters used for replay protection. Note that the EPS AS security context only exists when the UE is in ECM-CONNECTED state and is otherwise void.

NOTE: NH and NCC need to be stored also at the MME during connected mode.

EPS NAS security context: This context consists of K_{ASME} with the associated key set identifier, the UE security capabilities, and the uplink and downlink NAS COUNT values. In particular, separate pairs of NAS COUNT values are used for each EPS NAS security contexts, respectively. The distinction between native and mapped EPS security contexts also applies to EPS NAS security contexts. The EPS NAS security context is called 'full' if it additionally contains the keys K_{NASint} and K_{NASenc} and the identifiers of the selected NAS integrity and encryption algorithms.

Full native EPS security context: A native EPS security context for which the EPS NAS security context is full according to the above definition. A full native EPS context is either in state 'current' or state 'non-current'.

Forward security: In the context of K_{eNB} key derivation, forward security refers to the property that, for an eNB with knowledge of a $K_{eNB,}$ shared with a UE, it shall be computationally infeasible to predict any future $K_{eNB,}$ that will be used between the same UE and another eNB. More specifically, n hop forward security refers to the property that an eNB is unable to compute keys that will be used between a UE and another eNB to which the UE is connected after n or more handovers (n=1 or 2).

Legacy security context: A security context which has been established according to TS 33.102 [4].

Mapped security context: Security context created by converting the current security context in the source system to a security context for the target system in inter-system mobility, e.g., UMTS keys created from EPS keys. The EPS NAS security context of a mapped security context is full.

Native EPS security context: An EPS security context whose K_{ASME} was created by a run of EPS AKA.

Non-current EPS security context: A native EPS security context that is not the current one. A non-current EPS security context may be stored along with a current EPS security context in the UE and the MME. A non-current EPS security context does not contain an EPS AS security context. A non-current EPS security context is either of type 'full native' or of type 'partial native'.

Partial native EPS security context: A partial native EPS security context consists of K_{ASME} with the associated key set identifier, the UE security capabilities, and the uplink and downlink NAS COUNT values, which are initially set to zero. A partial native EPS security context is created by an EPS AKA, for which no corresponding successful NAS SMC has been run. A partial native context is always in state 'non-current'.

Re-derivation of NAS keys: derivation of new NAS keys from the same K_{ASME} but including different algorithms (and no freshness parameter)

Refresh of K_{eNB}: derivation of a new K_{eNB} from the same K_{ASME} and including a freshness parameter

Re-keying of K_{eNB}: derivation of a new K_{eNB} from a new K_{ASME} (i.e., after an AKA has taken place)

Re-keying of NAS keys: derivation of new NAS keys from a new K_{ASME}

UE security capabilities: The set of identifiers corresponding to the ciphering and integrity algorithms implemented in the UE. This includes capabilities for EPS AS and NAS, and includes capabilities for UTRAN and GERAN if these access types are supported by the UE.

UE EPS security capabilities: The UE security capabilities for EPS AS and NAS.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

|| Concatenation

$\oplus$ Bitwise Exclusive Or (XOR) operation

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AES	Advanced Encryption Standard
AK	Anonymity Key
AKA	Authentication and Key Agreement
AMF	Authentication Management Field
AN	Access Network
AS	Access Stratum
AUTN	Authentication token
AV	Authentication Vector
ASME	Access Security Management Entity
Cell-ID	CellIdentity as used in TS 36.331 [21]
CK	Cipher Key
CKSN	Cipher Key Sequence Number
C-RNTI	Cell RNTI as used in TS 36.331 [21]
DoS	Denial of Service
EARFCN-DL	E-UTRA Absolute Radio Frequency Channel Number-Down Link
ECM	EPS Connection Management
EEA	EPS Encryption Algorithm
EIA	EPS Integrity Algorithm
eKSI	Key Set Identifier in E-UTRAN
EMM	EPS Mobility Management
eNB	Evolved Node-B
EPC	Evolved Packet Core
EPS	Evolved Packet System
EPS-AV	EPS authentication vector
E-UTRAN	Evolved UTRAN
GERAN	GSM EDGE Radio Access Network
GUTI	Globally Unique Temporary Identity
HE	Home Environment
HFN	Hyper Frame Number
HO	Hand Over
HSS	Home Subscriber Server
IK	Integrity Key
IKE	Internet Key Exchange
IMEI	International Mobile Equiptment Identity
IMEI(SV)	IMEI (Software Version)
IMSI	International Mobile Subscriber Identity
IRAT	Inter-Radio Access Technology
ISR	Idle Mode Signaling Reduction
KDF	Key Derivation Function
KSI	Key Set Identifier
LSB	Least Significant Bit
LSM	Limited Service Mode
MAC-I	Message Authentication Code for Integrity (terminology of TS36.323 [12])
MACT	Message Authentication Code T used in AES CMAC calculation
ME	Mobile Equipment
MME	Mobility Management Entity
MS	Mobile Station
MSC	Mobile Switching Center
MSIN	Mobile Station Identification Number
NAS	Non Access Stratum
NAS-MAC	Message Authentication Code for NAS for Integrity (called MAC in TS24.301 [9])
NCC	Next hop Chaining Counter
NH	Next Hop
PCI	PhysicalCellIdentity as used in TS 36.331 [21]
PLMN	Public Land Mobile Network
PRNG	Pseudo Random Number Generator
P-TMSI	Packet- Temporary Mobile Subscriber Identity
PDCP	Packet Data Convergence Protocol
RAND	RANDom number
RAU	Routing Area Update
RRC	Radio Resource Control
SGSN	Serving GPRS Support Node

SIM	Subscriber Identity Module
SMC	Security Mode Command
SN	Serving Network
SN id	Serving Network identity
SQN	Sequence Number
SRB	Source Route Bridge
SRVCC	Single Radio Voice Call Continuity
S-TMSI	S-Temporary Mobile Subscriber Identity
TAI	Tracking Area Identity
TAU	Tracking Area Update
UE	User Equipment
UEA	UMTS Encryption Algorithm
UIA	UMTS Integrity Algorithm
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunication System
UP	User Plane
USIM	Universal Subscriber Identity Module
UTRAN	Universal Terrestrial Radio Access Network
XRES	Expected Response USIM

3.4 Conventions

All data variables in the present document are presented with the most significant substring on the left hand side and the least significant substring on the right hand side. A substring may be a bit, byte or other arbitrary length bitstring. Where a variable is broken down into a number of substrings, the leftmost (most significant) substring is numbered 0, the next most significant is numbered 1, and so on through to the least significant.

4 Overview of Security Architecture

Figure 4-1 gives an overview of the complete security architecture.

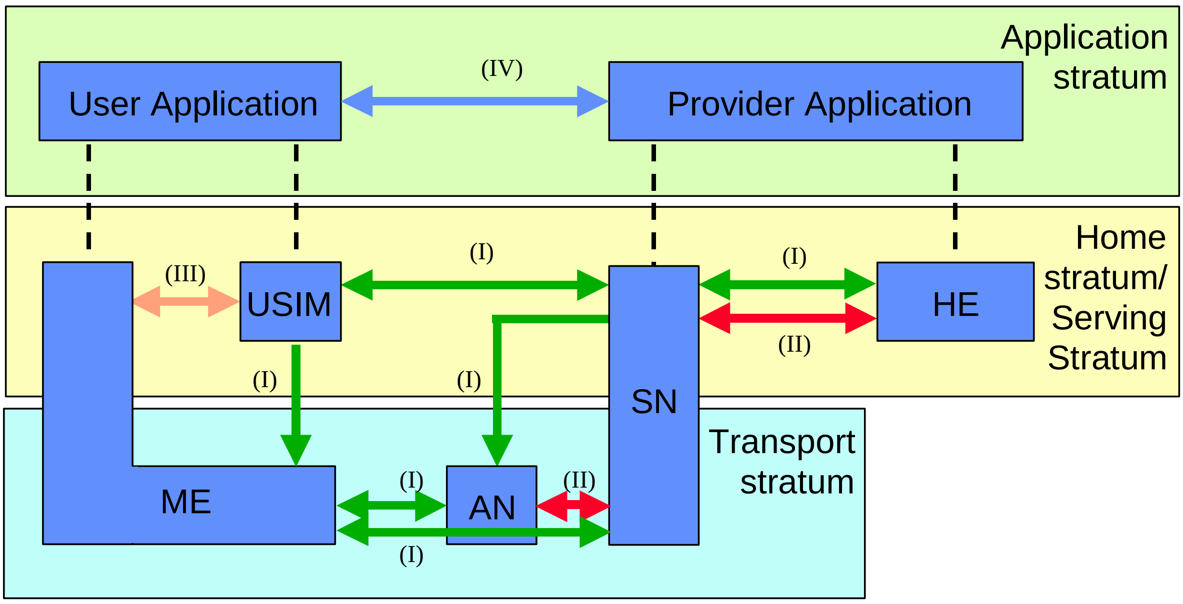

Figure 4-1: Overview of the security architecture

Five security feature groups are defined. Each of these feature groups meets certain threats and accomplishes certain security objectives:

- **Network access security (I):** the set of security features that provide users with secure access to services, and which in particular protect against attacks on the (radio) access link.

- **Network domain security (II):** the set of security features that enable nodes to securely exchange signalling data, user data (between AN and SN and within AN), and protect against attacks on the wireline network.
- **User domain security (III):** the set of security features that secure access to mobile stations.
- **Application domain security (IV):** the set of security features that enable applications in the user and in the provider domain to securely exchange messages.
- **Visibility and configurability of security (V):** the set of features that enables the user to inform himself whether a security feature is in operation or not and whether the use and provision of services should depend on the security feature.

5 Security Features

5.1 User-to-Network security

5.1.1 User identity and device confidentiality

User identity confidentiality is as defined by TS 33.102 [4] subclause 5.1.1

From subscriber's privacy point of view, the MSIN (also IMEI) should be confidentiality protected.

The UE shall provide its equipment identifier IMEI(SV) to the network, if the network asks for it.

The IMEI shall be securely stored in the terminal.

The UE shall not send IMEI(SV) to the network on a network request before the NAS security has been activated.

The IMEI(SV) shall be sent in the NAS protocol.

NOTE: In some cases, e.g., the very first attach procedure, MSIN has to be sent to network in cleartext. When NAS confidentiality protection is beyond an operator option, IMEI (SV) can not be confidentiality protected.

5.1.2 Entity authentication

Entity authentication is as defined by TS 33.102 [4] subclause 5.1.2

5.1.3 User data and signalling data confidentiality

5.1.3.1 Ciphering requirements

Ciphering may be provided to RRC-signalling to prevent UE tracking based on cell level measurement reports, handover message mapping, or cell level identity chaining. RRC signalling confidentiality is an operator option.

Synchronization of the input parameters for ciphering shall be ensured for the protocols involved in the ciphering.

The NAS signalling may be confidentiality protected. NAS signalling confidentiality is an operator option.

NOTE 1: RRC and NAS signalling confidentiality protection is recommended to be used.

User plane confidentiality protection shall be done at PDCP layer and is an operator option.

NOTE 2: User plane confidentiality protection is recommended to be used.

NOTE 3: Confidentiality protection for RRC and UP is applied at the PDCP layer, and no layers below PDCP are confidentiality protected. Confidentiality protection for NAS is provided by the NAS protocol.

5.1.3.2 Algorithm Identifier Values

All algorithms specified in this subclause are algorithms with a 128-bit input key except Null ciphering algorithm.

NOTE: Deviations from the above requirement have to be indicated explicitly in the algorithm identifier list below.

Each EPS Encryption Algorithm (EEA) will be assigned a 4-bit identifier. Currently, the following values have been defined for NAS, RRC and UP ciphering:

"0000_2" EEA0 Null ciphering algorithm

"0001_2" 128-EEA1 SNOW 3G based algorithm

"0010_2" 128-EEA2 AES based algorithm

The remaining values have been reserved for future use.

UEs and eNBs shall implement EEA0, 128-EEA1 and 128-EEA2 for both RRC signalling ciphering and UP ciphering.

UEs and MMEs shall implement EEA0, 128-EEA1 and 128-EEA2 for NAS signalling ciphering.

5.1.4 User data and signalling data integrity

5.1.4.1 Integrity requirements

Synchronization of the input parameters for integrity protection shall be ensured for the protocols involved in the integrity protection.

Integrity protection, and replay protection, shall be provided to NAS and RRC-signalling.

All NAS signaling messages except those explicitly listed in TS 24.301 [9] as exceptions shall be integrity-protected. All RRC signaling messages except those explicitly listed in TS 36.331 [21] as exceptions shall be integrity-protected.

User plane packets between the eNB and the UE shall not be integrity protected.

5.1.4.2 Algorithm Identifier Values

All algorithms specified in this subclause are algorithms with a 128-bit input key.

NOTE: Deviations from the above requirement have to be indicated explicitly in the algorithm identifier list below.

Each EPS Integrity Algorithm (EIA) will be assigned a 4-bit identifier. Currently, the following values have been defined:

"0001_2" 128-EIA1 SNOW 3G

"0010_2" 128-EIA2 AES

The remaining values have been reserved for future use.

UEs and eNBs shall implement 128-EIA1 and 128-EIA2 for RRC signalling integrity protection.

UEs and MMEs shall implement 128-EIA1 and 128-EIA2 for NAS signalling integrity protection.

5.2 Security visibility and configurability

Although in general the security features should be transparent to the user, for certain events and according to the user's concern, greater user visibility of the operation of following security feature shall be provided:

- indication of access network encryption: the property that the user is informed whether the confidentiality of user data is protected on the radio access link, in particular when non-ciphered calls are set-up;

The ciphering indicator feature is specified in 3GPP TS 22.101 [23].

Configurability is the property that the user can configure whether the use or the provision of a service should depend on whether a security feature is in operation. A service can only be used if all security features, which are relevant to that service and which are required by the configurations of the user, are in operation. The following configurability features are suggested:

- enabling/disabling user-USIM authentication: the user should be able to control the operation of user-USIM authentication, e.g., for some events, services or use.

5.3 Security requirements on eNodeB

5.3.1 General

The security requirements given in this section apply to all types of eNodeBs. More stringent requirements for specific types of eNodeBs may be defined in other documents.

5.3.2 Requirements for eNB setup and configuration

Setting up and configuring eNBs shall be authenticated and authorized so that attackers shall not be able to modify the eNB settings and software configurations via local or remote access.

1. Security associations are required between the EPS core and the eNB and between adjacent eNBs, connected via X2. These security association establishments shall be mutually authenticated and used for communication between the entities. The security associations shall be realized according to clause 11 and 12 of this specification.

2. Communication between the remote/local O&M systems and the eNB shall be mutually authenticated.

3. The eNB shall be able to ensure that software/data change attempts are authorized

4. The eNB shall use authorized data/software.

5. Sensitive parts of the boot-up process shall be executed with the help of the secure environment.

6. Confidentiality of software transfer towards the eNB shall be ensured.

7. Integrity protection of software transfer towards the eNB shall be ensured.

5.3.3 Requirements for key management inside eNB

The EPS core network provides subscriber specific session keying material for the eNBs, which also hold long term keys used for authentication and security association setup purposes. Protecting all these keys is important.

1. Keys stored inside eNBs shall never leave a secure environment within the eNB except when done in accordance with this or other 3GPP specifications.

5.3.4 Requirements for handling User plane data for the eNB

It is eNB's task to cipher and decipher user plane packets between the Uu reference pointand the S1/X2 reference points.

1. User plane data ciphering/deciphering shall take place inside the secure environment where the related keys are stored.
2. The transport of user data over S1-U and X2-U shall be integrity, confidentially and replay-protected from unauthorized parties. If this is to be accomplished by cryptographic means, clause 12 shall be applied.

NOTE: The use of cryptographic protection on S1-U and X2-U is an operator's decision. In case the eNB has been placed in a physically secured environment then the 'secure environment' may include other nodes and links beside the eNB.

5.3.4a Requirements for handling Control plane data for the eNB

It is eNB's task to provide confidentiality and integrity protection for control plane packets on the S1/X2 reference points.

1. Control plane data ciphering/deciphering shall take place inside the secure environment where the related keys are stored.
2. The transport of control plane data over S1-MME and X2-C shall be applied to integrity-, confidentiality- and replay-protected from unauthorized parties. If this is to be accomplished by cryptographic means, clause 11 shall be applied.

NOTE: The use of cryptographic protection on S1-MME and X2-C is an operator's decision. In case the eNB has been placed in a physically secured environment then the 'secure environment' may include other nodes and links beside the eNB.

5.3.5 Requirements for secure environment of the eNB

The secure environment is logically defined within the eNB and is a composition of functions for the support of sensitive operations.

1. The secure environment shall support secure storage of sensitive data, e.g. long term cryptographic secrets and vital configuration data.
2. The secure environment shall support the execution of sensitive functions, e.g. en-/decryption of user data and the basic steps within protocols which use long term secrets (e.g. in authentication protocols).
3. Sensitive data used within the secure environment shall not be exposed to external entities.
4. The secure environment shall support the execution of sensitive parts of the boot process.
5. The secure environment's integrity shall be assured.
6. Only authorised access shall be granted to the secure environment, i.e. to data stored and used within, and to functions executed within.

5.4 Void

6 Security Procedures between UE and EPC Network Elements

6.1 Authentication and key agreement

6.1.1 AKA procedure

EPS AKA is the authentication and key agreement procedure that shall be used over E-UTRAN.

A Rel-99 or later USIM application on a UICC shall be sufficient for accessing E-UTRAN, provided the USIM application does not make use of the separation bit of the AMF in a way described in TS 33.102 [4] Annex F. Access to E-UTRAN with a 2G SIM or a SIM application on a UICC shall not be granted.

An ME that has E-UTRAN radio capability shall support the USIM-ME interface as specified in TS 31.102 [13]

EPS AKA shall produce keying material forming a basis for user plane (UP), RRC, and NAS ciphering keys as well as RRC and NAS integrity protection keys.

NOTE 1: Key derivation requirements of AS and NAS keys can be found in subclause 7.2.1

During the authentication, the USIM shall verify the freshness of the authentication vector that is used. The MME sends to the USIM via ME the random challenge RAND and an authentication token AUTN for network authentication from the selected authentication vector. At receipt of this message, the USIM shall verify whether AUTN can be accepted and if so, produces a response RES. USIM shall compute CK and IK.

An ME accessing E-UTRAN shall check during authentication that the "separation bit" in the AMF field of AUTN is set to 1 and reject authentication otherwise with a CAUSE value. The "separation bit" is bit 0 of the AMF field of AUTN.

UE shall compute K_{ASME} from CK, IK, and serving network's identity (SN id) using the KDF as specified in Annex A. SN id binding implicitly authenticates the serving network's identity when the derived keys from K_{ASME} are successfully used.

NOTE 2: This separation bit in the AMF can not be used anymore for operator specific purposes as described by TS 33.102 [4], Annex F

NOTE 3: The HSS needs to ensure that the MME requesting the authentication data is entitled to use the SN id used to calculate K_{ASME}. The exact details of how to achieve this are not covered in this specification.

NOTE 4: If the keys CK, IK resulting from an EPS AKA run were stored in the fields already available on the USIM for storing keys CK and IK this could lead to overwriting keys resulting from an earlier run of UMTS AKA. This would lead to problems when EPS security context and UMTS security context were held simultaneously (as is the case when security context is stored e.g. for the purposes of Idle Mode Signaling Reduction). Therefore, "plastic roaming" where a UICC is inserted into another ME will necessitate an EPS AKA authentication run if the USIM does not support EMM parameters storage.

UE shall respond with User authentication response message including RES in case of successful AUTN verification as described in TS 33.102[4] and successful AMF verification as described above. Otherwise UE shall send User authentication reject message with a proper CAUSE value.

Figure 6.1.1-1 describes EPS AKA procedure, which is based on UMTS AKA (see TS 33.102[4]). The following keys are shared between UE and HSS:

- **K** is the permanent key stored on the USIM on a UICC and in the Authentication Centre AuC.
- **CK, IK** is the pair of keys derived in the AuC and on the USIM during an AKA run. CK, IK shall be handled differently depending on whether they are used in an EPS security context or a legacy security context, as described in subclause 6.1.2.

As a result of the authentication and key agreement, an intermediate key $\mathbf{K_{ASME}}$ shall be generated which is shared between UE and ASME i.e. the MME cfr Figure 6.1.1-1. How this is done is described in subclause 6.1.2.

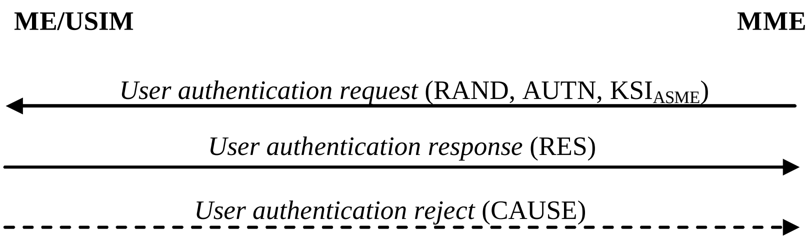

Figure 6.1.1-1: EPS user authentication (EPS AKA)

6.1.2 Distribution of authentication data from HSS to serving network

The purpose of this procedure is to provide the MME with one or more EPS authentication vectors (RAND, AUTN, XRES, K_{ASME}) from the user's HE (HSS) to perform a number of user authentications.

NOTE 1: It is recommended that the MME fetch only one EPS authentication vector at a time as the need to perform AKA runs has been reduced in EPS through the use of a more elaborate key hierarchy. In particular, service requests can be authenticated using a stored K_{ASME} without the need to perform AKA. Furthermore, the sequence number management schemes in TS 33.102, Annex C [4], designed to avoid re-synchronisation problems caused by interleaving use of batches of authentication vectors, are only optional. Re-synchronisation problems in EPS can be avoided, independently of the sequence number management scheme, by immediately using an authentication vector retrieved from the HSS in an authentication procedure between UE and MME.

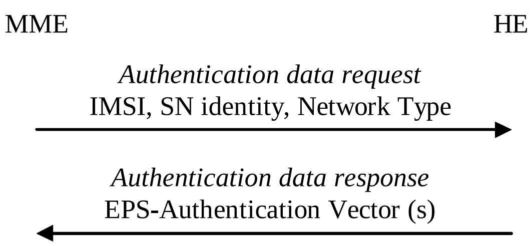

Figure 6.1.2-1: Distribution of authentication data from HE to MME

An EPS authentication vector is derived from the authentication vector defined in TS 33.102 [4] clause 6.3.2. To derive the key K_{ASME} in the HE, the KDF as specified in Annex A is used which shall contain following mandatory input parameters: CK, IK and SN identity.

If the Network Type equals E-UTRAN then the "separation bit" in the AMF field of AUTN shall be set to 1 to indicate to the UE that the authentication vector is only usable for AKA in an EPS context, if the "separation bit" is set to 0, the vector is usable in a non-EPS context only (e.g. GSM, UMTS). For authentication vectors with the "separation bit" set to 1, the secret keys CK and IK generated during AKA shall never leave the HSS.

The MME invokes the procedures by requesting authentication vectors from the HE (Home environment).

The *authentication data request* shall include the IMSI, the Serving Network identity i.e. MCC + MNC, and the Network Type (I.e. E-UTRAN)

Upon the receipt of the *authentication data request* from the MME, the HE may have pre-computed the required number of EPS authentication vectors and retrieve them from the HSS database or may compute them on demand.

NOTE 2: For K_{ASME} the possibilities for pre-computation are restricted due to the PLMN-binding.

The HE sends an authentication response back to the MME that contains the requested information. If multiple EPS authentication vectors had been requested then they are ordered based on their sequence numbers.

6.1.3 User identification by a permanent identity

The user identification mechanism should be invoked by the serving network whenever the user cannot be identified by means of a temporary identity (GUTI). In particular, it should be used when the serving network cannot retrieve the IMSI based on the GUTI by which the user identifies itself on the radio path.

The mechanism described in figure 6.1.3-1 allows the identification of a user on the radio path by means of the permanent subscriber identity (IMSI).

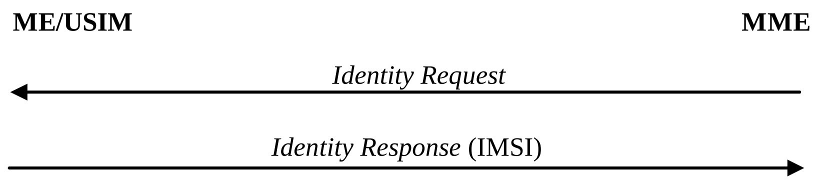

Figure 6.1.3-1: User identity query

The mechanism is initiated by the MME that requests the user to send its permanent identity. The user's response contains the IMSI in cleartext. This represents a breach in the provision of user identity confidentiality.

6.1.4 Distribution of IMSI and authentication data within one serving network domain

The purpose of this procedure is to provide a newly visited MME with authentication data from a previously visited MME within the same serving network domain.

NOTE: The following procedure in this clause is based on TAU procedure and it can also be applied for Attach procedure where all the corresponding texts for 'TAU' in the following procedure should be replaced with 'Attach'.

The procedure is shown in Figure 6.1.4-1

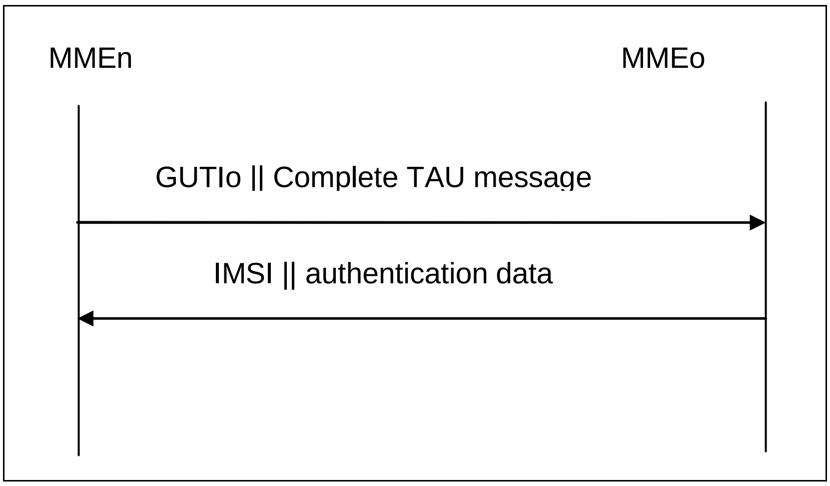

Figure 6.1.4-1: Distribution of IMSI and authentication data within one serving domain

The procedure shall be invoked by the newly visited MMEn after the receipt of a Tracking Area update request from the user wherein the user is identified by means of a temporary user identity GUTIo and the Tracking area identity TAIo under the jurisdiction of a previously visited MMEo that belongs to the same serving network domain as the newly visited MMEn.

The protocol steps are as follows:

a) The MMEn sends a message to the MMEo, this message contains GUTIo and the received TAU message.

b) The MMEo searches the user data in the database and checks the integrity protection on the TAU message.

If the user is found and the integrity check succeeds, the MMEo shall send a response back that:

i) shall include the IMSI,

ii) may include a number of unused EPS-authentication vectors ordered on a first-in / first-out basis, and

iii) may include any EPS security contexts it holds

The MMEo subsequently deletes the EPS-authentication vectors and any EPS security contexts which have been sent.

If the user cannot be identified or the integrity check fails, then the MMEo shall send a response indicating that the user identity cannot be retrieved.

c) If the MMEn receives a response with an IMSI, it creates an entry and stores any EPS-authentication vectors and any EPS security context that may be included.

If the MMEn receives a response indicating that the user could not be identified, it shall initiate the user identification procedure described in clause 6.1.3.

6.1.5 Distribution of IMSI and authentication data between different serving network domains

In general, the distribution of IMSI and authentication between MMEs belonging to different serving network domains of shall be performed as described for the distribution of IMSI and authentication data within the same service network domain in subclause 6.1.4. In particular, the current EPS security context data may be transferred between MMEs belonging to different serving network domains. However, the following three cases are exceptions related to the distribution of authentication vectors between SGSNs and MME's:

a) MME to MME

Unused EPS authentication vectors shall not be distributed between MME's belonging to different serving domains (PLMNs)

UMTS authentication vectors that were previously received from an SGSN shall not be forwarded between MME's.

b) SGSN to MME

An SGSN may forward unused UMTS authentication vectors to an MME.

An MME shall not use unused UMTS authentication vectors forwarded from an SGSN in E-UTRAN procedures.

c) MME to SGSN

UMTS AVs which were previously stored in the MME may be forwarded back towards the same SGSN.

UMTS AVs which were previously stored in the MME shall not be forwarded towards other SGSNs.

EPS authentication vectors shall not be forwarded from an MME towards an SGSN.

NOTE: This is due to the fact that in an EPS-AV the CK and IK are not available for the MME and hence also not for the SGSN when an EPS-AV would be forwarded.

6.2 EPS key hierarchy

Requirements on EPC and E-UTRAN related to keys:

a) The EPC and E-UTRAN shall allow for use of encryption and integrity protection algorithms for AS and NAS protection having keys of length 128 and for future use the network interfaces shall be prepared to support 256 bit keys.

b) The keys used for UP, NAS and AS protection shall be dependent on the algorithm with which they are used.

c) As part of the initial attach request from the UE, ME shall signal the UE security capabilities to the MME.

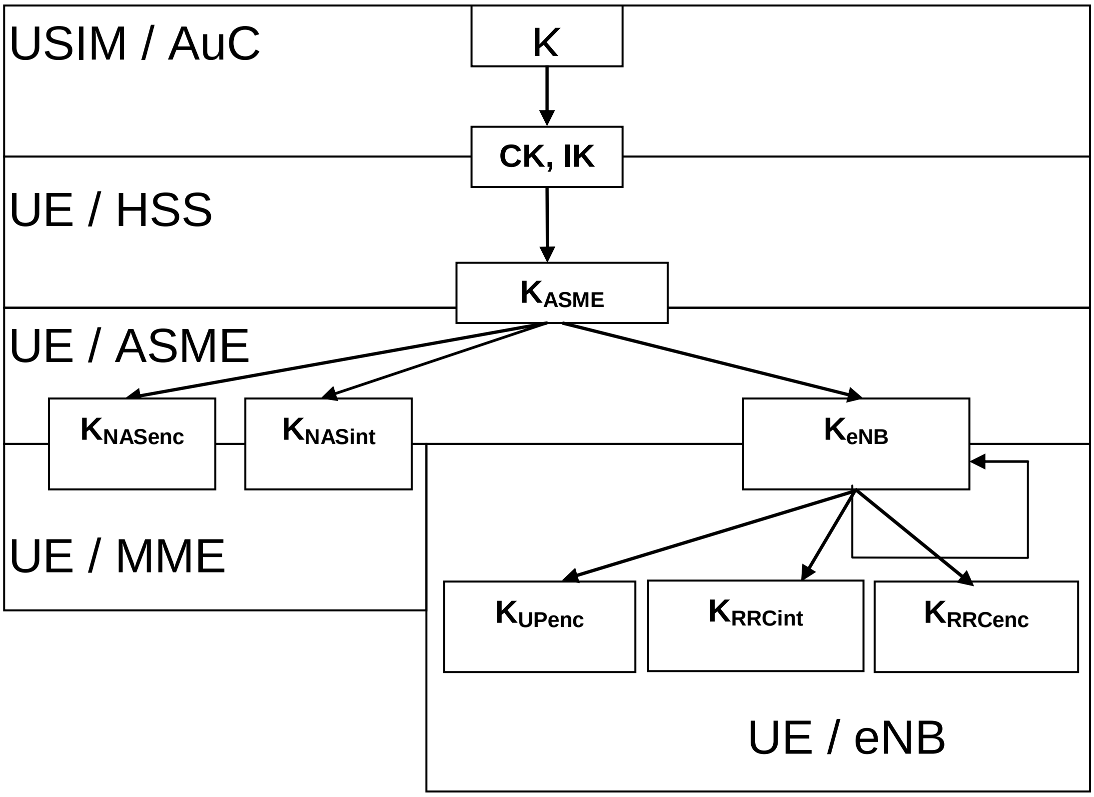

Figure 6.2-1: Key hierarchy in E-UTRAN

The key hierarchy (see Figure 6.2-1) includes following keys: $\mathbf{K_{eNB}}$, $\mathbf{K_{NASint}}$, $\mathbf{K_{NASenc}}$, $\mathbf{K_{UPenc}}$, $\mathbf{K_{RRCint}}$ and $\mathbf{K_{RRCenc}}$

- $\mathbf{K_{eNB}}$ is a key derived by UE and MME from K_{ASME} when the UE goes to ECM-CONNECTED state or by UE and target eNB during eNB handover.

Keys for NAS traffic:

- $\mathbf{K_{NASint}}$ is a key, which shall only be used for the protection of NAS traffic with a particular integrity algorithm This key is derived by UE and MME from $K_{ASME,}$ as well as an identifier for the integrity algorithm using the KDF as specified in Annex A.

- $\mathbf{K_{NASenc}}$ is a key, which shall only be used for the protection of NAS traffic with a particular encryption algorithm. This key is derived by UE and MME from $K_{ASME,}$ as well as an identifier for the encryption algorithm using the KDF as specified in Annex A.

Keys for UP traffic:

- $\mathbf{K_{UPenc}}$ is a key, which shall only be used for the protection of UP traffic with a particular encryption algorithm. This key is derived by UE and eNB from K_{eNB}, as well as an identifier for the encryption algorithm using the KDF as specified in Annex A.

Keys for RRC traffic:

- $\mathbf{K_{RRCint}}$ is a key, which shall only be used for the protection of RRC traffic with a particular integrity algorithm. $\mathbf{K_{RRCint}}$ is derived by UE and eNB from K_{eNB}, as well as an identifier for the integrity algorithm using the KDF as specified in Annex A.

- $\mathbf{K_{RRCenc}}$ is a key, which shall only be used for the protection of RRC traffic with a particular encryption algorithm. K_{RRCenc} is derived by UE and eNB from K_{eNB} as well as an identifier for the encryption algorithm using the KDF as specified in Annex A.

Intermediate keys:

- **NH** is a key derived by UE and MME to provide forward security as described in clause 7.2.8. The NH is sent by the MME to the eNB using S1signalling.
- $\mathbf{K_{eNB}*}$ is a key derived by UE and eNB when performing an horizontal or vertical key derivation as specified in clause 7.2.8 using a KDF as specified in Annex A.

Figure 6.2-2 shows the dependencies between the different keys, and how they are derived from the network nodes point of view. Figure 6.2-3 shows the corresponding relations and derivations as performed in the ME.

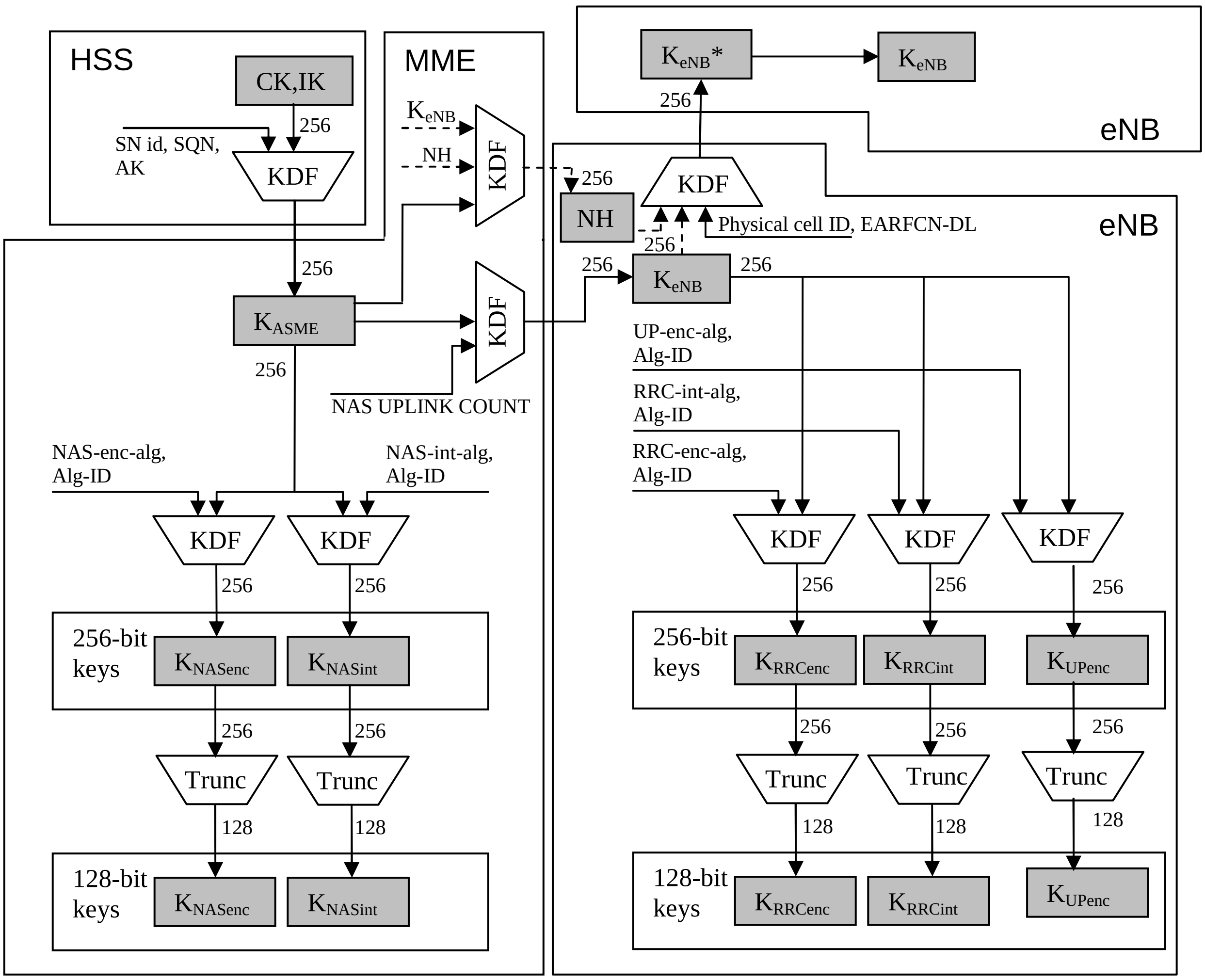

Figure 6.2-2: Key distribution and key derivation scheme for EPS (in particular E-UTRAN) for network nodes. The basic derivations are covered in the figure, but derivations performed at, e.g. inter-RAT mobility is not shown. Two dashed input to a KDF means one of the input is taken depending on condition met.

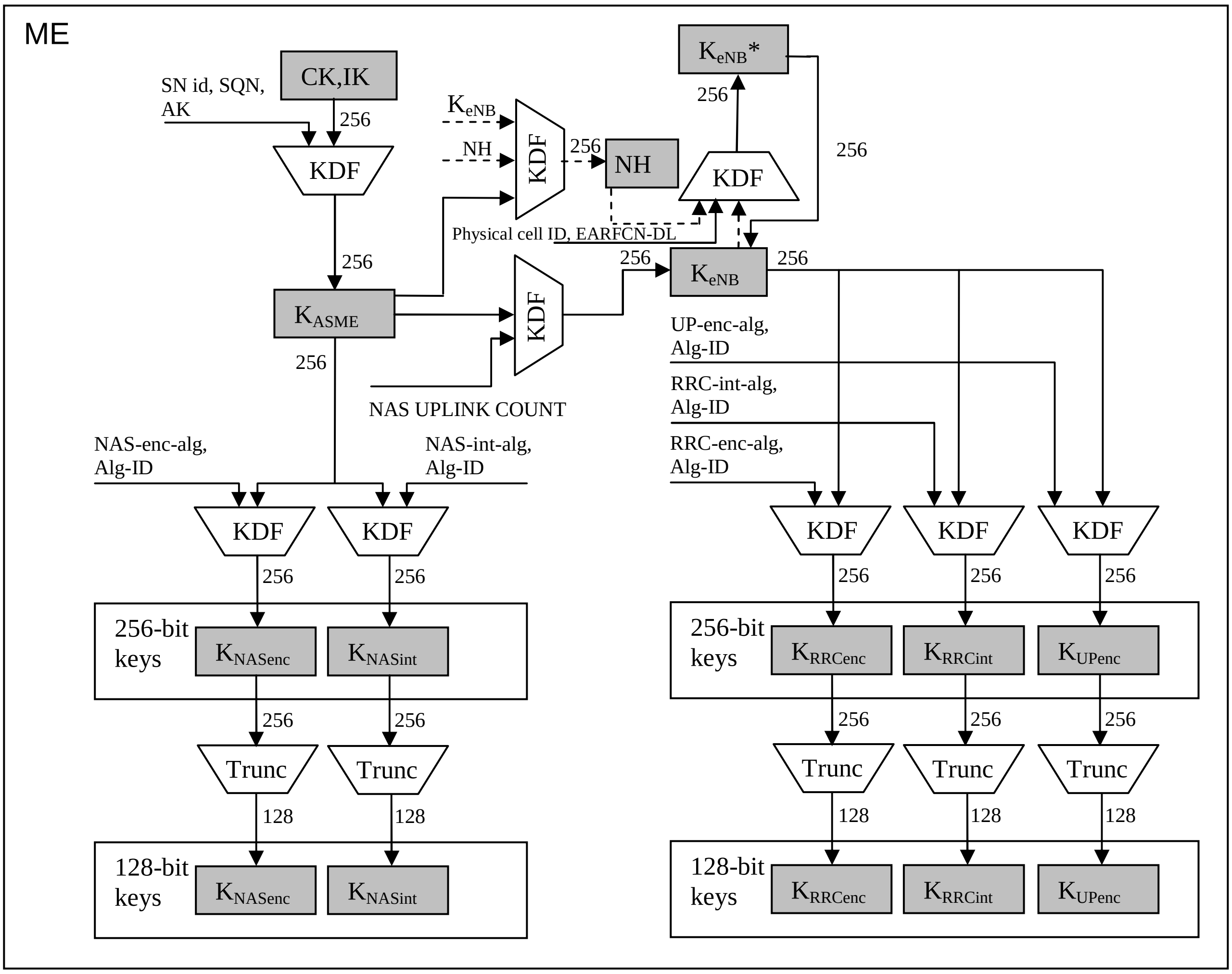

Figure 6.2-3: Key derivation scheme for EPS (in particular E-UTRAN) for the ME. The basic derivations are covered in the figure, but derivations performed at, e.g. inter-RAT mobility is not shown. Two dashed input to a KDF means one of the input is taken depending on condition met.

As the figures 6.2-2 and 6.2-3 show, the length of K_{ASME} and K_{eNB} is 256 bits, the length of NH is 256bits, and 256-bit NAS, UP and RRC keys are always derived from K_{ASME} and K_{eNB} respectively. In case the encryption or integrity algorithm used to protect NAS, UP or RRC requires a 128-bit key as input, the key is truncated and the 128 least significant bits are used.

NOTE: Figures 6.2-2 and 6.2-3 do not include the key handling branches for forward security. This is described in clause 7.2.8 and Figure 7.2.8.1-1.

The function Trunc takes as input a 256-bit string, and returns a truncated output as defined in Annex A.7.

6.3 EPS key identification

The key K_{ASME} shall be identified by the key set identifier eKSI. eKSI may be either of type KSI_{ASME} or of type KSI_{SGSN}. An eKSI shall be stored in the UE and the MME together with K_{ASME} and the temporary identifier GUTI, if available.

NOTE 1: the GUTI points to the MME where the K_{ASME} is stored.

The key set identifier KSI_{ASME} is a parameter which is associated with the K_{ASME} derived during EPS AKA authentication. The key set identifier KSI_{ASME} is allocated by the MME and sent with the authentication request message to the mobile station where it is stored together with the K_{ASME}.

The key set identifier KSI_{SGSN} is a parameter which is associated with the mapped K_{ASME} derived from UMTS keys during inter-RAT mobility, cf. clauses 9 and 10 of the present specification. The key set identifier KSI_{SGSN} is generated in both the UE and the MME respectively when deriving the mapped K_{ASME} during idle procedures in E-UTRAN and during handover from GERAN/UTRAN to E-UTRAN. The KSI_{SGSN} is stored together with the mapped K_{ASME}.

The purpose of the KSI_{ASME} is to make it possible for the UE and the MME to identify a native K_{ASME} without invoking the authentication procedure. This is used to allow re-use of the K_{ASME} during subsequent connection set-ups.

The purpose of the KSI_{SGSN} is to make it possible for the UE and the MME to indicate the use of the mapped K_{ASME} in inter-RAT mobility procedures. For details cf. clauses 9 and 10.

KSI_{ASME} and KSI_{SGSN} have the same format. The format of eKSI shall allow a recipient of such a parameter to distinguish whether the parameter is of type 'KSI_{ASME}' or of type 'KSI_{SGSN}'. The format shall further contain a value field. The value fields of KSI_{ASME} nd KSI_{SGSN} are three bits each. Seven values are used to identify the key set. A value of '111' is used by the mobile station to indicate that a valid K_{ASME} is not available for use. Format of eKSI is described in [9].

At deletion of the K_{ASME}, the KSI_{ASME} is set to '111'. The value '111' in the other direction from network to mobile station is reserved.

NOTE 2: In addition to EPS security contexts, the UE may also cache UMTS security contexts. These UMTS security contexts are identified by the KSI, as defined in TS 33.102 [4].

6.4 Handling of EPS security contexts

Any EPS security context shall be deleted from the ME if:

a) the UICC is removed from the ME when the ME is in power on state;

b) the ME is powered up and the ME discovers that a UICC different from the one which was used to create the EPS security context has been inserted to the ME;

c) the ME is powered up and the ME discovers that no UICC has been inserted to the ME.

K_{ASME} shall never be transferred from the EPC to an entity outside the EPC.

Both the UE and MME shall be capable of storing one non-current EPS security context and one current EPS security context in volatile memory. In addition, while connected to E-UTRAN the UE and MME shall be capable of storing in volatile memory the NCC, NH and the related K_{ASME} used to compute keying material for the current EPS AS security context.

Any successful run of an EPS AKA creates, by the definition in clause 3, a partial native EPS security context. This context shall overwrite any existing non-current EPS security context.

After a successful run of a NAS SMC relating to the eKSI associated with an EPS security context, this context becomes the current EPS security context and shall overwrite any existing current EPS security context.

The rules for handling security contexts at transition to EMM-DEREGISTRED are given in clause 7.2.5.1. The rules for handling security contexts after a handover to E-UTRAN are given in clause 9.2.2.1.

Storage of the EPS NAS security context, excluding the keys K_{NASint} and K_{NASenc}, in the UE during power-off:

a) If the ME does not have a full native EPS NAS security context in volatile memory, any existing native EPS NAS security context stored on the UICC or in non-volatile memory of the ME shall be marked as invalid.

b) If the USIM supports EMM parameters storage, then the ME shall store the full native EPS NAS security context parameters on the USIM, mark the native EPS NAS security context on the USIM as valid, and not keep any native EPS NAS security context in non-volatile ME memory.

c) If the USIM does not support EMM parameters storage, then the ME shall store the full native EPS NAS security context in a non-volatile part of its memory, and mark the native EPS NAS security context in its non-volatile memory as valid.

After power-on of the ME, the ME shall retrieve native EPS NAS security context stored on the USIM if the USIM supports EMM parameters storage and if the stored native EPS NAS security context on the USIM is marked as valid. If the USIM does not support EMM parameters storage the ME shall retrieve the stored native EPS NAS security context from its non-volatile memory if the native EPS NAS security context is marked as valid. The ME shall derive the K_{NASint} and K_{NASenc} after retrieving the stored EPS NAS security context; see Annex A on NAS key derivation. If the ME cannot retrieve native EPS NAS security context from any of these two places, the ME shall signal "no key available" in the Attach Request. The retrieved native security context shall be the current EPS security context.

NOTE: Only native EPS NAS security context is stored in the EMM parameters file on the USIM or in non-volatile ME memory. A mapped EPS NAS security context is never stored in these two places.

7 Security Procedures between UE and EPC Access Network Elements

7.1 Mechanism for user identity confidentiality

The MME shall allocate a GUTI to a UE in order to support the subscriber identity confidentiality. The GUTI is defined in TS 23.003 [3].

S-TMSI, the shortened form of the GUTI, is used to support the subscriber identity confidentiality with more efficient radio signalling procedures (e.g. paging and Service Request). A new GUTI shall be sent to the UE only after a successful activation of NAS security.

7.2 Handling of user-related keys in E-UTRAN

7.2.1 E-UTRAN key setting during AKA

Authentication and key setting are triggered by the authentication procedure. Authentication and key setting may be initiated by the network as often as the network operator wishes. Key setting can occur as soon as the identity of the mobile subscriber (i.e. GUTI or IMSI) is known by the MME. Key K_{ASME} is stored in the MME and key K_{eNB} is derived using the KDF as specified in Annex A from the key K_{ASME} and transferred to the UE's serving eNB when needed. K_{ASME} is stored in the UE and MME and updated with the next authentication procedure.

The RRC and UP keys are derived from the K_{eNB} using the KDF as specified in Annex A when needed.

If an authentication procedure is performed while the UE is in ECM-CONNECTED state, the new NAS keys shall be taken in use in the MME and the UE by means of the NAS security mode set-up procedure (see subclause 7.2.4). The AS keys shall be taken into use with the next AS security mode set-up procedure (see subclause 7.2.4) or with the key change on the fly procedure (see subclause 7.2.9).

7.2.2 E-UTRAN key identification

Clause 6.3 of this specification states how the key K_{ASME} is identified, namely by the key set identifier eKSI. Keys K_{NASenc} and K_{NASint} in the E-UTRAN key hierarchy specified in clause 6.2, which are derived from K_{ASME}, can be uniquely identified by eKSI together with those parameters from the set {algorithm distinguisher, algorithm identifier}, which are used to derive these keys from K_{ASME} according to Annex A.

The intermediate key NH as defined in clause 7 can be uniquely determined by the key set identifier, eKSI, together with the initial K_{eNB} derived from the current NAS security context for use during the ongoing CONNECTED state and a counter counting how many NH-derivations have already been performed from this initial K_{eNB}.according to Annex A.4. The next hop chaining count, NCC, represents the 3 least significant bits of this counter..

Intermediate key K_{eNB}*, defined in clause 7, as well as keys K_{eNB}, K_{RRCint}, K_{RRCenc}, and K_{UPenc} in the E-UTRAN key hierarchy specified in clause 6.2, which are derived from K_{ASME}, can be uniquely identified by eKSI together with those parameters from the set {uplink NAS COUNT, algorithm distinguisher, algorithm identifier, NCC, and sequence of PCIs and EARFCN-DLs used in horizontal key derivations with this NCC}, which are used to derive these keys from K_{ASME} according to clause 7 and Annex A.

It is specified in the remainder of clause 7, as well as in clause 9 and 10, which of the above parameters need to be included in a security-relevant message to allow the entity receiving the message to uniquely identify a certain key.

7.2.3 E-UTRAN key lifetimes

All E-UTRAN keys are derived based on a K_{ASME}. The key hierarchy which is described in clause 6.2 does not allow direct update to RRC and UP keys, but fresh RRC and UP keys are derived based on a fresh $K_{eNB,}$ which is bound to certain dynamic parameters (like PCI) and fresh key derivation parameter(s) in state transitions (like NAS uplink COUNT). This results as fresh RRC and UP keys in the eNB between inter-eNB handovers and state transitions (see subclauses 7.2.6 to 7.2.8).. The handling (creation, modification and update) of the E-UTRAN keys in the various state transitions is described in clauses 7.2.5, 7.2.6, 7.2.7 and 7.2.8.

K_{ASME} shall be created only by running a succesful AKA or by the inter-RAT procedures towards E-UTRAN (cfr clauses 9 and 10). In case K_{ASME} is invalidated by the UE, a KSI_{ASME} with value "111" shall be sent by the UE to the network, which can initiate (re-)authentication procedure to get a new K_{ASME} based on a successful AKA authentication.

7.2.4 Security mode command procedure and algorithm negotiation

7.2.4.1 Requirements for algorithm selection

a) An active UE and a serving network shall agree upon algorithms for

- RRC ciphering and RRC integrity protection (to be used between UE and eNB)
- UP ciphering (to be used between UE and eNB)
- NAS ciphering and NAS integrity protection (to be used between UE and MME)

b) The serving network shall select the algorithms to use dependent on

- the UE security capabilities of the UE,
- the configured allowed list of security capabilities of the currently serving network entity

c) The same set of ciphering and integrity algorithms shall be supported by the UE both for AS and NAS level.

d) Each selected algorithm shall be acknowledged to the UE in an integrity protected way such that the UE is ensured that the algorithm selection was not manipulated, i.e. that the UE security capabilities were not bidden down.

e) The UE security capabilities the ME sent to the network shall be repeated in an integrity protected NAS level message to the ME such that "bidding down attacks" against the UE's security capabilities can be detected by the ME. The UE security capabilities apply to both AS and NAS level security.

f) Separate AS and NAS level security mode command procedures are required. AS level security mode command procedure configures AS security (RRC and UP) and NAS level security mode command procedure configures NAS security.

   a. Both integrity protection and ciphering for RRC are activated within the same AS SMC procedure, but not necessarily within the same message.

   b. User plane ciphering is activated at the same time as RRC ciphering.

g) It shall be possible that the selected AS and NAS algorithms are different at a given point of time.

7.2.4.2 Procedures for AS algorithm selection

7.2.4.2.1 Initial AS security context establishment

Each eNB shall be configured via network management with lists of algorithms which are allowed for usage. There shall be one list for integrity algorithms, and one for ciphering algorithms. These lists shall be ordered according to a priority decided by the operator. When AS security context is established in the eNB, the MME shall send the UE EPS security capabilities to the eNB. The eNB shall choose the ciphering algorithm which has the highest priority from its configured list and is also present in the UE EPS security capabilities. The eNB shall choose the integrity algorithm which has the highest priority from its configured list and is also present in the UE EPS security capabilities. The chosen algorithms shall be indicated to the UE in the AS SMC. The ciphering algorithm is used for ciphering of the user plane and RRC traffic. The integrity algorihtm is used for integrity protection of the RRC traffic.

7.2.4.2.2 X2-handover

At handover from a source eNB over X2 to a target eNB, the source eNB shall include the UE EPS security capabilities in the handover request message. The target eNB shall select the algorithm with highest priority from the UE EPS security capabilities according to the prioritized locally configured list of algorithms (this applies for both integrity and ciphering algorithms). The chosen algorithms shall be indicated to the UE in the handover command. In the path-switch message, the target eNB shall send the UE EPS security capabilities received from the source eNB to the MME. The MME shall verify that the UE EPS security capabilities received from the eNB are the same as the UE EPS security capabilities that the MME has stored. If there is a mismatch, the MME may log the event and may take additional measures, such as raising an alarm.

7.2.4.2.3 S1-handover

At handover from a source eNB to a target eNB over S1 (possibly including an MME change and hence a transfer of the UE security capabilities from source MME to target MME), the target MME shall send the UE EPS security capabilities to the target eNB in the S1 AP HANDOVER REQUEST message. The target eNB shall select the algorithm with highest priority from the UE EPS security capabilities according to the prioritized locally configured list of algorithms (this applies for both integrity and ciphering algorithms). The chosen algorithms shall be indicated to the UE in the handover command.

7.2.4.2.4 Intra-eNB handover

It is not required to change the AS security algorithm during intra-eNB handover.

7.2.4.3 Procedures for NAS algorithm selection

7.2.4.3.1 Initial NAS security context establishment

Each MME shall be configured via network management with lists of algorithms which are allowed for usage. There shall be one list for NAS integrity algorithms, and one for NAS ciphering algorithms. These lists shall be ordered according to a priority decided by the operator.

When the NAS security context is established, e.g., by a TAU Accept, Attach Accept or by NAS SMC message, the MME shall choose one NAS ciphering algorithm and one NAS integrity protection algorithm, and indicate them in the corresponding integrity protected message to UE and shall also include the UE security capabilities into that message. The UE shall verify that the message from the MME contains the correct UE security capabilities. If so, the UE shall reply with an integrity protected SMC Complete message, protected by the integrity algorithm selected by the MME. This enables detection of attacks where an attacker has modified the UE security capabilities in the initial NAS message. The MME shall select the NAS algorithms which have the highest priority according to the ordered lists.

7.2.4.3.2 MME change

In case there is change of MMEs and algorithms to be used for NAS, the target MME shall send an integrity protected Security Mode Command message which includes integrity and ciphering algorithms selected for NAS protection and UE EPS security capabilities. The UE verifies that the message from the MME contains the correct UE security capabilities. If so, the UE shall reply with an integrity protected SMC Complete message. The MME shall select the NAS algorithms which have the highest priority according to the ordered lists (see 7.2.4.3.1).

NOTE: After an S1-handover with MME change a TAU procedure is executed. The same is true for an inter-RAT handover to E-UTRAN and for both inter- and intra-RAT idle mode mobility resulting in a change of MMEs.

7.2.4.4 NAS security mode command procedure

The NAS SMC procedure consists of a roundtrip of messages between MME and UE. The MME sends the NAS security mode command to the UE and the UE replies with the NAS security mode complete message.

The NAS security mode command message from MME to UE shall contain the replayed UE security capabilities, the selected NAS algorithms, the eKSI for identifying K_{ASME}, and both NONCEue and NONCEmme in case as specified in section 9.1.2. This message shall be integrity protected with NAS integrity key based on K_{ASME} indicated by the eKSI in the message. See figure 7.2.4.4-1.

UE shall verify the integrity of the NAS security mode command message. If successfully verified, UE shall start NAS integrity protection and ciphering/deciphering and sends the NAS security mode complete message to MME ciphered and integrity protected with the selected NAS algorithm indicated in the NAS security mode command message and NAS keys based on K_{ASME} indicated by the eKSI in the NAS security mode command message.

NAS downlink ciphering at the MME shall start after receiving the NAS security mode complete message. NAS uplink deciphering at the MME starts after sending the NAS security mode command message. NAS uplink ciphering and downlink deciphering at the UE shall start after receiving and successfully verifying the NAS security mode command message. The NAS security mode complete message shall include IMEI in case MME requested it in the NAS SMC Command message.

If any verification of the NAS security mode command is not successful in the ME, the ME shall reply with an unprotected NAS security mode reject message (see TS 24.301 [9]).

Only after EPS AKA the NAS security mode command message shall reset NAS uplink and downlink COUNT values. Both the NAS security mode command and NAS security mode complete messages are protected based on reset COUNT values (zero). NAS SMC always changes the NAS keys (i.e. due to EPS AKA with new K_{ASME} and eKSI or due to the algorithms change).

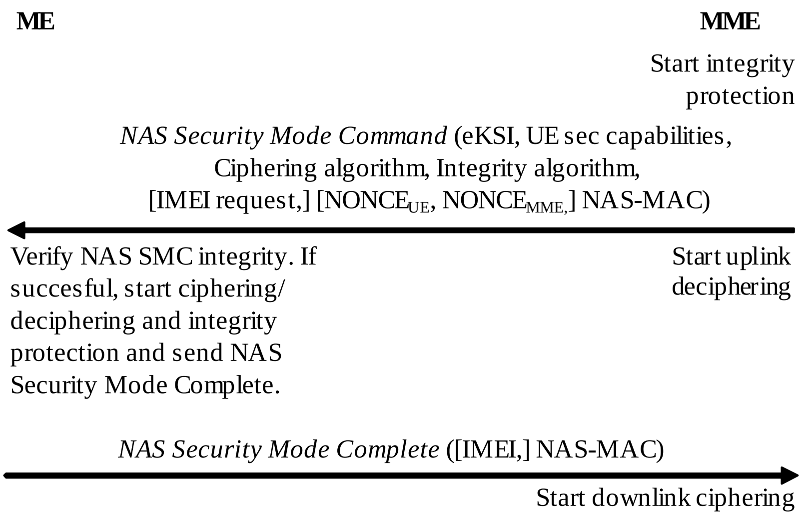

Figure 7.2.4.4-1: NAS security mode command procedure

7.2.4.5 AS security mode command procedure

The AS SMC procedure consists of a roundtrip of messages between eNB and UE. The eNB sends the AS security mode command to the UE and the UE replies with the AS security mode complete message. See figure 7.2.4.5-1.

The AS security mode command message from eNB to UE shall contain the selected AS algorithms. This message shall be integrity protected with RRC integrity key based on the current K_{ASME}.

The AS security mode complete message from UE to eNB shall be integrity protected with the selected RRC algorithm indicated in the AS security mode command message and RRC integrity key based on the current K_{ASME}.

RRC and UP downlink ciphering (encryption) at the eNB shall start after sending the AS security mode command message. RRC and UP uplink deciphering (decryption) at the eNB shall start after receiving and successful verification of the AS security mode complete message.

RRC and UP uplink ciphering (encryption) at the UE shall start after sending the AS security mode complete message. RRC and UP downlink deciphering (decryption) at the UE shall start after receiving and successful verification of the AS security mode command message

If any control of the AS security mode command is not successful in the ME, the ME shall reply with an unprotected security mode failure message (see TS 36.331[21]).

AS security mode command always changes the AS keys.

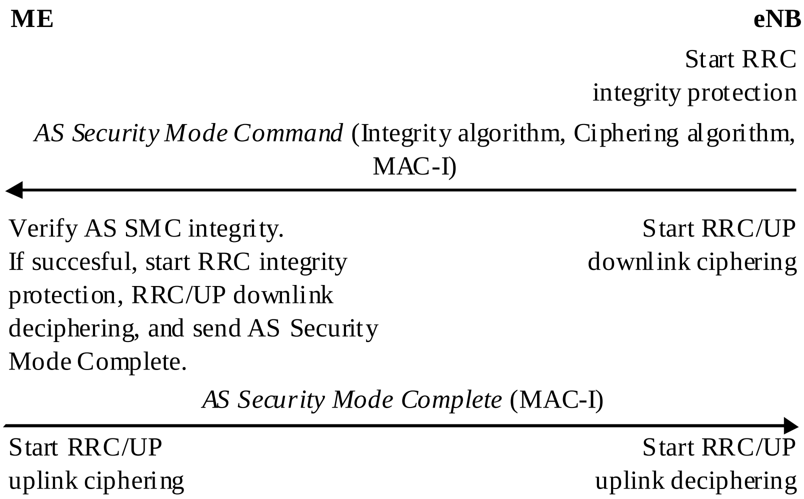

Figure 7.2.4.5-1: AS security setup

7.2.5 Key handling at state transitions to and away from EMM-DEREGISTERED

7.2.5.1 Transition to EMM-DEREGISTERED

There are different reasons for transition to the EMM-DEREGISTERED state. In all cases the UE and MME shall do the following

1. If they have a full non-current native security context and a current mapped security context, then they shall make the non-current native security context the current one.
2. They shall delete any mapped or partial security contexts they hold.

Handling of the remaining authentication data for each of these cases are given below. As these are NAS messages, they will be integrity protected when a security context exists in the UE and MME.

1. Attach reject: All authentication data shall be removed from the UE and MME
2. Detach:
   a. UE-initiated
      i. If the reason is switch off then all the remaining authentication data shall be removed from the UE and MME with the exception of:
         - the current native EPS NAS security context (as in clause 6.1.1), which should remain stored in the MME and UE, and
         - any unused authentication vectors, which may remain stored in the MME.
      ii. If the reason is not switch off then MME and UE shall keep all the remaining authentication data.
   b. MME-initiated
      i. Explicit: all the remaining authentication data shall be kept in the UE and MME if the detach type is re-attach.
      ii. Implicit: all the remaining authentication data shall be kept in the UE and MME.
   c. HSS-initiated: If the message is "subscription withdrawn" then all the remaining authentication data shall be removed from the UE and MME.

If the USIM supports EMM parameters storage then the ME shall update the EPS NAS security context parameters on the USIM, excluding the keys K_{NASint} and $K_{NASenc,}$ with the values of the full native EPS security context if it has one and if so mark the EPS NAS security context on the USIM as valid. Otherwise, the ME shall update the EPS NAS security context, excluding the keys K_{NASint} and K_{NASenc}, in its non-volatile memory with its values of the full native EPS security context if it has one and if so mark the EPS NAS security context in its non-volatile memory as valid.

3. TAU reject: There are various reasons for TAU reject. The action to be taken shall be as given in TS 24.301.

For the case that the MME or the UE enter EMM-DEREGISTERED state without using any of the above procedures, the handling of the remaining authentication data shall be as specified in TS 24.301 [9].

7.2.5.2 Transition away from EMM-DEREGISTERED

7.2.5.2.1 General

When the UE transits from EMM-DEREGISTERED to EMM-REGISTERED/ECM-CONNECTED, there are two cases to consider, either a complete native EPS NAS security context exists, or it does not.

7.2.5.2.2 With existing native EPS NAS security context

If there is native EPS NAS security context the ME first derives the keys K_{NASint} and K_{NASenc}, then the UE shall transmit a NAS Attach Request message. This message is integrity protected, and provided there is no NAS SMC procedure before the AS SMC the NAS COUNT of the Attach Request message shall be used to derive the K_{eNB} with the KDF as specified in Annex A. As a result of the NAS Attach Request, the eNB shall send an AS SMC to the UE to activate AS security. The K_{eNB} used, is derived in the current EPS NAS security context.

When the UE receives the AS SMC without having received a NAS Security Mode Command after the Attach/Service Request, it shall use the NAS COUNTof the Attach/Service Request message (i.e. the uplink NAS COUNT) that triggered the AS SMC to be sent as freshness parameter in the derivation of the K_{eNB}. From this K_{eNB} the RRC protection keys and the UP protection keys shall be derived as described in subclause 7.2.1.

The same procedure for refreshing K_{eNB} can be used regardless of the fact if the UE is connecting to the same MME to which it was connected previously or to a different MME. In case UE connects to a different MME and this MME supports different NAS algorithms, the NAS keys have to be re-derived in the MME with the new algorithm IDs as input using the KDF as specified in Annex A.

In addition, there is a need for the MME to send a NAS SMC to the UE to indicate the change of NAS algorithms and to take the re-derived NAS keys into use. The UE shall assure that the NAS keys used to verify the integrity of the NAS SMC are derived using the algorithm ID specified in the NAS SMC. The NAS SMC Command and NAS SMC Complete messages are protected with the new keys.

If there is a NAS Security Mode Command after the Attach/Service Request but before the AS SMC, the UE and MME use the NAS COUNT of the NAS Security Mode Complete (i.e. the uplink NAS COUNT) and the related K_{ASME} as the parameter in the derivation of the K_{eNB}. From this K_{eNB} the RRC protection keys and the UP protection keys are derived as described in subclause 7.2.1.

If the USIM supports EMM parameters storage, the ME shall mark the stored EPS NAS security context on the USIM as invalid at the end of the transition away from EMM-DEREGISTRED. Otherwise, the ME shall mark the stored EPS NAS security context on its non-volatile memory as invalid at the end of the transition.

7.2.5.2.3 With run of EPS AKA

If there is no native EPS NAS security context available an EPS AKA run is required. If there is native EPS NAS security context available the MME may decide to run an EPS AKA and a NAS SMC procedure (which activates the EPS NAS security context based on the K_{ASME} derived during the EPS AKA run) after the Attach Request but before the corresponding AS SMC), the NAS (uplink and downlink) COUNTs are reset to start values, and the start value of the uplink NAS COUNTshall be used as freshness parameter in the K_{eNB} derivation from the fresh K_{ASME} (after AKA) when UE receives AS SMC the K_{eNB} is derived from the current EPS NAS security context, i.e., the fresh K_{ASME} is used to derive the K_{eNB} The KDF as specified in Annex A shall be used to derive the K_{eNB}.

NOTE: Using the start value for the uplink NAS COUNTin this case cannot lead to the same combination of K_{ASME} and NAS COUNTbeing used twice. This is guaranteed by the fact that the first integrity protected NAS message the UE sends to the MME after AKA is the NAS SMC complete message.

The NAS SMC complete message shall include the start value of the NAS COUNTthat is used as freshness parameter in the K_{eNB} derivation and the K_{ASME} is fresh. After an AKA, a NAS SMC needs to be sent from the MME to the UE in order to take the new NAS keys into use. Both NAS SMC and NAS SMC Complete messages are protected with the new NAS keys.

If the USIM supports EMM parameters storage, the ME shall mark the stored EPS NAS security context on the USIM as invalid. Otherwise, the ME shall mark the storedEPS NAS security context on its non-volatile memory as invalid.

7.2.6 Key handling in ECM-IDLE to ECM-CONNECTED and ECM-CONNECTED to ECM-IDLE transitions when in EMM-REGISTERED state

7.2.6.1 General

As a general principle, on ECM-IDLE to ECM-CONNECTED transitions when in EMM-REGISTERED state, RRC protection keys and UP protection keys shall be generated as described in subclause 7.2.1 while K_{ASME} is assumed to be already available in the MME.

K_{ASME} may have been established in the MME as a result of an AKA run, or as a result of a security context transfer from another MME during handover or idle mode mobility. On ECM-CONNECTED to ECM-IDLE transitions, eNBs shall delete the keys they store such that state in the network for ECM-IDLE state UEs will only be maintained in the MME.

7.2.6.2 ECM-IDLE to ECM-CONNECTED transition

The procedure the UE uses to transit from ECM-IDLE to ECM-CONNECTED when in EMM-REGISTERED state is initiated by a NAS Service Request message from the UE to the MME. As the UE is in EMM-REGISTERED state, an EPS security context exists in the UE and the MME, and this EPS security context further contains uplink and downlink NAS COUNTs. The NAS Service Request message sent in EMM-REGISTERED shall be integrity protected and contain the next-in-sequence uplink NAS sequence number.

Upon receipt of the NAS Service Request message, if the MME does not requires a NAS SMC procedure before initiating the S1-AP procedure INITIAL CONTEXT SETUP, the MME shall derive key K_{eNB} as specified in subclause A.3 using the NAS COUNT [9] corresponding to the NAS Service Request and the K_{ASME} of the current EPS NAS security context. The MME shall further initialize the value of the Next hop Chaining Counter (NCC) to zero. The MME shall further derive a next hop parameter NH as specified in subclause A.4 using the newly derived K_{eNB} and the K_{ASME} as basis for the derivation. The MME shall further set the the value of the Next hop Chaining Counter (NCC) to one. This fresh {NH, NCC=1} pair shall be stored in the MME and shall be used for the next forward security key derivation. The MME shall communicate the K_{eNB} pair to the serving eNB in the S1-AP procedure INITIAL CONTEXT SETUP. The UE shall derive the K_{eNB} from the K_{ASME} of the current EPS NAS security context.

As a result of the NAS Service Request, radio bearers are established, and the eNB sends an AS SMC to the UE. When the UE receives the AS SMC without having received a NAS Security Mode Command, it shall use the NAS uplink COUNT of the NAS Service Request message that triggered the AS SMC as freshness parameter in the derivation of the K_{eNB}. The KDF as specified in Annex A shall be used for the K_{eNB} derivation using the K_{ASME} of the current EPS NAS security context. The UE shall further derive the NH parameter from the newly derived K_{eNB} and the K_{ASME} in the same way as the MME. From the K_{eNB} the RRC protection keys and the UP protection keys are derived by the UE and the eNB as described in subclause 6.2.

If the ECM-IDLE to ECM-CONNECTED procedure contains an EPS AKA run and a NAS SMC procedure (which are optional), the NAS uplink and downlink COUNT for the new K_{ASME} shall be set to the start values (i.e. zero). If the ECM-IDLE to ECM-CONNECTED procedure contains an NAS SMC (which is optional), the value of the uplink NAS COUNT from the NAS Security Mode Complete shall be used as freshness parameter in the K_{eNB} derivation from fresh K_{ASME} of the current EPS NAS security context when executing an AS SMC. The KDF as specified in Annex A shall be used for the K_{eNB} derivation also in this case.

On transitions to ECM-CONNECTED, the MME should be able to check whether a new authentication is required, e.g. because of prior inter-provider handover.

If the USIM supports EMM parameters storage, the ME shall mark the stored EPS NAS security context on the USIM as invalid. Otherwise, the ME shall mark the stored EPS NAS security context in its non-volatile memory as invalid.

7.2.6.3 ECM-CONNECTED to ECM-IDLE transition

On ECM-CONNECTED to ECM-IDLE transitions the eNB does no longer need to store state information about the corresponding UE.

In particular, on ECM-CONNECTED to ECM-IDLE transitions:

- The eNB and the UE shall delete the AS security context.
- MME and the UE shall keep the EPS NAS security context stored. MME shall delete NH and NCC.

  If the USIM supports EMM parameters storage, then the ME shall update the EPS NAS security context parameters on the USIM, excluding the keys K_{NASint} and K_{NASenc}, with its values of the full native EPS NAS security context if it has one and if so mark the EPS NAS security context on the USIM as valid. Otherwise, the ME shall update the EPS NAS security context, excluding the keys K_{NASint} and K_{NASenc}, in its non-volatile memory with the values of the full native EPS NAS security context if it has one and if so mark the EPS NAS security context in its non-volatile memory as valid.

7.2.7 Key handling in ECM-IDLE mode mobility

If the "active flag" is not set in the TAU request, the TAU procedure does not establish any RRC or UP level security. Because of this, there is no need to derive any K_{eNB} in this case. If the "active flag" is set in the TAU request message, radio bearers will be established as part of the TAU procedure. In this case a K_{eNB} derivation is necessary, and if there was no subsequnet NAS SMC, the uplink NAS COUNTof the TAU request message sent from the UE to the MME is used as freshness parameter in the K_{eNB} derivation using the KDF as specified in Annex A. The TAU request shall be integrity protected..

In the case an AKA is run successfully followed by a NAS SMC from the MME as part of the TAU procedure, the uplink and downlink NAS COUNTshall be set to the start values (i.e. zero).

In the case source and target MME use different NAS algorithms, the target MME re-derives the NAS keys from K_{ASME} with the new algorithm identities as input and provides the new algorithm identifiers within a NAS SMC. The UE shall assure that the NAS keys used to verify the integrity of the NAS SMC are derived using the algorithm identity specified in the NAS SMC.

If there is a NAS Security Mode Command after the TAU Request but before the AS SMC, the UE and MME use the NAS COUNT of the NAS Security Mode Complete (i.e. the uplink NAS COUNT) and the related K_{ASME} as the parameter in the derivation of the K_{eNB}. From this K_{eNB} the RRC protection keys and the UP protection keys are derived as described in subclause 7.2.1.

7.2.8 Key handling in handover

7.2.8.1 General

7.2.8.1.1 Access stratum

The general principle of key handling at handovers is depicted in Figure 7.2.8.1-1.

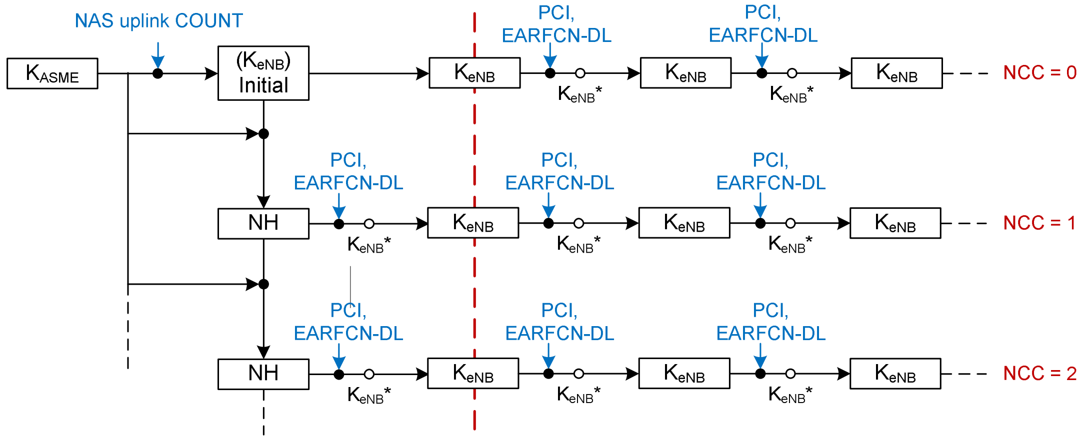

Figure 7.2.8.1-1 Model for the handover key chaining

The following is an outline of the key handling model to clarify the intended structure of the key derivations. The detailed specification is provided in subclauses 7.2.8.3 and 7.2.8.4.

Whenever an initial AS security context needs to be established between UE and eNB, MME and the UE shall derive a K_{eNB} and a Next Hop parameter (NH). The K_{eNB} and the NH are derived from the K_{ASME}. A NH Chaining Counter (NCC) is associated with each K_{eNB} and NH parameter. Every K_{eNB} is associated with the NCC corresponding to the NH value from which it was derived. At initial setup, the K_{eNB} is derived directly from K_{ASME}, and is then considered to be associated with a virtual NH parameter with NCC value equal to zero. At initial setup, the derived NH value is associated with the NCC value one.

Whether the MME sends the K_{eNB} key or the {NH, NCC} pair to the serving eNB is described in detail in subclauses 7.2.8.3 and 7.2.8.4. The MME shall not send the NH value to eNB at the initial connection setup.

NOTE 1: Since the MME does not send the NH value to eNB at the initial connection setup, the NH value associated with the NCC value one can not be used in the next X2 handover or the next intra-eNB handover, for the next X2 handover or the next intra-eNB handover the horizontal key derivation (see Figure 7.2.8.1-1) will apply.

The UE and the eNB use the K_{eNB} to secure the communication between each other. On handovers, the basis for the K_{eNB} that will be used between the UE and the target eNB, called K_{eNB}*, is derived from either the currently active K_{eNB} or from the NH parameter. If K_{eNB}* is derived from the currently active K_{eNB} this is referred to as a horizontal key derivation (see Figure 7.2.8.1-1) and if the K_{eNB}* is derived from the NH parameter the derivation is referred to as a vertical key derivation (see Figure 7.2.8.1-1). On handovers with vertical key derivation the NH is further bound to the target PCI and its frequency EARFCN-DL before it is taken into use as the K_{eNB} in the target eNB. On handovers with horizontal key derivation the currently active K_{eNB} is further bound to the target PCI and its frequency EARFCN-DL before it is taken into use as the K_{eNB} in the target eNB.

As NH parameters are only computable by the UE and the MME, it is arranged so that NH parameters are provided to eNBs from the MME in such a way that forward security can be achieved.

7.2.8.1.2 Non access stratum

A NAS aspect that needs to be considered is possible NAS algorithm change at MME change that could occur at a handover. At an eNB handover with MME relocation, there is the possibility that the source MME and the target MME do not support the same set of NAS algorithms or have different priorities regarding the use of NAS algorithms. In this case, the target MME re-derives the NAS keys from K_{ASME} using the NAS algorithm identities as input to the NAS key derivation functions (see Annex A) and sends NAS SMC. All inputs, in particular the K_{ASME}, will be the same in the re-derivation except for the NAS algorithm identity.

It is essential that the NAS COUNTs are not reset to the start values unless a new K_{ASME} is taken into use. This prevents that, in the case a UE moves back and forth between two MMEs the same NAS keys will be re-derived time and time again resulting in key stream re-use

The NAS COUNTs shall only be reset to the start value in the following cases:

- for an EPS NAS security context created by an AKA run successfully followed by a NAS SMC,

- or for an EPS NAS security context created through a context mapping during a handover from UTRAN/GERAN to E-UTRAN,

- or for an EPS NAS security context created through a context mapping during idle mode mobility from UTRAN/GERAN to E-UTRAN.

The NAS COUNT shall not be reset during idle mode mobility or handover for an already existing native EPS NAS security context.

In case the target MME decides to use NAS algorithms different from the ones used by the source MME, a NAS SMC including eKSI (new or current value depending on whether AKA was run or not) shall be sent from the MME to the UE.

The start value of NAS COUNT shall be zero (0).

This NAS Key and algorithm handling also applies to other MME changes e.g. TAU with MME changes.

NOTE: It is per operator's policy how to configure selection of handover types. Depending on an operator's security requirements, the operator can decide whether to have X2 or S1 handovers for a particular eNB according to the security characteristics of a particular eNB.

7.2.8.2 Void

7.2.8.3 Key derivations for context modification procedure

As outlined in subclause 7.2.8.1, whenever a fresh K_{eNB} is calculated from the K_{ASME} (as described in Annex A.3), the MME shall transfer the K_{eNB} to the serving eNB in a message modifying the security context in the eNB. The MME and the UE shall also compute the NH parameter from the K_{ASME} and the fresh K_{eNB} as described in Annex A.4 according to the rules in clause 7.2.9.2. An NCC value 1 is associated with the NH parameter derived from the fresh K_{eNB} and NCC value 0 with the K_{eNB}. The UE shall compute K_{eNB} and NH in the same way as the MME.

NOTE: Since MME does not send the NH value to eNB in S1 UE CONTEXT MODIFICATION REQUEST, the NH value associated with the NCC value one can not be used in the next X2 handover or the next intra-eNB handover. So for the next X2 handover or the next intra-eNB handover the horizontal key derivation (see Figure 7.2.8.1-1) will apply.

7.2.8.4 Key derivations during handovers

7.2.8.4.1 Intra-eNB Handover

When the eNB decides to perform an intra-eNB handover it shall derive K_{eNB}* as in Annex A.5 using target PCI, its frequency EARFCN-DL, and either NH or the current K_{eNB} depending on the following criteria: the eNB shall use the NH for deriving K_{eNB}* if an unused {NH, NCC} pair is available in the eNB (this is referred to as a vertical key derivation), otherwise if no unused {NH, NCC} pair is available in the eNB, the eNB shall derive K_{eNB}* from the current K_{eNB} (this is referred to as a horizontal key derivation).

The eNB shall use the K_{eNB}* as the K_{eNB} after handover. The eNB shall send the NCC used for K_{eNB}* derivation to UE in HO Command message.

7.2.8.4.2 X2-handover

As in intra-eNB handovers, for X2 handovers the source eNB shall perform a vertical key derivation in case it has an unused {NH,NCC} pair. The source eNB shall first compute K_{eNB}* from target PCI, its frequency EARFCN-DL, and either from currently active K_{eNB} in case of horizontal key derivation or from the NH in case of vertical key derivation as described in Annex A.5. The target eNB shall associate the NCC value received from source eNB with the K_{eNB}.

Next the source eNB shall forward the {K_{eNB}*, NCC} pair to the target eNB. The target eNB shall include the received NCC into the prepared HO Command message, which is sent back to the source eNB in a transparent container and forwarded to the UE by source eNB.

The target eNB shall use the received K_{eNB}* directly as K_{eNB} to be used with the UE.

When the target eNB has completed the handover signaling with the UE, it shall send a S1 PATH SWITCH REQUEST to the MME. Upon reception of the S1 PATH SWITCH REQUEST, the MME shall increase its locally kept NCC value by one and compute a new fresh NH by using the K_{ASME} and its locally kept NH value as input to the function defined in Annex A.4. The MME shall then send the newly computed {NH, NCC} pair to the target eNB in the S1 PATH SWITCH REQUEST ACKNOWLEDGE message. The target eNB shall store the received {NH, NCC} pair for further handovers and remove other existing unused stored {NH, NCC} pairs if any.

NOTE: Because the path switch message is transmitted after the radio link handover, it can only be used to provide keying material for the next handover procedure and target eNB. Thus, for X2-handovers key separation happens only after two hops because the source eNB knows the target eNB keys. The target eNB can immediately initiate an intra-cell handover to take the new NH into use once the new NH has arrived in the S1 PATH SWITCH REQUEST ACKNOWLEDGE.

7.2.8.4.3 S1-Handover

When an S1-handover is performed, the source eNB shall not send any keys to the MME in the S1 HANDOVER REQUIRED message.

Upon reception of the HANDOVER REQUIRED message the source MME shall compute a fresh {NH, NCC} pair from its stored data using the function defined in Annex A.4. The source MME shall store that fresh pair and send it to the target MME in the S10 FORWARD RELOCATION REQUEST message. The S10 FORWARD RELOCATION REQUEST message shall in addition contain the K_{ASME} that is currently used to compute {NH, NCC} pairs and its corresponding eKSI.

The target MME shall store locally the {NH, NCC} pair received from the source MME.

The target MME shall then send the received {NH, NCC} pair to the target eNB within the S1 HANDOVER REQUEST.

Upon receipt of the S1 HANDOVER REQUEST from the target MME, the target eNB shall compute the K_{eNB} to be used with the UE by performing the key derivation defined in Annex A.5 with the fresh{NH, NCC} pair in the S1 HANDOVER REQUEST and the target PCI and its frequency EARFCN-DL. The target eNB shall include the NCC value from the received {NH, NCC} pair into the HO Command to the UE and remove any existing unused stored {NH, NCC} pairs.

NOTE: The source MME may be the same as the target MME in the description in this subclause. If so the single MME performs the roles of both the source and target MME, i.e. the MME calculates and stores the fresh {NH, NCC} pair and sends this to the target eNB.

7.2.8.4.4 UE handling

The UE behaviour is the same regardless if the handover is S1, X2 or intra-eNB.

If the NCC value the UE receieved in the HO Command message from target eNB via source eNB is equal to the NCC value associated with the currently active K_{eNB}, the UE shall dervie the K_{eNB}* from the currently active K_{eNB} and the target PCI and its frequency EARFCN-DL using the function defined in Annex A.5.

If the UE received an NCC value that was different from the NCC associated with the currently active K_{eNB}, the UE shall first synchronize the locally kept NH parameter by computing the function defined in Annex A.4 iteratively (and increasing the NCC value until it matches the NCC value received from the source eNB via the HO command message. When the NCC values match, the UE shall compute the K_{eNB}* from the synchronized NH parameter and the target PCI and its frequency EARFCN-DL using the function defined in Annex A.5.

The UE shall use the K_{eNB}* as the K_{eNB} when communicating with the target eNB.

7.2.9 Key-change-on-the fly

7.2.9.1 General

Key-change-on-the fly consists of re-keying or key-refresh.

Key refresh shall be possible for K_{eNB} , $K_{RRC-enc}$, $K_{RRC-int}$, and K_{UP-enc} and shall be initiated by the eNB when a PDCP COUNTs is about to be re-used with the same Radio Bearer identity and with the same K_{eNB}. The procedure is described in clause 7.2.9.3.

Re-keying shall be possible for the K_{eNB} , $K_{RRC-enc}$, $K_{RRC-int}$, and K_{UP-enc} . This re-keying shall be initiated by the MME when an EPS AS security context different from the currently active one shall be activated. The procedures for doing this are described in clause 7.2.9.2.

Re-keying shall be possible for $K_{NAS-enc}$ and $K_{NAS-int}$. Re-keying of $K_{NAS-enc}$ and $K_{NAS-int}$ shall be initiated by the MME when a NAS EPS security context different from the currently active one shall be activated. The procedures for doing this are described in clause 7.2.9.4.

Re-keying of the entire EPS key hierarchy including K_{ASME} shall be achieved by first re-keying K_{ASME}, then$K_{NAS-enc}$ and $K_{NAS-int}$, followed by re-keying of the K_{eNB} and derived keys. For NAS key change-on-on-the fly, activation of NAS keys is accomplished by a NAS SMC procedure.

AS Key change on-the-fly is accomplished using a procedure based on intra-cell handover. The following AS key changes on-the-fly shall be possible: local K_{eNB} refresh (performed when PDCP COUNTs are about to wrap around), K_{eNB} re-keying performed after an AKA run, activation of a native context after handover from UTRAN or GERAN.

7.2.9.2 K_{eNB} re-keying

The re-keying procedure is initiated by the MME after a successful AKA run with the UE to activate a partial native EPS security context, or to re-activate a non-current full native EPS security context after handover from GERAN or UTRAN according to subclauses 9.2.2.1 and 10.3.2.

In case the procedure is initiated by the MME after a successful AKA run with the UE, the MME derives the new K_{eNB} using the same key derivation function as is used for ECM-IDLE to ECM-CONNECTED state transitions (see Annex A) using the new K_{ASME} and the NAS COUNT used in the NAS Security Mode Complete message. The K_{eNB} is sent to the eNB after a successfully completed NAS SMC in a S1 AP UE CONTEXT MODIFICATION REQUEST message triggering the eNB to perform the re-keying. The eNB runs the key-change-on-the-fly procedure with the UE. During this procedure the eNB shall indicate to the UE that a key change on-the-fly is taking place. The procedure used is based on an intra-cell handover, and hence the same K_{eNB} derivation steps shall be taken as in a normal handover procedure.

When the UE receives an indication that the procedure is a key change on-the-fly procedure, the UE shall use the K_{ASME} from the current EPS NAS security context as the basis for K_{eNB} derivations.

NOTE 1: To perform a key change on-the-fly of the entire key hierarchy, the MME has to change the EPS NAS security context before changing the AS security context.

If the UE has determined that the eKSI has changed, the UE shall derive a temporary K_{eNB} by applying the same key derivation function as is used in ECM-IDLE to ECM-CONNECTED state transitions (see Annex A) using the NAS COUNT in the NAS Security Mode Complete message and the new K_{ASME} as input. From this temporary K_{eNB} the UE shall derive the K_{eNB}* as normal (see Annex A). The eNB shall take the K_{eNB} it received from the MME, which is equal to the temporary $K_{eNB,}$ as basis for its K_{eNB}* derivations. From this step onwards, the key derivations continue as in a normal handover.

If the AS level re-keying fails, then the MME shall complete another NAS security mode procedure before initiating a new AS level re-keying. This ensures that a fresh K_{eNB} is used.

In case the re-keying procedure is initiated by the MME to re-activate a non-current full native EPS security context after handover from GERAN or UTRAN the same procedure as above applies.

The NH parameter shall be handled according to the following rules:

- UE and MME shall use NH derived from old K_{ASME} before the context modification is complete, i.e. for the UE when it sends the RRC Connection Reconfiguration Complete, and for the MME when it receives the UE CONTEXT MODIFICATION RESPONSE. In particular, the MME shall send an NH derived from old K_{ASME} in the S1AP HANDOVER RESOURCE ALLOCATION, S10 FORWARD RELOCATION, and S1AP PATH SWITCH REQUEST ACKNOWLEDGE messages before the context modification is complete.

- The eNB shall delete any old NH upon completion of the context modification.

- The UE and MME shall delete any old NH upon completion of the context modification. After the completion of the context modification, the UE and the MME shall derive any new NH parameters from the new K_{eNB} and the new K_{ASME} according to Annex A.4.

In case the eNB has scheduled the UE for a handover when the re-keying message is received from the MME, the eNB and the UE shall perform the same key derivation steps as if it was an intra-cell handover with the sole purpose of a K_{eNB} re-keying.

NOTE 2: It is left to stage 3 specifications whether re-keying and inter-cell handover can be combined, and, if not, in which order these two procedures are executed.

7.2.9.3 KeNB refresh

This procedure is based on an intra-cell handover. The K_{eNB} chaining that is performed during a handover ensures that the K_{eNB} is re-freshed w.r.t. the RRC and UP COUNT after the procedure.

7.2.9.4 NAS key re-keying

After an AKA has taken place, new NAS keys from a new K_{ASME} shall be derived, according to Annex A.7.

To re-activate a non-current full native EPS security context after handover from GERAN or UTRAN, cf. clause 9.2.2 B step 7, the UE and the MME take the NAS keys into use by running a NAS SMC procedure according to clause 7.2.4.5.

MME shall activate fresh NAS keys from an EPS AKA run or activate native security context with sufficiently low NAS COUNT values before the NAS uplink or downlink COUNT wraps around with the current security context.

7.2.10 Rules on Concurrent Running of Security Procedures

Concurrent runs of security procedures may, in certain situations, lead to mismatches between security contexts in the network and the UE. In order to avoid such mismatches, the following rules shall be adhered to:

1. MME shall not initiate any of the S1 procedures Initial Context Setup or UE Context Modification including a new K_{eNB} towards a UE if a NAS Security Mode Command procedure is ongoing with the UE.

2. The MME shall not initiate a NAS Security Mode Command towards a UE if one of the S1 procedures Initial Context Setup or UE Context Modification including a new K_{eNB} is ongoing with the UE.

3. When the MME has initiated a NAS SMC procedure in order to take a new K_{ASME} into use, the MME shall continue to include AS security context parameters based on the old K_{ASME} in the HANDOVER REQUEST or PATH SWITCH REQUEST ACKNOWLEDGE message, until the MME takes a K_{eNB} derived from the new K_{ASME} into use by means of an S1 Initial Context Setup procedure or a UE Context Modification procedure.

4. When the UE has received a NAS SMC message in order to take a new K_{ASME} into use, the UE shall continue to use AS security context parameters based on the old K_{ASME} in handover until the network indicates in an AS security mode command procedure or an RRCConnectionReconfiguration procedure to take a K_{eNB} derived from the new K_{ASME} into use.

5. The source eNB shall reject an S1 UE Context Modification Request when the eNB has initiated, but not yet completed, an inter-eNB handover. When a RRCConnectionReconfiguration procedure is ongoing the source eNB shall wait for the completion of this procedure before initiating any further handover procedure.

6. When the MME has initiated a NAS SMC procedure in order to take a new K_{ASME} into use and receives a request for an inter-MME handover from the serving eNB, the MME shall wait for the completion of the NAS SMC procedure before sending an S10 FORWARD RELOCATION message.

7. When the MME has initiated a UE Context Modification procedure in order to take a new K_{eNB} into use and receives a request for an inter-MME handover from the serving eNB, the MME shall wait for the (successful or unsuccessful) completion of the UE Context Modification procedure before sending an S10 FORWARD RELOCATION message.

8. When the MME has successfully performed a NAS SMC procedure taking a new K_{ASME} into use, but has not yet successfully performed a UE Context Modification procedure, which takes a K_{eNB} derived from the new

K_{ASME} into use, , the MME shall include both the old K_{ASME} with the corresponding eKSI, NH, and NCC, and a full EPS NAS security context based on the new K_{ASME} in the S10 FORWARD RELOCATION message.

9. When an MME receives a S10 FORWARD RELOCATION message including both the old K_{ASME} with the corresponding eKSI, NH, and NCC, and a full EPS NAS security context based on the new K_{ASME} the MME shall use the new K_{ASME} in NAS procedures, but shall continue to include AS security context parameters based on the old K_{ASME} in the HANDOVER REQUEST or PATH SWITCH REQUEST ACKNOWLEDGE message until the completion of a UE Context Modification procedure, which takes a K_{eNB} derived from the new K_{ASME} into use.

7.3 UP security mechanisms

7.3.1 UP confidentiality mechanisms

The user plane data is ciphered by the PDCP protocol between the UE and the eNB as specified in TS 36.323 [12]..

The use and mode of operation of the 128-EEA algorithms are specified in Annex B.

The input parameters to the 128-bit EEA algorithms as described in Annex B are an 128-bit cipher key K_{UPenc} as KEY, a 5-bit bearer identity BEARER which value is assigned as specified by TS 36.323 [12], the 1-bit direction of transmission DIRECTION, the length of the keystream required LENGTH and a bearer specific, time and direction dependent 32-bit input COUNT which corresponds to the 32-bit PDCP COUNT.

7.4 RRC security mechanisms

7.4.1 RRC integrity mechanisms

RRC integrity protection shall be provided by the PDCP layer between UE and eNB and no layers below PDCP shall be integrity protected.

The use and mode of operation of the 128-EIA algorithms are specified in Annex B.

The input parameters to the 128-bit EIA algorithms as described in Annex B are an 128-bit integrity key K_{RRCint} as KEY,, a 5-bit bearer identity BEARER which value is assigned as specified by TS 36.323 [12], the 1-bit direction of transmission DIRECTION and a bearer specific, time and direction dependent 32-bit input COUNT which corresponds to the 32-bit PDCP COUNT.

The supervision of failed RRC integrity checks shall be performed both in the ME and the eNB. In case of failed integrity check (i.e. faulty or missing MAC-I) is detected after the start of integrity protection, the concerned message shall be discarded. This can happen on the eNB side or on the ME side.

NOTE: This text does not imply that the concerned message is silently discarded. In fact, TS 36.331 [21] specifies that the UE shall trigger a recovery procedure upon detection of a failed RRC integrity check. When the cause for integrity protection failure is not a context mismatch, such as a key or HFN mismatch, the run of a recovery procedure unnecessarily adds load to the system. However, in the absence of a means for the UE to reliably detect the cause of an integrity protection failure and the fact that the only identified consequence of an active attack is limited to non-persistent DoS effects, priority was given to a procedure allowing recovery from the deadlock caused by a context mismatch.

7.4.2 RRC confidentiality mechanisms

RRC confidentiality protection is provided by the PDCP layer between UE and eNB.

The use and mode of operation of the 128-EEA algorithms are specified in Annex B.

The input parameters to the 128-bit EEA algorithms as described in Annex B are an 128-bit cipher Key K_{RRCenc} as KEY, a 5-bit bearer identity BEARER which corresponds to the radio bearer identity, the 1-bit direction of transmission DIRECTION, the length of the keystream required LENGTH and a bearer specific, time and direction dependent 32-bit input COUNT which corresponds to the 32-bit PDCP COUNT.

7.4.3 K_{eNB}* and Token Preparation for the RRCConnectionRe-establishment Procedure

The K_{eNB}* and token calculation at handover preparation are cell specific instead of eNB specific. At potential RRC Connection re-establishment (e.g, in handover failure case), the UE may select a cell different from the target cell to initiate the re-establishment procedure. To ensure that the UE RRCConnectionRe-establishment attempt is successful when the UE selects another cell under the control of the target eNB at handover preparation, the serving eNB could prepare multiple K_{eNB}*s and tokens for mulitple cells which are under the control of the target eNB.

The preparation of these cells includes sending security context containing K_{eNB}*s and tokens for each cell to be prepared, as well as the corresponding NCC, the UE EPS security capabilities, and the security algorithms used in the source cell for computing the token, to the target eNB. The source eNB shall derive the K_{eNB}*s as described in Annex A.5 based on the corresponding target cell"s physical cell ID and frequency EARFCN-DL.

In order to calculate the token, the source eNB shall use the negotiated EIA-algorithm from the AS Security context from the source eNB with the following inputs: source C-RNTI, source PCI and target Cell-ID as defined by V*arShortMAC-Input* in TS 36.331 [21], where source PCI and source C-RNTI are associated with the cell the UE last had an active RRC connection with and target cell ID is the identity of the target cell where the RRCConnectionReestablishmentRequest is sent to.

- KEY shall be set to K_{RRCint} of the source cell;

- all BEARER bits shall be set to 1;

- DIRECTION bit shall be set to 1;

- all COUNT bits shall be set to 1.

The token shall be the 16 least significant bits of the output of the used integrity algorithm.

For X2 handover, the target eNB shall use these received multiple K_{eNB}*s. But for S1 handover, the target eNB discards the multiple K_{eNB}*s received from the source eNB, and derives the K_{eNB}*s as described in Annex A.5 based on the received fresh {NH, NCC} pair from MME for forward security purpose.

When an RRCConnectionReestablishmentRequest is initiated by the UE, the RRCConnectionReestablishmentRequest shall contain the token corresponding to the cell the UE tries to reconnect to. This message is transmitted over SRB0 and hence not integrity protected.

The target eNB receiving the RRCConnectionReestablishmentRequest shall respond with an RRCConnectionReestablishment message containing the NCC received during the preparation phase if the token is valid, otherwise the target eNB shall reply with an RRCConnectionReestablishmentReject message. The RRCConnectionReestablishment and RRCConnectionReestablishmentReject messages are also sent on SRB0 and hence not integrity protected. Next the target eNB and UE shall do the following: The UE shall firstly synchronize the locally kept NH parameter as defined in Annex A.4 if the received NCC value is different from the current NCC value in the UE itself. Then the UE shall derive K_{eNB}* as described in Annex A.5 based on the selected cell"s physical cell ID and its frequency EARFCN-DL. The UE shall use this K_{eNB}* as K_{eNB}. The eNB uses the K_{eNB}* corresponding to the selected cell as K_{eNB}. Then, UE and eNB shall derive and activate keys for integrity protection and verification from this K_{eNB}.

The UE shall respond with an RRCReestablishmentComplete on SRB1,integrity protected and ciphered using these new keys. The RRCConnectionReconfiguration procedure used to re-establish the remaining radio bearers shall only include integrity protected and ciphered messages.

7.5 Signalling procedure for periodic local authentication

The following procedure is used optionally by the eNB to periodically perform a local authentication. At the same time, the amount of data sent during the AS connection is periodically checked by the eNB and the UE for both up and down streams. If UE receives the Counter Check request, it shall respond with Counter Check Response message.

The eNB is monitoring the PDCP COUNT values associated to each radio bearer. The procedure is triggered whenever any of these values reaches a critical checking value. The granularity of these checking values and the values themselves are defined by the visited network. All messages in the procedure are integrity protected.

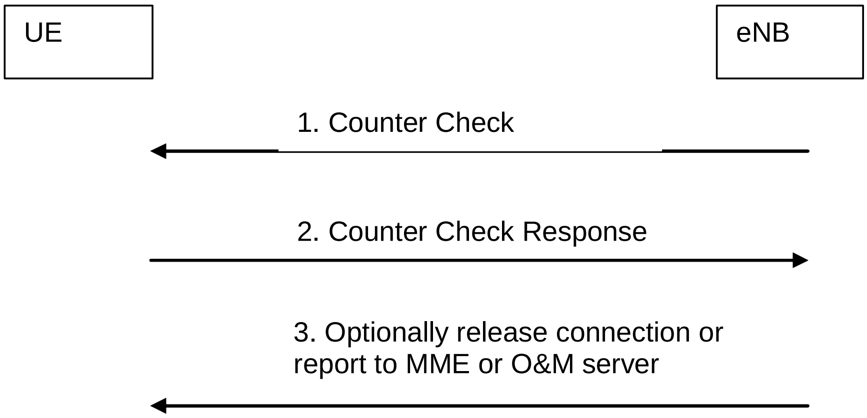

Figure 7.5-1: eNB periodic local authentication procedure

1. When a checking value is reached (e.g. the value in some fixed bit position in the hyperframe number is changed), a Counter Check message is sent by the eNB. The Counter Check message contains the most significant parts of the PDCP COUNT values (which reflect amount of data sent and received) from each active radio bearer.

2. The UE compares the PDCP COUNT values received in the Counter Check message with the values of its radio bearers. Different UE PDCP COUNT values are included within the Counter Check Response message.

3. If the eNB receives a counter check response message that does not contain any PDCP COUNT values, the procedure ends. If the eNB receives a counter check response that contains one or several PDCP COUNT values, the eNB may release the connection or report the difference of the PDCP COUNT values for the serving MME or O&M server for further traffic analysis for e.g. detecting the attacker.

8 Security mechanisms for non-access stratum signalling

8.1 NAS integrity mechanisms

Integrity protection for NAS signalling messages shall be provided as part of the NAS protocol.

8.1.1 NAS input parameters and mechanism

Input parameters to the NAS 128-bit integrity algorithms as described in Annex B are an 128-bit integrity key K_{NASint} as KEY, an 5-bit bearer identity BEARER which shall equal the constant value 0x00, the direction of transmission DIRECTION, and a bearer specific, time and direction dependent 32-bit input COUNT which is constructed as follows:

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

Where

- the leftmost 8 bits are padding bits including all zeros.

- NAS OVERFLOW is a 16-bit value which is incremented each time the NAS SQN is incremented from the maximum value.

- NAS SQN is the 8-bit sequence number carried within each NAS message.

NOTE: The BEARER identity is not necessary since there is only one NAS signalling connection per pair of MME and UE, but is included as a constant value so that the input parameters for AS and NAS will be the same, which simplifies specification and implementation work.

The use and mode of operation of the 128-bit integrity algorithms are specified in Annex B.

The supervision of failed NAS integrity checks shall be performed both in the ME and the MME. In case of failed integrity check (i.e. faulty or missing NAS-MAC) is detected after the start of NAS integrity protection, the concerned message shall be discarded except for some NAS messages specified in TS 24.301 [9]. For those exceptions the MME shall take the actions specified in TS 24.301 [9] when receiving a NAS message with faulty or missing NAS-MAC. Discarding NAS messages can happen on the MME side or on the ME side.

8.1.2 NAS integrity activation

NAS integrity shall be activated with the help of the NAS SMC procedure immediately after successful authentication. NAS integrity stays activated until the EPS security context is deleted. The EPS security context may only be deleted if UE is in EMM-DEREGISTERED. While the EPS security context exists, all NAS messages shall be integrity protected. In particular the NAS service request shall always be integrity protected and the NAS attach request message shall be integrity protected if the EPS security context is not deleted while UE is in EMM-DEREGISTERED. The length of the NAS-MAC is 32 bit. The full NAS-MAC shall be appended to all integrity protected messages except for the NAS service request. Only the 16 least significant bits of the 32 bit NAS-MAC shall be appended to the NAS service request message.

The use and mode of operation of the 128-EIA algorithms are specified in Annex B.

8.2 NAS confidentiality mechanisms

The input parameters for the NAS 128-bit ciphering algorithms shall be the same as the ones used for NAS integrity protection as described in clause 8.1, with the exception that a different key, K_{NASenc} , is used as KEY, and there is an additional input parameter, namely the length of the key stream to be generated by the encryption algorithms.

The use and mode of operation of the 128-bit ciphering algorithms are specified in Annex B.

9 Security interworking between E-UTRAN and UTRAN

9.1 Idle mode procedures

9.1.1 Idle mode procedures in UTRAN

This subclause covers both the cases of idle mode mobility from E-UTRAN to UTRAN and of Idle Mode Signaling Reduction (ISR), as defined in TS 23.401 [2].

NOTE 1: TS 23.401 states conditions under which a valid P-TMSI or a P-TMSI that is mapped from a valid GUTI ('mapped GUTI') is inserted in the Information Element 'old P-TMSI' in the Routing Area Update Request. It depends on the old P-TMSI which security context can be taken into use after completion of the Routing Area Update procedure.

Use of an existingUMTS securitycontext

If the UE sends the RAU Request with the "old P-TMSI" Information Element including a valid P-TMSI it shall also include the KSI relating to this P-TMSI. This KSI is associated with the UMTS security context stored on the UE, and it indicates this fact to the SGSN. In this case the UE shall include P-TMSI signature into the RAU Request if a P-TMSI signature was assigned by the old SGSN. If the network does not have a valid security context for this KSI it shall run AKA. In case of an SGSN change keys from the old SGSN shall overwrite keys in the new SGSN if any.

NOTE 2: if the UE has a valid UMTS security context then this context is stored on the USIM according to TS 33.102 [4].

Mapping of EPS security context to UMTS security context

If the UE sends the RAU Request with the "old P-TMSI" Information Element including mapped GUTI it shall also include the KSI equal to the value of the eKSI associated with the current EPS security context (cf. clause 3). The UE shall include a truncated NAS-token, as defined in this clause further below, into the P-TMSI signature IE. The MME shall transfer UE's UTRAN and GERAN security capabilities and CK' || IK' with KSI equal to the value of the eKSI associated with the current EPS security context to SGSN with Context Response/SGSN Context Response message. The MME and UE shall derive CK' and IK' from the K_{ASME} and the NAS downlink COUNT value corresponding to the truncated NAS-token received by the MME from SGSN as specified in Annex A. Keys CK' and IK' and KSI sent from the MME shall replace the keys and KSI in the target SGSN if any. Keys CK' and IK' and the KSI shall replace the currently stored values on the USIM. START shall be reset to 0 on USIM.

NOTE 3: The new derived security context (including CK", IK" and START value) replacing the old stored values in the USIM is for allowing to reuse the derived security context without invoking the authentication procedure in the subsequent connection set-ups , and also for avoiding that one KSI indicates to two different key sets and consequently leads to security context desynchronization.

NOTE 4: An operator concerned about the security of keys received from another operator may want to enforce a policy in SGSN to run a UMTS AKA as soon as possible after the run of an idle mode mobility procedure. An example of ensuring this is the deletion of the mapped UMTS security context in the SGSN after the completion of the idle mode mobility procedure.

SGSN shall include the allowed security algorithm and transfer them to RNC. An SMC shall be sent to the UE containing the selected algorithms.

The x bits available in the P-TMSI signature field (at minimum 16 bits) shall be filled with the truncated NAS-token, which is defined as the x least significant bits of the NAS-token.

The NAS-token is derived as specified in Annex A.9.

SGSN shall forward the P-TMSI signature including the truncated NAS token to the old MME, which compares the received bits of the truncated NAS-token with the corresponding bits of a NAS-token generated in the MME, for the UE identified within the context request. If they match, the context request message is authenticated and authorized and MME shall provide the needed information for the SGSN. Old MME shall respond with an appropriate error cause if it does not match the value stored in the old MME. This should initiate the security functions in the new SGSN.

To avoid possible race condition problems, the MME shall compare the received truncated NAS-token with the x least significant bits of NAS-tokens generated from the current NAS downlink COUNT value down to current NAS COUNT-L downlink values, i.e. the interval [current NAS downlink COUNT - L, current NAS downlink COUNT]. A suitable value for the parameter L can be configured by the network operator. MME shall not accept the same NAS-token for the same UE twice except in retransmission cases happening for the same mobility event.

9.1.2 Idle mode procedures in E-UTRAN

This subclause covers both the cases of idle mode mobility from UTRAN to E-UTRAN and of Idle Mode Signaling Reduction, as defined in TS 23.401 [2].

The TAU Request and ATTACH Request message shall include the UE security capabilities. The MME shall store these UE security capabilities for future use. The MME shall not make use of any UE security capabilities received from the SGSN.

NOTE 1: TS 23.401 states conditions under which a valid GUTI or a GUTI that is mapped from a valid P-TMSI is inserted in the Information Element 'old GUTI' in the Tracking Area Update Request.The value in the 'old' GUTI IE informs the MME, which SGSN/MME to fetch the UE context from.

Case 1: P-TMSI not included in 'old GUTI' IE in TAU Request

The UE shall use the current EPS security context to protect the TAU Request and include the corresponding GUTI and eKSI value. The TAU Request shall be integrity-protected, but not confidentiality-protected. UE shall use the current EPS security context algorithms to protect the TAU Request message.

Case 2: Mapped P-TMSI included in 'old GUTI' IE in TAU Request

If the UE sends a TAU Request to the MME, and was previously connected to UTRAN where the SGSN assigned a P-TMSI signature to the UE, the UE shall include that P-TMSI signature in the TAU Request.

If the MME received a P-TMSI signature from the UE, the MME shall include that P-TMSI signature in the Context Request message sent to the SGSN. The SGSN shall transfer CK || IK to MME in the Context Response/SGSN Context Response message. MME shall derive K'_{ASME} from CK || IK as described in Annex A. In case the MM context in the Context Response/SGSN Context Response indicates GSM security mode, the MME shall abort the procedure.

MME shall select security algorithms, based on the UE EPS security capabilities received in the TAU request, and indicate them to the UE by e.g. with NAS SMC.

If the UE has a current security context, the UE shall use this security context to protect the TAU Request. If not the UE shall send the TAU Request unprotected.

In both cases 2.1 or 2.2, the UE shall include in the TAU Request:

- the KSI with corresponding P-TMSI and old RAI to point to the right source SGSN and key set. there. This allows the UE and MME to generate the mapped security context, as described in Case 2.2 below, if current EPS security context is not available in the UE and network. The KSI shall correspond to the set of keys most recently generated (either by a successful AKA run or mapping from an EPS security context).
- a 32bit $NONCE_{UE}$ (see clause A.11 for requirements on the randomness of $NONCE_{UE}$).

NOTE 2: The current EPS security context may be of type "mapped", and hence the value of the eKSI be of type "KSI_{SGSN}". This value of KSI_{SGSN} may be different from the KSI pointing to the set of keys most recently generated in UTRAN as an AKA run may have happened in UTRAN after the current mapped EPS security context indicated by the eKSI with the value KSI_{SGSN} was generated

Case 2.1: Current EPS security context in the UE

The UE shall include the corresponding GUTI and eKSI value in the TAU Request. The TAU Request shall be integrity-protected, but not confidentiality-protected. The UE shall use the current security context algorithms to protect the TAU Request message.

NOTE 3: Case 2.1 relates to the following scenario: a UE established a current EPS security context during a previous visit to EPS, then moves to UTRAN/GERAN from E-UTRAN and storing the current EPS security context. When the UE moves back to E-UTRAN there is a current EPS security context.

In case MME has the current security context it shall verify the TAU Request message. If it is successful, the MME shall reply with a TAU Accept message protected with the security context. In case the TAU Request had the active flag set or there is pending downlink UP data, K_{eNB} is calculated as described in clause 7.2.7..

If the MME changes the algorithms, they shall be indicated to the UE in an integrity protected NAS SMC, which shall also include the UE security capabilities and KSI_{ASME}. This message shall not be ciphered. UE shall reply with integrity protected NAS SMC COMPLETE protected based on the new selected algorithms and K_{ASME}.

If the USIM supports EMM parameters storage then the new native EPS NAS security context shall be stored on the USIM.

If the MME does not have the security context indicated by the UE in the TAU request, then Case 2.2 below applies.

Case 2.2: Creating a Mapped EPS security context

If a current EPS security context is not available in the UE, the UE shall send the TAU request unprotected.

If the MME does not have the context indicated by the UE in the TAU request, or the TAU request was received unprotected, the MME shall create a new mapped security context. In this case, the MME shall generate a 32bit $NONCE_{MME}$ (see clause A.10 for requirements on the randomness of $NONCE_{MME}$). and use the received $NONCE_{UE}$ with the $NONCE_{MME}$ to generate a fresh mapped K'_{ASME} from CK and IK, where CK, IK were identified by the KSI and P-TMSI in the TAU Request. See Annex A.11 for more information on how to derive the fresh K'_{ASME}. If the TAU Request had the active flag set or there is pending downlink UP data, the uplink NAS Count which is set to zero shall be used to derive the K_{eNB} in MME and UE as specified in Annex A. MME shall deliver the K_{eNB} to the target eNB on the S1 interface. The uplink and downlink NAS COUNT for mapped security context shall be set to start value (i.e., 0) when new mapped security context is created in UE and MME.

The selected algorithms and the KSI_{SGSN} shall be indicated to the UE in an integrity protected NAS SMC COMMAND message protected with NAS keys based on K'_{ASME}, which shall also include the UE security capabilities, $NONCE_{UE}$, and $NONCE_{MME}$. The UE shall generate a mapped K'_{ASME} from CK and IK in the same way as the MME. The UE shall compare the received $NONCE_{UE}$ with the $NONCE_{UE}$ sent, and if the $NONCE_{UE}$ was modified in the TAU request, then the UE shall return NAS security mode reject message. If the received $NONCE_{UE}$ is the same as the $NONCE_{UE}$ sent and the integrity check succeeds, then UE shall reply with integrity protected and ciphered NAS SMC COMPLETE message based on the selected algorithms and NAS keys based on K'_{ASME} so that the MME can be sure that they were not modified in the TAU Request message by an outsider.

TAU Accept shall be protected using the NAS keys based on the fresh K'_{ASME}.

9.2 Handover

9.2.1 From E-UTRAN to UTRAN

NAS and AS security shall always be activated before handover from E-UTRAN to UTRAN can take place. Consequently the source system in the handover shall always send a key set to the target system during handover. The security policy of the target PLMN determines the selected algorithms to be used within the UTRAN HO command. UE and MME shall derive a confidentiality key CK', and an integrity key IK' from the K_{ASME} and the NAS downlink COUNT value of the current security context with the help of a one-way key derivation function KDF as specified in Annex A.

Whether ciphering is considered active in the target UTRAN after handover from E-UTRAN shall be determined according to the principles for handover to UTRAN in TS 25.331 [24].

UE and MME shall assign the value of eKSI to KSI. MME shall transfer CK' || IK' with KSI to SGSN. The target SGSN shall replace all stored parameters CK, IK, KSI, if any, with CK' , IK', KSI received from the MME. The UE shall replace all stored parameters CK, IK, KSI, if any, with CK' , IK', KSI in both ME and USIM. START shall be reset to 0. For the definition of the Key Derivation Function see Annex A.

NOTE 1: The new derived security conterxt (including CK", IK" and START value) replacing the stored values in the USIM is for allowing to reuse the derived security context without invoking the authentication procedure in subsequent connection set-ups, and also for avoiding that one KSI value indicates to two different key sets and consequently leads to security context desynchronization.

NOTE 2: An operator concerned about the security of keys received from an E-UTRAN of another operator may want to enforce a policy in SGSN to run a UMTS AKA as soon as possible after the handover. One example of ensuring this is the deletion of the mapped UMTS security context in the SGSN after the UE has left active state.

MME shall also provide at least the 4 LSB of the current NAS downlink COUNT value to the source eNB, which then shall include the bits to the MobilityFromE-UTRANCommand to the UE.

MME shall transfer the UE security capabilties to the SGSN. The selection of the algorithms in the target system proceeds as described in TS 33.102 [4] for UTRAN.

9.2.2 From UTRAN to E-UTRAN

9.2.2.1 Procedure

The procedure for handover from UTRAN to E-UTRAN, as far as relevant for security, proceeds in the following two consecutive steps:

A) Handover signalling using the mapped security context (cf. also Figure 9.2.2.1-1);

B) Subsequent NAS signalling to determine whether a native context is taken in use (not shown in Figure).

The activation of NAS and AS security in E-UTRAN, and selection of the key set from the source system for the handover shall be according to following principles:

i) As described for inter-SGSN PS handover cases in TS 33.102 [4], the source SGSN shall select the key set most recently generated (either by a successful AKA run or mapping from an EPS security context) and transfer this key set to the MME in the Forward Relocation Request.

NOTE x: The MME is considered as a target SGSN in case of Gn/Gp interface.

ii) Activation of AS security (for details cf. TS 36.331 [21]):

The E-UTRAN HO command received at the UE shall activate AS security.

The HO Complete received at the eNB shall activate AS security.

iii) Activation of NAS security (for details cf. TS 24.301 [9]):

The E-UTRAN HO command received at the UE shall activate NAS security.

The HO Notify received at the MME shall activate NAS security. In case the MME does not have the UE security capabilities stored from a previous visit, then no NAS message shall be sent or accepted by the MME other than a TAU request before a successful check of the UE security capabilities in the TAU request was performed by the MME.

iv) Both AS and NAS ciphering and integrity protection algorithms shall be selected according to the policy of the target PLMN.

The above four principles consequentially always activate ciphering (potentially NULL ciphering) in E-UTRAN even if it was not active in the source system.

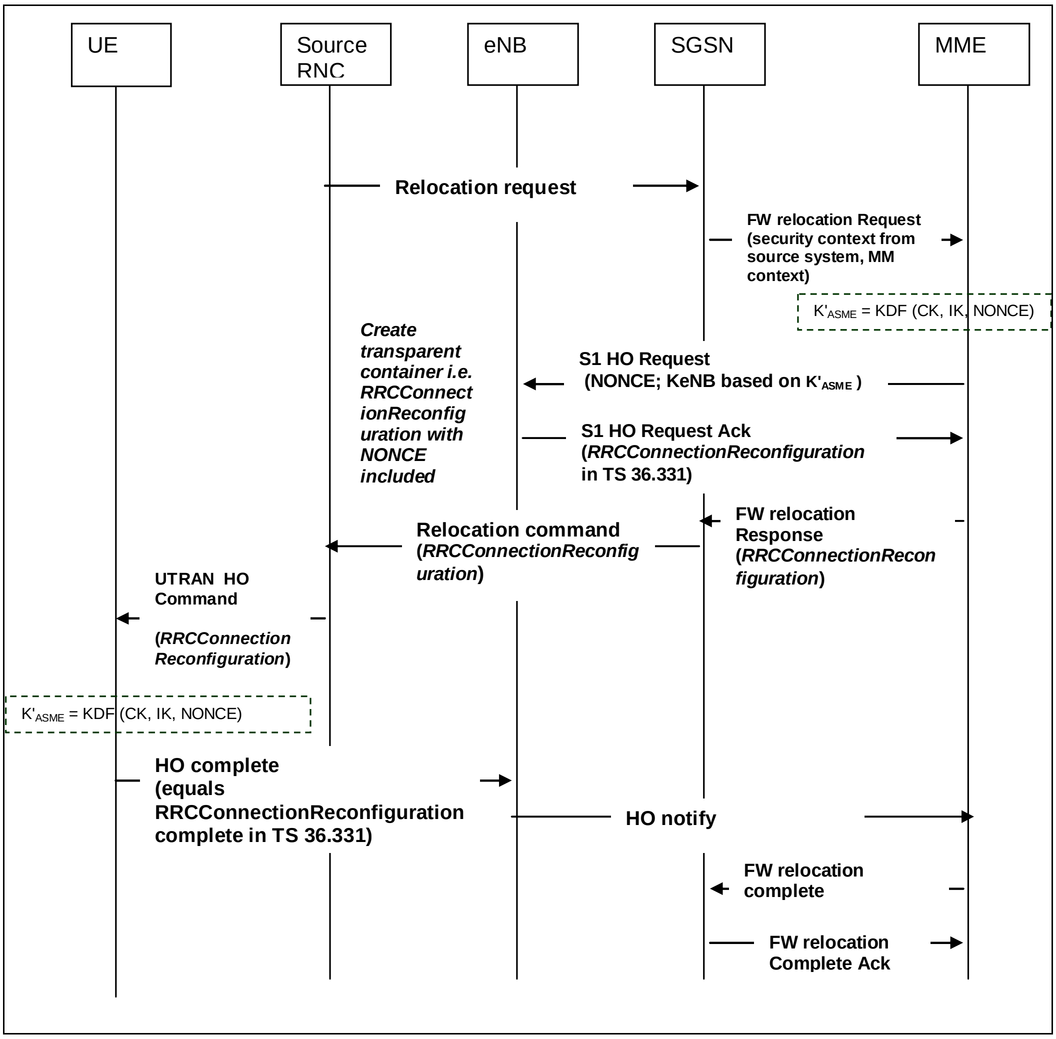

Figure 9.2.2.1-1: Handover from UTRAN to E-UTRAN

A) Handover signalling in case of successful handover

The RNC shall send a Relocation Request message to the SGSN. This message does not contain any security-relevant parameters.

1. The SGSN shall transfer MM context (including CK and IK (or the Kc), KSI and the UE security capabilities) to MME in the Forward relocation request message. In case the MM context in the Forward relocation request message indicates GSM security mode(i.e., it contains a Kc), the MME shall abort the procedure. The UE security capabilities, including the UE EPS security capabilities, were sent by the UE to the SGSN via the MS Network Capability IE, that is extended to include also UE EPS security capabilities, in Attach Request and RAU Request. It is possible that an SGSN does not forward the UE EPS security capabilities to the MME. When the MME does not receive UE EPS security capabilities from the SGSN, the MME shall assume that the following default set of EPS security algorithms is supported by the UE (and shall set the UE EPS security capabilities in the mapped EPS NAS security context according to this default set):
   - EEA0, 128-EEA1 and 128-EEA2 for NAS signalling ciphering, RRC signalling ciphering and UP ciphering;
   - 128-EIA1 and 128-EIA2 for NAS signalling integrity protection and RRC signalling integrity protection.

NOTE 0: Subclauses 5.1.3.2 and 5.1.4.2 of this specification mandate the UE to support the default set of EPS security algorithms, so, for the Rel-8 version of this specification, the default set of EPS security algorithms includes all security algorithms standardised for EPS. The notion of default set of EPS security algorithms is introduced here in order to make this specification future-proof as more security algorithms may be standardised for EPS in future releases.

2. The MME shall create a $NONCE_{MME}$ to be used in the K'_{ASME} derivation (see clause A.10 for requirements on the randomness of $NONCE_{MME}$).. MME shall derive K'_{ASME} from CK and IK with the help of a one-way key

derivation function as defined in Annex A and associate it with a Key Set Identifier KSI_{SGSN}. The value field of the KSI_{SGSN} shall be derived by assigning the KSI corresponding to the set of keys most recently generated (either by a successful AKA run or mapping from an EPS security context). MME shall derive the NAS keys and K_{eNB} from K'_{ASME}. The uplink and downlink NAS COUNT values for the mapped security context shall be reset to start value (i.e. 0) in the UE and the MME.

3. MME shall select the NAS security algorithms, MME shall include KSI_{SGSN}, $NONCE_{MME}$, the selected NAS security algorithms in the NAS Security Transparent Container IE of S1 HO Request message to the target eNB. MME further shall include K_{eNB} and the UE EPS security capabilities, either the capabilities received from the SGSN or, in the absence of these, the default set of EPS security algorithms, in the S1 HO Request message to the target eNB.

4. The target eNB shall select the RRC and UP algorithms based on the UE EPS Security Capabilities. The target eNB shall create a transparent container (RRCConnectionReconfiguration) including the selected RRC, UP algorithms and the NAS Security Transparent Container IE, and send it in the S1 HO Request Ack message towards the MME.

NOTE 1: This transparent container is not protected by the target eNB.

5. MME shall include the transparent container received from the target eNB in the FW Relocation Response messgage sent to SGSN.

6. SGSN shall include the transparent container in the relocation command sent to the RNC.

7. The RNC shall include the transparent container in the UTRAN HO command sent to the UE.

NOTE 2: The UTRAN HO command is integrity protected and optionally ciphered as specified by TS 33.102 [4].

8. The UE shall derive K'_{ASME} in the same way the MME did in step 2, associate it with KSI_{SGSN} and derive NAS, RRC and UP keys accordingly. The UE shall send a RRCConnectionReconfiguration Complete messages to the eNB. The uplink and downlink NAS COUNT values for the mapped context shall be set to start value (i.e. 0) in the UE and the MME.

9. The mapped EPS security context shall become the current (cf. subclause 3.1) EPS security context at AS and NAS level and overwrite any existing current mapped EPS security context. If the current security context is of type native, then it shall become the non-current native security context and overwrite any exisiting non-current security context. The HO Complete messages and all following AS messages in E-UTRAN shall be ciphered and integrity protected according to the policy of the target PLMN.

B) Subsequent NAS signalling

In order to prevent that successful bidding down on the UE security capabilities in a previous RAT have an effect on the selection of EPS security algorithm for NAS and AS, the UE security capabilities shall be included in the TAU request after IRAT-HO and be verified by the MME.

NOTE 3: Any TAU request following the handover will be integrity protected. Details are described in subclause 9.2.2.1

In any case UE security capability information received from the UE overwrites any capabilities received with the context transfer as specified in TS 23.401 [2].

It can happen that the MME receives UE security capabilities in the TAU Request that contains an algorithm with higher priority (according to the priority list stored in the MME) than any of the algorithms the MME received from the source SGSN. It can also happen that the MME uses the default set of EPS security algorithms for the UE according to A) step 1 above, and the TAU Request contains an algorithm with higher priority (according to the priority list stored in the MME) than the default set. If any of these cases happen, the MME shall run a NAS security mode command procedure to change the NAS algorithms according to subclause 7.2.4.4 and a S1 CONTEXT MODIFICATION procedure to inform the eNB about the correct UE EPS security capabilities and trigger a change of AS algorithms.

1. If the MME has native security context for the UE and does not receive a TAU request within a certain period after the HO it shall assume that UE and MME share a native security context.

NOTE 4: A TAU procedure following handover from UTRAN to E-UTRAN is mandatory if the Tracking Area has changed, but optional otherwise, cf. TS 23.401[2].

2. When the UE sends a TAU request it shall protect the request using the mapped EPS security context identified by KSI_{SGSN}. The UE shall also include KSI_{ASME} in the TAU request if and only if it has native EPS security context. The KSI_{ASME} shall be accompanied by a GUTI. When the MME receives a TAU request with a KSI_{ASME} and GUTI corresponding to the native EPS security context stored on that MME it knows that UE and MME share a native security context.

3. Void.

NOTE 5: Void.

4. When the MME receives a TAU request without a KSI_{ASME} it shall delete any native EPS security context for any GUTI it may have for the user who sent the TAU request.

5. If the MME shares the native EPS security context indexed by the KSI_{ASME} and GUTI from the TAU Request with the UE, the MME may run a NAS security mode command procedure with the UE to activate the native EPS NAS security context according to clause 7.2.9.4. The MME may in addition change the K_{eNB} on the fly according to clause 7.2.9.2). In case the GUTI received in the TAU Request message pointed to a different MME, the allocation of a new GUTI, replacing the received GUTI, and the association of this new GUTI with KSI_{ASME} is required.

6. Void.

NOTE 6: The TAU Request is integrity protected with the mapped EPSsecurity context even if the UE and the MME share a native EPSsecurity context since the UE cannot know for sure if the MME still has the native EPS security context at the time of sending the TAU Request.

7. When the MME knows, after having completed the TAU procedure in the preceding steps, that it shares a native EPS security context with the UE, the MME may (depending on configured policy and if the MME did not do it already in step 5) activate this native EPS security context. This activation may occur in three ways:
   a. During ECM-CONNECTED state: the MME shall initiate a key change on the fly procedure according to subclause 7.2.9 for the entire EPS key hierarchy.
   b. After the next transition to ECM-IDLE state following the handover from UTRAN: Upon receiving the first message from the UE after the UE has gone to ECM-IDLE state the MME shall use the procedures defined in subclauses 7.2.4.4 and 7.2.4.5 to activate the native EPS security context.
   c. At the next transition to EMM-DEREGISTERED (see clause 7.2.5.1).

8. If native EPS security context has been established, then the UE and the MME shall delete the mapped EPS security context and set the native EPS security context to the current EPS security context.

NOTE 7: The run of an NAS SMC procedures ensures that the uplink NAS COUNT has increased since the last time a K_{eNB} was derived from the K_{ASME}.

NOTE 8: For the handling of native and mapped contexts after a state transition to EMM-DEREGISTERED cf. subclause 7.2.5.1.

9.2.2.2 Derivation of NAS keys and K_{eNB} during Handover from UTRAN to E-UTRAN

MME and UE shall derive the NAS keys from the mapped key K'_{ASME} as specified in Annex A.

MME and UE shall derive K_{eNB} from K'_{ASME} as follows:

The MME sets NAS COUNT equal to zero and uses it with the mapped key K'_{ASME} to derive K_{eNB} by applying the KDF defined in Annex A for IDLE to CONNECTED transition.

9.3 Recommendations on AKA at IRAT-mobility to E-UTRAN

After a handover from GERAN or UTRAN into E-UTRAN, it is strongly recommended to run an AKA and perform a key change on-the-fly of the entire key hierarchy as soon as possible after the handover if there is no native security context in E-UTRAN.

When a UE moves in IDLE mode from GERAN or UTRAN into E-UTRAN, it is strongly recommended to run an AKA if there is no native security context in E-UTRAN, either after the TAU procedure that establishes an EPS security context in the MME and UE, or when the UE transits into ECM-CONNECTED state.

10 Security interworking between E-UTRAN and GERAN

10.1 General

An SGSN supporting interworking between E-UTRAN and GERAN is capable of handling UMTS security contexts and supports the key conversion function c3 specified in TS 33.102 [4]. Furthermore, as a consequence of the UE being able to access EPS, the user has a USIM, and the ME and the HSS are UMTS-capable. Hence, UMTS AKA is used when the UE is authenticated even when attached to GERAN, and UMTS security contexts are available. The security procedures for interworking between E-UTRAN and GERAN are therefore quite similar to those between E-UTRAN and UTRAN.

10.2 Idle mode procedures

10.2.1 Idle mode procedures in GERAN

This subclause covers both the cases of idle mode mobility from E-UTRAN to GERAN and of Idle Mode Signaling Reduction, as defined in TS 23.401 [2].

As the SGSN is capable of handling UMTS security contexts clause 9.1.1 applies here with the following changes

- the SGSN and UE shall derive Kc from CK' and IK' with the help of the key conversion function c3 of TS 33.102;
- SGSN shall select the encryption algorithm to use in GERAN.

10.2.2 Idle mode procedures in E-UTRAN

This subclause covers both the cases of idle mode mobility from GERAN to E-UTRAN and of Idle Mode Signaling Reduction, as defined in TS 23.401 [2].

As the SGSN shares a UMTS security context with the UE clause 9.1.2 applies here without changes.

10.3 Handover

10.3.1 From E-UTRAN to GERAN

As the SGSN is capable of handling UMTS security contexts clause 9. 2.1 applies here with the following changes:

- SGSN and UE shall derive Kc from CK' and IK' with the help of the key conversion function c3 of TS 33.102.
- SGSN shall select the encryption algorithm to use in GERAN after handover.
- Whether ciphering is considered active in the target GERAN after handover from E-UTRAN shall be determined according to the principles for handover to GERAN in TS 44.060 [25].

10.3.2 From GERAN to E-UTRAN

10.3.2.1 Procedures

As the SGSN shares a UMTS security context with the UE clause 9.2.2 applies here without changes.

10.4 Recommendations on AKA at IRAT-mobility to E-UTRAN

See recommendation provided by subclause 9.3.

11 Network Domain Control Plane protection

The protection of IP based control plane signalling for EPS and E-UTRAN shall be done according to TS 33.210 [5].

NOTE 1: In case control plane interfaces are trusted (e.g. physically protected), there is no need to use protection according to TS 33.210 [5].

In order to protect the S1 and X2 control plane, it is required to implement IPsec ESP according to RFC 4303 [7] as specified by TS 33.210 [5]. For both S1-MME and X2-C, IKEv2 certificates based authentication according to TS 33.310 [6] shall be implemented. For S1-MME and X2-C, tunnel mode IPsec is mandatory to implement on the eNB. On the core network side a SEG may be used to terminate the IPsec tunnel.

Transport mode IPsec is optional for implementation on the X2-C and S1-MME.

NOTE 2: Transport mode can be used for reducing the protocol overhead added by IPsec.

12 Backhaul link user plane protection

The protection of user plane data between the eNB and the UE by user specific security associations is covered by clause 5.1.3 and 5.1.4.

In order to protect the S1 and X2 user plane as required by clause 5.3.3, it is required to implement IPsec ESP according to RFC 4303 [7] as profiled by TS 33.210 [5], with confidentiality, integrity and replay protection.

On the X2-U and S1-U, transport mode IPsec is optional for implementation.

NOTE 1: Transport mode can be used for reducing the protocol overhead added by IPsec.

Tunnel mode IPsec is mandatory to implement on the eNB for X2-U and S1-U. On the core network side a SEG may be used to terminate the IPsec tunnel..

For both S1 and X2 user plane, IKEv2 with certificates based authentication shall be implemented. The certificates shall be implemented according to the profile described by TS 33.310 [6]. IKEv2 shall be implemented conforming to the IKEv2 profile described in TS 33.310 [6]

NOTE 2: In case S1 and X2 user plane interfaces are trusted (e.g. physically protected), the use of IPsec/IKEv2 based protection is not needed.

13 Management plane protection over the S1 interface

Clause 5.3.2 requires that eNB setup and configuration traffic, i.e. the management plane, to be protected between the EPS core and the eNB. This traffic is typically carried over the same backhaul link as the S1 interface. Therefore, the protection mechanism defined for S1-MME and S1-U may be re-used for S1 management plane, S1-M.

In this case and in order to achieve such protection, it is required to implement IPsec ESP according to RFC 4303 [7] as profiled by TS 33.210 [5], with confidentiality, integrity and replay protection.

Tunnel mode IPsec is mandatory to implement on the eNB for supporting the S1 management plane. On the core network side a SEG may be used to terminate the IPsec tunnel. If no SEG is used, the IPsec tunnel may be terminated in the element manager.

For the S1 management plane, IKEv2 with certificates based authentication shall be implemented on the eNB. The certificates shall be implemented according to the profile described by TS 33.310 [6]. IKEv2 shall be implemented conforming to the IKEv2 profile described in TS 33.310 [6]

NOTE 1: X2 does not carry management plane traffic.

NOTE 2: In case the S1 management plane interfaces are trusted (e.g. physically protected), the use of IPsec/IKEv2 based protection is not needed

14 SRVCC between E-UTRAN and Circuit Switched UTRAN/GERAN

14.1 From E-UTRAN to Circuit Switched UTRAN/GERAN

Single Radio Voice Call Continuity (SRVCC) is specified in 3GPP TS 23.216 [22].

The MME and the UE shall derive a confidentiality key CK_{SRVCC}, and an integrity key IK_{SRVCC} from K_{ASME} and the NAS downlink COUNT with the help of a one-way key derivation function KDF as specified in Annex A.

The KDF returns a 256-bit output, where the 128 most significant bits are identified with CK_{SRVCC} and the 128 least significant bits are identified with IK_{SRVCC}.

The MME shall also provide the 4 LSB of the current NAS downlink COUNT value to the source eNB, which then includes the bits to the HO Command to the UE.

UE and MME shall assign the value of eKSI to KSI. MME shall transfer CK_{SRVCC}, IK_{SRVCC} with KSI and the UE security capability to the enhanced MSC server. The enhanced MSC server shall replace the stored parameters CK, IK, KSI, if any, with CK_{SRVCC}, IK_{SRVCC}, KSI received from the MME. The UE shall replace the stored parameters CK, IK, KSI, if any, with CK_{SRVCC}, IK_{SRVCC}, KSI in both ME and USIM. START shall be reset to 0.

NOTE 1: The new derived security context (including CK_{SRVCC}, IK_{SRVCC} , KSI and START value) replacing the stored values in the USIM is for allowing to reuse the derived security context without invoking the authentication procedure in subsequent connection set-ups, and also for avoiding that one KSI value indicates to two different key sets and consequently leads to security context desynchronization.

NOTE 2: An operator concerned about the security of keys received from an E-UTRAN of another operator may want to enforce a policy in the enhanced MSC server to run a UMTS AKA as soon as possible after the handover. One example of ensuring this is the deletion of the mapped UMTS security context in the the enhanced MSC server after the UE has left active state.

If the SRVCC is from E-UTRAN to GERAN, the enhanced MSC server and the UE shall derive Kc from CK_{SRVCC} and IK_{SRVCC} with the help of the key conversion function c3 as specified in TS 33.102 [4]. The UE and the enhanced MSC Server shall assign the value of eKSI to CKSN.

NOTE: Non-voice bearers may be handed over during the SRVCC handover operation. Key derivation for non-voice bearers is specified in clause 9 of the present specification.

Annex A (normative): Key derivation functions

A.1 KDF interface and input parameter construction

A.1.1 General

The input parameters and their lengths shall be concatenated into a string S as follows:

1. The length of each input parameter encoding measured in octets shall be encoded into a two octets long string:

   a) express the number of octets in input parameter Pi as a number k in the range [0, 65535];

   b) Li is then a two-octet representation of the number k written in base 2 and using the bit ordering specified in clause 3.4. Any unused most significant bits of Li shall be set to zero.

EXAMPLE: If Pi contains 258 octets then Li will be the two-octet bit-string $(0000000100000010)_2$, or 0x01 0x02 in hex notation.

2. Given a non-negative integer j expressing the value to be encoded in Pi, Pi shall be formed by writing j in base 2. The least significant bit of Pi shall be equal to the least significant bit of j, i.e., according to clause 3.4 of this specification. Any unused most significant bits of Pi shall be set to zero to meet the octet length prescribed by Li.

EXAMPLE: If Pi is the integer value 259 and the the length of parameter Pi is two octets, Pi consists of the bit-pattern $(0000000100000011)_2$ or 0x01 0x03 in hex representation.

3. String S shall be constructed from n input parameters as follows:

   S = FC || P0 || L0 || P1 || L1 || P2 || L2 || P3 || L3 ||... || Pn || Ln

   where:

   FC is single octet used to distinguish between different instances of the algorithm,

   P0 ... Pn are the n input parameter encodings, and

   L0 ... Ln are the two-octet representations of the length of the corresponding input parameter encodings P0 ... Pn.

4. The final output, i.e. the derived key is equal to the KDF computed on the string S using the key Key. The present document defines the following KDF:

   derived key = HMAC-SHA-256 (Key, S),

as specified in [10] and [11], which has the KDF identity 1.

All key derivations for EPS shall be performed using the key derivation function (KDF) specified in this Annex. This clause specifies the set of input strings, S_i, to the KDF (which are input together with the relevant key). For each of the distinct usages of the KDF, the input parameters S_i are specified below.

A.1.2 FC value allocations

The FC number space is used controlled by TS 33.220 [8], FC values allocated for this specification are in range of 0x10 – 0x1F.

A.2 KASME derivation function (S_{10})

When deriving a K_{ASME} from CK, IK and SN id when producing authentication vectors, and when the UE computes K_{ASME} during AKA, the following parameters shall be used to form the input S to the KDF.

- FC = 0x10,
- P0 = SN id,
- L0 = length of SN id (i.e. 0x00 0x03),
- P1 = SQN ⊕ AK
- L1 = length of SQN ⊕ AK (i.e. 0x00 0x06)

NOTE: The string S indexes start from 10 to align with the FC values and Annex subclause numbering.

The exclusive or of the Sequence Number (SQN) and the Anonymity Key (AK) is sent to the UE as a part of the Authentication Token (AUTN), see TS 33.102. If AK is not used, AK shall be treated in accordance with TS 33.102, i.e. as 000…0.

The SN id consists of MCC and MNC, and shall be encoded as an octet string according to Figure A.2-1.

8 7 6 5	4 3 2 1	
MCC digit 2	MCC digit 1	octet 1
MNC digit 3	MCC digit 3	octet 2
MNC digit 2	MNC digit 1	octet 3

Figure A.2-1 Encoding of SN id as an octet string

The coding of the digits of MCC and MNC shall be done according to TS 24.301 [9].

The input key Key shall be equal to the concatenation CK || IK of CK and IK.

A.3 K_{eNB} derivation function used at ECM-IDLE to ECM-CONNECTED transition, ECM-IDLE mode mobility, transition away from EMM-DEREGISTERED to EMM-REGISTERED/ECM-CONNECTED, key change on-the-fly and TAU and handover from UTRAN/GERAN to EUTRAN (S_{11})

When deriving a K_{eNB} from K_{ASME} and the uplink NAS COUNT in the UE and the MME the following parameters shall be used to form the input S to the KDF. During TAU and handover from UTRAN/GERAN to EUTRAN and when mapped context is used, the uplink NAS COUNT is set to 0 by the UE and the MME.

- FC = 0x11,
- P0 = Uplink NAS COUNT,
- L0 = length of uplink NAS COUNT (i.e. 0x00 0x04)

The input key shall be the 256-bit K_{ASME}.

A.4 NH derivation function (S_{12})

When deriving a NH from K_{ASME} the following parameters shall be used to form the input S to the KDF.

- FC = 0x12
- P0 = SYNC-input
- L0 = length of SYNC-input (i.e. 0x00 0x20)

The SYNC-input parameter shall be the newly derived K_{eNB} for the initial NH derivation, and the previous NH for all subsequent derivations. This results in a NH chain, where the next NH is always fresh and derived from the previous NH.

The input key shall be the 256-bit K_{ASME}.

A.5 K_{eNB}* derivation function (S_{13})

When deriving a K_{eNB}* from current K_{eNB} or from fresh NH and the target physical cell ID in the UE and eNB as specified in clause 7.2.8 for handover purposes the following parameters shall be used to form the input S to the KDF.

- FC = 0x13
- P0 = PCI (target physical cell id)
- L0 = length of PCI (i.e. 0x00 0x02)
- P1 = EARFCN-DL (target physical cell downlink frequency)
- L1 length of EARFCN-DL (i.e. 0x00 0x02)

The input key shall be the 256-bit NH when the index in the handover increases, otherwise the current 256-bit K_{eNB}.

A.6 Void

A.7 Algorithm key derivation functions (S_{15})

When deriving keys for NAS integrity and NAS encryption algorithms from K_{ASME} and algorithm types and algorithm IDs, and keys for RRC integrity and RRC/UP encryption algorithms from K_{eNB}, in the UE, MME and eNB the following parameters shall be used to form the string S.

- FC = 0x15
- P0 = algorithm type distinguisher
- L0 = length of algorithm type distinguisher (i.e. 0x00 0x01)
- P1 = algorithm identity
- L1 = length of algorithm identity (i.e. 0x00 0x01)

The algorithm type distinguisher shall be NAS-enc-alg for NAS encryption algorithms and NAS-int-alg for NAS integrity protection algorithms. The algorithm type distinguisher shall be RRC-enc-alg for RRC encryption algorithms, RRC-int-alg for RRC integrity protection algorithms and UP-enc-alg for UP encryption algorithms (see table A.6-1). The values 0x06 to 0xf0 are reserved for future use, and the values 0xf1 to 0xff are reserved for private use.

Table A.7-1: Algorithm type distinguishers

Algorithm distinguisher	Value
NAS-enc-alg	0x01
NAS-int-alg	0x02
RRC-enc-alg	0x03
RRC-int-alg	0x04
UP-enc-alg	0x05

The algorithm identity (as specified in clause 5) shall be put in the four least significant bits of the octet. The two least significant bits of the four most significant bits are reserved for future use, and the two most significant bits of the most significant nibble are reserved for private use. The entire four most significant bits shall be set to all zeros.

For NAS algorithm key derivations, the input key shall be the 256-bit K_{ASME}, and for UP and RRC algorithm key derivations, the input key shall be the 256-bit K_{eNB}.

For an algorithm key of length n bits, where n is less or equal to 256, the n least significant bits of the 256 bits of the KDF output shall be used as the algorithm key.

A.8 K_{ASME} to CK', IK' derivation (S_{16})

This input string is used when there is a need to derive CK' || IK' from K_{ASME} during mapping of security contexts from E-UTRAN to GERAN/UTRAN. K_{ASME} is a 256-bit entity, and so is the concatenation of CK and IK (which are 128 bits each). The following input parameters shall be used.

- FC = 0x16
- P0 = NAS downlink COUNT value
- L0 = length of NAS downlink COUNT value (i.e. 0x00 0x04)

The input key shall be K_{ASME}.

A.9 NAS token derivation for inter-RAT mobility (S_{17})

The NAS-token used to ensure that a RAU is originating from the correct UE during IDLE mode mobility from E-UTRAN to UTRAN and GERAN, shall use the following input parameters.

- FC = 0x17
- P0 = Downlink NAS COUNT
- L0 = length of downlink NAS COUNT (i.e. 0x00 0x04)

The input key shall be the 256-bit K_{ASME}.

A.10 K''_{ASME} from CK, IK derivation during handover (S_{18})

This input string is used when there is a need to derive a K'_{ASME} from concatenation of CK and IK and a $NONCE_{MME}$ during mapping of security contexts between GERAN/UTRAN and E-UTRAN during handover to E-UTRAN.

K'_{ASME} is a 256-bit value. The $NONCE_{MME}$ is a 32-bit value. The following input parameters shall be used.

- FC = 0x18
- P0 = $NONCE_{MME}$
- L0 = length of $NONCE_{MME}$ (i.e. 0x00 0x04)

The input key shall be the concatenation of CK || IK.

The generation of $NONCE_{MME}$ shall be sufficiently random such that both the probability of the MME generating equal values of $NONCE_{MME}$ and the probability of an attacker being able to predict future values of $NONCE_{MME}$ over the duration of practical eavesdropping attacks on a particular user are extremely low.

NOTE: A well-seeded strong PRNG would meet this requirement. A true RNG is not required.

A.11 K''_{ASME} from CK, IK derivation during idle mode mobility (S_{19})

This input string is used when there is a need to derive a K_{ASME} from CK || IK, $NONCE_{UE}$, and $NONCE_{MME}$ during mapping of security contexts from GERAN/UTRAN to E-UTRAN. K_{ASME} is a 256-bit entity, and so is the concatenation of CK and IK (which are 128 bits each). The following input parameters shall be used, where NONCEs are 32 bits long.

- FC = 0x19,
- P0 = $NONCE_{UE}$
- L0 = length of the $NONCE_{UE}$ (i.e. 0x00 0x04)
- P1 = $NONCE_{MME}$
- L1 = length of the $NONCE_{MME}$ (i.e. 0x00 0x04)

The input key shall be the concatenation of CK || IK.

The generation of $NONCE_{UE}$ shall be sufficiently random such that both the probability of the UE generating equal values of $NONCE_{UE}$ and the probability of an attacker being able to predict future values of $NONCE_{UE}$ over the duration of practical eavesdropping attacks on a particular user are extremely low.

NOTE: A well-seeded strong PRNG would meet this requirement. A true RNG is not required.

The generation of $NONCE_{MME}$ shall be as defined in clause A.10.

A.12 K_{ASME} to CK_{SRVCC}, IK_{SRVCC} derivation (S_{1A})

This input string is used when there is a need to derive CK_{SRVCC}|| IK_{SRVCC} used in CS domain from K_{ASME} during mapping of security contexts between E-UTRAN and GERAN/UTRAN. K_{ASME} is a 256-bit element, and so is the concatenation of CK_{SRVCC} and IK_{SRVCC} (which are 128 bits each).

- FC = 0x1A
- P0 = NAS downlink COUNT value
- L0 = length of NAS downlink COUNT value (i.e. 0x00 0x04)

The input key shall be K_{ASME}.

Annex B (normative): Algorithms for ciphering and integrity protection

B.0 EEA0 Null ciphering algorithm

The EEA0 algorithm shall be implemented such that it has the same effect as if it generates a KEYSTREAM of all zeroes (see subclause B.1.1). The length of the KEYSTREAM generated shall be equal to the LENGTH input parameter. The generated KEYSTREAM requires no other input parameters but the LENGTH. Apart from this, all processing performed in association with ciphering shall be exactly the same as with any of the ciphering algorithms specified in this annex.

NOTE: that EEA0 provides no security.

B.1 128-bit ciphering algorithm

B.1.1 Inputs and outputs

The input parameters to the ciphering algorithm are a 128-bit cipher key named KEY, a 32-bit COUNT, a 5-bit bearer identity BEARER, the 1-bit direction of the transmission i.e. DIRECTION, and the length of the keystream required i.e. LENGTH. The DIRECTION bit shall be 0 for uplink and 1 for downlink.

Figure B.1-1 illustrates the use of the ciphering algorithm EEA to encrypt plaintext by applying a keystream using a bit per bit binary addition of the plaintext and the keystream. The plaintext may be recovered by generating the same keystream using the same input parameters and applying a bit per bit binary addition with the ciphertext.

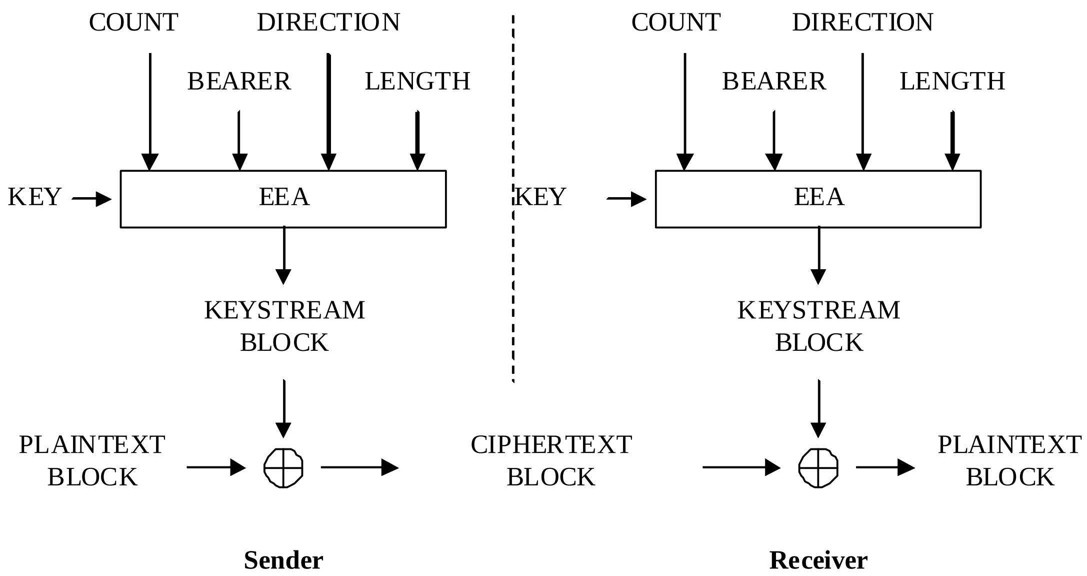

Figure B.1-1: Ciphering of data

Based on the input parameters the algorithm generates the output keystream block KEYSTREAM which is used to encrypt the input plaintext block PLAINTEXT to produce the output ciphertext block CIPHERTEXT.

The input parameter LENGTH shall affect only the length of the KEYSTREAM BLOCK, not the actual bits in it.

B.1.2 128-EEA1

128-EEA1 is based on SNOW 3G and is identical to UEA2 as specified in [14]. The used IV is constructed the same way as in subclause 3.4 of that TS.

B.1.3 128-EEA2

128-EEA2 is based on 128-bit AES [15] in CTR mode [16]

The sequence of 128-bit counter blocks needed for CTR mode T_1, T_2, …, T_i, … shall be constructed as follows:

The most significant 64 bits of T_1 consist of COUNT[0] .. COUNT[31] │ BEARER[0] .. BEARER[4] │ DIRECTION │ 0^{26} (i.e. 26 zero bits). These are written from most significant on the left to least significant on the right, so for example COUNT[0] is the most significant bit of T_1.

The least significant 64 bits of T_1 are all 0.

Subsequent counter blocks are then obtained by applying the standard integer incrementing function (according to Appendix B1 in [16]) mod 2^{64} to the least significant 64 bits of the previous counter block.

B.2 128-Bit integrity algorithm

B.2.1 Inputs and outputs

The input parameters to the integrity algorithm are a 128-bit integrity key named KEY, a 32-bit COUNT, a 5-bit bearer identity called BEARER, the 1-bit direction of the transmission i.e. DIRECTION, and the message itself i.e MESSAGE. The DIRECTION bit shall be 0 for uplink and 1 for downlink. The bit length of the MESSAGE is LENGTH.

Figure B.2-1 illustrates the use of the integrity algorithm EIA to authenticate the integrity of messages.

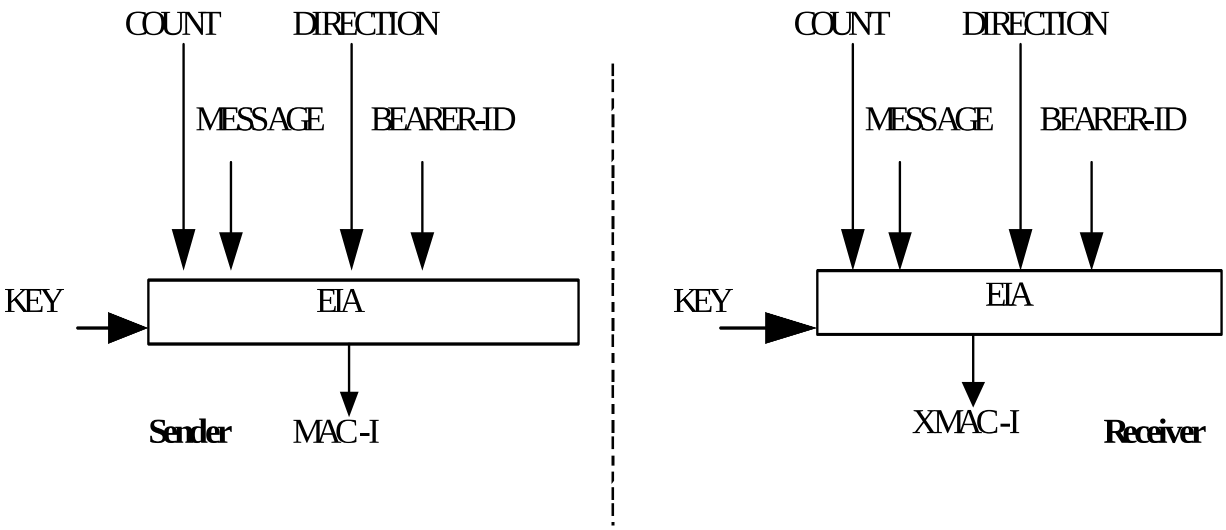

Figure B.2-1: Derivation of MAC-I/NAS-MAC (or XMAC-I/XNAS-MAC)

Based on these input parameters the sender computes a 32-bit message authentication code (MAC-I/NAS-MAC) using the integrity algorithm EIA. The message authentication code is then appended to the message when sent. The receiver computes the expected message authentication code (XMAC-I/XNAS-MAC) on the message received in the same way as the sender computed its message authentication code on the message sent and verifies the data integrity of the message by comparing it to the received message authentication code, i.e. MAC-I/NAS-MAC.

B.2.2 128-EIA1

128-EIA1 is based on SNOW 3G and is implemented in the same way as UIA2 as specified in [14]. The used IV is constructed the same way as in subclause 4.4 of that TS, with the only difference being that FRESH [0], … FRESH [31] shall be replaced by BEARER[0] … BEARER[4] $|$ 0^{27} (i.e. 27 zero bits)

B.2.3 128-EIA2

128-EIA2 is based on 128-bit AES [15] in CMAC mode [17].

The bit length of MESSAGE is BLENGTH.

The input to CMAC mode is a bit string M of length Mlen (see [18, section 5.5]). M is constructed as follows:

M_0 .. M_{31} = COUNT[0] .. COUNT[31]

M_{32} .. M_{36} = BEARER[0] .. BEARER[4]

M_{37} = DIRECTION

M_{38} .. M_{63} = 0^{26} (i.e. 26 zero bits)

M_{64} .. $M_{BLENGTH+63}$ = MESSAGE[0] .. MESSAGE[BLENGTH-1]

and so Mlen = BLENGTH + 64.

AES in CMAC mode is used with these inputs to produce a Message Authentication Code T (MACT) of length Tlen = 32. T is used directly as the 128-EIA2 output MACT[0] .. MACT[31], with MACT[0] being the most significant bit of T.

Annex C (informative): Algorithm test data

C.1 128-EEA2

This section includes eight test data sets; all are presented in hex, while the first is also presented in binary. Some intermediate computational values are included to assist implementers in tracing bugs. Some notation is taken from the specification of CTR mode [16].

Bit ordering should be largely self explanatory, but in particular:

- The 5-bit BEARER is written in hex in a 'right aligned' form, i.e. as a two-hex-digit value in the range 00 to 1F inclusive, with BEARER [0] as the msb of the first digit.
- Similarly the single DIRECTION bit is written in hex in 'right aligned' form, i.e. the DIRECTION bit is the lsb of the hex digit.
- Where the length of plaintext and ciphertext is not a multiple of 32 bits, they are written in hex in a 'left aligned' form, i.e. the least significant few bits of the last word will be zero.

C.1.1 Test Set 1

```
Key        = (hex) d3c5d592 327fb11c 4035c668 0af8c6d1

Key        = (bin) 11010011 11000101 11010101 10010010 00110010 01111111 10110001 00011100

                   01000000 00110101 11000110 01101000 00001010 11111000 11000110 11010001

Count      = (hex) 398a59b4

Count      = (bin) 00111001 10001010 01011001 10110100

Bearer     = (hex) 15

Bearer     = (bin) 10101

Direction  = (hex) 1

Direction  = (bin) 1

Length     = 253 bits

Plaintext  = (hex) 981ba682 4c1bfb1a b4854720 29b71d80 8ce33e2c c3c0b5fc 1f3de8a6 dc66b1f0

Plaintext  = (bin) 10011000 00011011 10100110 10000010 01001100 00011011 11111011 00011010

                   10110100 10000101 01000111 00100000 00101001 10110111 00011101 10000000

                   10001100 11100011 00111110 00101100 11000011 11000000 10110101 11111100

                   00011111 00111101 11101000 10100110 11011100 01100110 10110001 11110


Counter block T1   = (hex) 398a59b4 ac000000 00000000 00000000

Counter block T1   = (bin) 00111001 10001010 01011001 10110100 10101100 00000000 00000000 00000000

                           00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Keystream block 1  = (hex) 71e57e24 710ea81e 6398b52b da5f3f94

Keystream block 1  = (bin) 01110001 11100101 01111110 00100100 01110001 00001110 10101000 00011110

                           01100011 10011000 10110101 00101011 11011010 01011111 00111111 10010100

Counter block T2   = (hex) 398a59b4 ac000000 00000000 00000001
```

```
Counter block T2   = (bin) 00111001 10001010 01011001 10110100 10101100 00000000 00000000 00000000
                           00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001
Keystream block 2  = (hex) 3eede9f6 11328620 231f3f1b 328b3f88
Keystream block 2  = (bin) 00111110 11101101 11101001 11110110 00010001 00110010 10000110 00100000
                           00100011 00011111 00111111 00011011 00110010 10001011 00111111 10001000

Ciphertext = (hex) e9fed8a6 3d155304 d71df20b f3e82214 b20ed7da d2f233dc 3c22d7bd eeed8e78
Ciphertext = (bin) 11101001 11111110 11011000 10100110 00111101 00010101 01010011 00000100
                   11010111 00011101 11110010 00001011 11110011 11101000 00100010 00010100
                   10110010 00001110 11010111 11011010 11010010 11110010 00110011 11011100
                   00111100 00100010 11010111 10111101 11101110 11101101 10001110 01111
```

C.1.2 Test Set 2

```
Key        = 2bd6459f 82c440e0 952c4910 4805ff48
Count      = c675a64b
Bearer     = 0c
Direction  = 1
Length     = 798 bits
Plaintext  = 7ec61272 743bf161 4726446a 6c38ced1 66f6ca76 eb543004 4286346c ef130f92
             922b0345 0d3a9975 e5bd2ea0 eb55ad8e 1b199e3e c4316020 e9a1b285 e7627953
             59b7bdfd 39bef4b2 484583d5 afe082ae e638bf5f d5a60619 3901a08f 4ab41aab
             9b134880

Counter block T1   = c675a64b 64000000 00000000 00000000
Keystream block 1  = 27a77221 27fdbabd e67d5d34 44bd9d78
Counter block T2   = c675a64b 64000000 00000000 00000001
Keystream block 2  = 7695ef70 3d743aa3 d242fc6a 268a0b5d
Counter block T3   = c675a64b 64000000 00000000 00000002
Keystream block 3  = b66ecf15 b626681d 412b5dd3 a55db6d9
Counter block T4   = c675a64b 64000000 00000000 00000003
Keystream block 4  = f83d506c 9df187ad a578c902 ee14296f
Counter block T5   = c675a64b 64000000 00000000 00000004
Keystream block 5  = 50f44f36 635604e0 8ff25047 8c750516
Counter block T6   = c675a64b 64000000 00000000 00000005
Keystream block 6  = 735839e3 7ebe8579 7be34641 08f730bc
Counter block T7   = c675a64b 64000000 00000000 00000006
Keystream block 7  = 8b4f1b53 87da3277 a56f567d 8066fce2

Ciphertext = 59616053 53c64bdc a15b195e 288553a9 10632506 d6200aa7 90c4c806 c99904cf
             2445cc50 bb1cf168 a4967373 4e081b57 e324ce52 59c0e78d 4cd97b87 0976503c
```

```
             0943f2cb 5ae8f052 c7b7d392 239587b8 956086bc ab188360 42e2e6ce 42432a17
             105c53d0
```

C.1.3 Test Set 3

```
Key        = 0a8b6bd8 d9b08b08 d64e32d1 817777fb
Count      = 544d49cd
Bearer     = 04
Direction  = 0
Length     = 310 bits
Plaintext  = fd40a41d 370a1f65 74509568 7d47ba1d 36d2349e 23f64439 2c8ea9c4 9d40c132
             71aff264 d0f24800

Counter block T1   = 544d49cd 20000000 00000000 00000000
Keystream block 1  = 8835a92a 83b1bdc1 aa8ba14b 2691367b
Counter block T2   = 544d49cd 20000000 00000000 00000001
Keystream block 2  = 737eee32 87777c9a 9c4ad826 3a44db65
Counter block T3   = 544d49cd 20000000 00000000 00000002
Keystream block 3  = 158c20f6 a275b8f5 0e8ae073 997c58ed

Ciphertext = 75750d37 b4bba2a4 dedb3423 5bd68c66 45acdaac a48138a3 b0c471e2 a7041a57
             6423d292 7287f000
```

C.1.4 Test Set 4

```
Key        = aa1f95ae a533bcb3 2eb63bf5 2d8f831a
Count      = 72d8c671
Bearer     = 10
Direction  = 1
Length     = 1022 bits
Plaintext  = fb1b96c5 c8badfb2 e8e8edfd e78e57f2 ad81e741 03fc430a 534dcc37 afcec70e
             1517bb06 f27219da e49022dd c47a068d e4c9496a 951a6b09 edbdc864 c7adbd74
             0ac50c02 2f3082ba fd22d781 97c5d508 b977bca1 3f32e652 e74ba728 576077ce
             628c535e 87dc6077 ba07d290 68590c8c b5f1088e 082cfa0e c961302d 69cf3d44

Counter block T1   = 72d8c671 84000000 00000000 00000000
Keystream block 1  = 24afd669 7bcdeafb 0728abd5 49368fe7
Counter block T2   = 72d8c671 84000000 00000000 00000001
Keystream block 2  = cff4c44a df954e9e e34041a2 5d428c58
Counter block T3   = 72d8c671 84000000 00000000 00000002
Keystream block 3  = 2568dbf2 3827f27c 857b98af 68fa8925
Counter block T4   = 72d8c671 84000000 00000000 00000003
```

```
Keystream block 4  = 20576f12 1bca2154 8dd17c7c 19d93aff

Counter block T5   = 72d8c671 84000000 00000000 00000004

Keystream block 5  = 90e7f4ed 0669897e 16751e7b 6001c02c

Counter block T6   = 72d8c671 84000000 00000000 00000005

Keystream block 6  = 11f20436 a370d97d 68c5a2ba fee7e5cf

Counter block T7   = 72d8c671 84000000 00000000 00000006

Keystream block 7  = dcf3aa29 fdca4acf aaf961b4 d22dc84d

Counter block T8   = 72d8c671 84000000 00000000 00000007

Keystream block 8  = e31145b7 015ef36b f3a20e77 36e2b523


Ciphertext = dfb440ac b3773549 efc04628 aeb8d815 6275230b dc690d94 b00d8d95 f28c4b56

             307f60f4 ca55eba6 61ebba72 ac808fa8 c49e2678 8ed04a5d 606cb418 de74878b

             9a22f8ef 29590bc4 eb57c9fa f7c41524 a885b897 9c423f2f 8f8e0592 a9879201

             be7ff977 7a162ab8 10feb324 ba74c4c1 56e04d39 09720965 3ac33e5a 5f2d8864
```

C.1.5 Test Set 5

```
Key        = 9618ae46 891f8657 8eebe90e f7a1202e

Count      = c675a64b

Bearer     = 0c

Direction  = 1

Length     = 1245 bits

Plaintext  = 8daa17b1 ae050529 c6827f28 c0ef6a12 42e93f8b 314fb18a 77f790ae 049fedd6

             12267fec aefc4501 74d76d9f 9aa7755a 30cd90a9 a5874bf4 8eaf70ee a3a62a25

             0a8b6bd8 d9b08b08 d64e32d1 817777fb 544d49cd 49720e21 9dbf8bbe d33904e1

             fd40a41d 370a1f65 74509568 7d47ba1d 36d2349e 23f64439 2c8ea9c4 9d40c132

             71aff264 d0f24841 d6465f09 96ff84e6 5fc517c5 3efc3363 c38492a8


Counter block T1   = c675a64b 64000000 00000000 00000000

Keystream block 1  = 1c369b82 78628c59 fb87dfff 0e6dc8bc

Counter block T2   = c675a64b 64000000 00000000 00000001

Keystream block 2  = eea7d8e7 3e0211da 44a91a2a e3169673

Counter block T3   = c675a64b 64000000 00000000 00000002

Keystream block 3  = cd094951 ffc2780d f1afaa3f 665736ba

Counter block T4   = c675a64b 64000000 00000000 00000003

Keystream block 4  = 0a6e3336 1f2a36e1 30a83f44 fe3603d2

Counter block T5   = c675a64b 64000000 00000000 00000004

Keystream block 5  = 153f3c6e 9e33cc1c 66afbdc0 febd679c

Counter block T6   = c675a64b 64000000 00000000 00000005

Keystream block 6  = 2d0840a1 c52d3c4a 356982e0 61a53ad7

Counter block T7   = c675a64b 64000000 00000000 00000006
```

```
Keystream block 7  = 3264f90b 15a0e1f7 6b25f3ac 8891feef

Counter block T8   = c675a64b 64000000 00000000 00000007

Keystream block 8  = c72e3a58 a72bf62a 65fadfe6 7f49e86f

Counter block T9   = c675a64b 64000000 00000000 00000008

Keystream block 9  = 5650cdf1 b2c13995 4d522303 627993f9

Counter block T10  = c675a64b 64000000 00000000 00000009

Keystream block 10 = 7d081374 f517153b e1bafb97 3f9dd804


Ciphertext = 919c8c33 d6678970 3d05a0d7 ce82a2ae ac4ee76c 0f4da050 335e8a84 e7897ba5

             df2f36bd 513e3d0c 8578c7a0 fcf043e0 3aa3a39f baad7d15 be074faa 5d9029f7

             1fb457b6 47834714 b0e18f11 7fca1067 7945096c 8c5f326b a8d6095e b29c3e36

             cf245d16 22aafe92 1f7566c4 f5d644f2 f1fc0ec6 84ddb213 49747622 e209295d

             27ff3f95 623371d4 9b147c0a f486171f 22cd04b1 cbeb2658 223e6938
```

C.1.6 Test Set 6

```
Key        = 54f4e2e0 4c83786e ec8fb5ab e8e36566

Count      = aca4f50f

Bearer     = 0b

Direction  = 0

Length     = 3861 bits

Plaintext  = 40981ba6 824c1bfb 4286b299 783daf44 2c099f7a b0f58d5c 8e46b104 f08f01b4

             1ab48547 2029b71d 36bd1a3d 90dc3a41 b46d5167 2ac4c966 3a2be063 da4bc8d2

             808ce33e 2cccbfc6 34e1b259 060876a0 fbb5a437 ebcc8d31 c19e4454 318745e3

             fa16bb11 adae2488 79fe52db 2543e53c f445d3d8 28ce0bf5 c560593d 97278a59

             762dd0c2 c9cd68d4 496a7925 08614014 b13b6aa5 1128c18c d6a90b87 978c2ff1

             cabe7d9f 898a411b fdb84f68 f6727b14 99cdd30d f0443ab4 a6665333 0bcba110

             5e4cec03 4c73e605 b4310eaa adcfd5b0 ca27ffd8 9d144df4 79275942 7c9cc1f8

             cd8c8720 2364b8a6 87954cb0 5a8d4e2d 99e73db1 60deb180 ad0841e9 6741a5d5

             9fe4189f 15420026 fe4cd121 04932fb3 8f735340 438aaf7e ca6fd5cf d3a195ce

             5abe6527 2af607ad a1be65a6 b4c9c069 3234092c 4d018f17 56c6db9d c8a6d80b

             88813861 6b681262 f954d0e7 71174878 0d92291d 86299972 db741cfa 4f37b8b5

             6cdb18a7 ca8218e8 6e4b4b71 6a4d0437 1fbec262 fc5ad0b3 819b187b 97e55b1a

             4d7c19ee 24c8b4d7 723cfedf 045b8aca e4869517 d80e5061 5d9035d5 d9c5a40a

             f602280b 542597b0 cb18619e eb359257 59d195e1 00e8e4aa 0c38a3c2 abe0f3d8

             ff04f3c3 3c295069 c23694b5 bbeacdd5 42e28e8a 94edb911 9f412d05 4be1fa72

             00b09000


Counter block T1   = aca4f50f 58000000 00000000 00000000

Keystream block 1  = 1c2f37c8 5ecb94ee 2467b0ca d7fecb8d

Counter block T2   = aca4f50f 58000000 00000000 00000001
```

```
Keystream block 2  = d65d92eb fd4cc1e2 6c336195 8c29aeb9

Counter block T3   = aca4f50f 58000000 00000000 00000002

Keystream block 3  = 6d1831a8 1b97ad6f 1d93ef80 8d97b46b

Counter block T4   = aca4f50f 58000000 00000000 00000003

Keystream block 4  = 116f1fa6 124ee978 41e59943 748ddd5b

Counter block T5   = aca4f50f 58000000 00000000 00000004

Keystream block 5  = dffad96b 48107b02 b6435c44 8df6bae4

Counter block T6   = aca4f50f 58000000 00000000 00000005

Keystream block 6  = 63590c08 50b9749a 929049fb 8f596a46

Counter block T7   = aca4f50f 58000000 00000000 00000006

Keystream block 7  = 734d3988 b6cc534d 501ea089 b83c9c5c

Counter block T8   = aca4f50f 58000000 00000000 00000007

Keystream block 8  = 9facb4de 01a3e60f 58144b8b 81b206ec

Counter block T9   = aca4f50f 58000000 00000000 00000008

Keystream block 9  = 15eba802 e1e8abd9 43840ee1 c9279262

Counter block T10  = aca4f50f 58000000 00000000 00000009

Keystream block 10 = e52928bf 91a5d242 1eb062cb e22178df

Counter block T11  = aca4f50f 58000000 00000000 0000000a

Keystream block 11 = 5129400b 020be828 8183657f ef5c59d6

Counter block T12  = aca4f50f 58000000 00000000 0000000b

Keystream block 12 = 9f52addc e66ecef8 78ce4453 3dae4917

Counter block T13  = aca4f50f 58000000 00000000 0000000c

Keystream block 13 = 900c24e3 91ee8591 685f3fbf 922e40ec

Counter block T14  = aca4f50f 58000000 00000000 0000000d

Keystream block 14 = 8d884ac7 bb03a3f8 271cd7b3 d1e9b515

Counter block T15  = aca4f50f 58000000 00000000 0000000e

Keystream block 15 = f9b25b07 60a82c6f 1774bd4d 7ccf1dec

Counter block T16  = aca4f50f 58000000 00000000 0000000f

Keystream block 16 = e1399a88 a0604f6b 6097da9f b3ddb5c0

Counter block T17  = aca4f50f 58000000 00000000 00000010

Keystream block 17 = 561ad7cf f0798b74 fa971c1f e91517e6

Counter block T18  = aca4f50f 58000000 00000000 00000011

Keystream block 18 = 55cf8f89 08bb4c66 c87abd4a 8f2a0b9c

Counter block T19  = aca4f50f 58000000 00000000 00000012

Keystream block 19 = f33ff05d 3bde2054 d904f3a9 a08e5172

Counter block T20  = aca4f50f 58000000 00000000 00000013

Keystream block 20 = 034f5c3d b6cdf0a6 6c078846 bc83c91c

Counter block T21  = aca4f50f 58000000 00000000 00000014

Keystream block 21 = 6c0726d8 8353ed9d 3dbfa7b2 2687709d

Counter block T22  = aca4f50f 58000000 00000000 00000015
```

```
Keystream block 22 = 74b698ea 0d1783ab d0df36fd c82cca6e

Counter block T23  = aca4f50f 58000000 00000000 00000016

Keystream block 23 = 32348e64 fe86518e b5477cbb 97578dd2

Counter block T24  = aca4f50f 58000000 00000000 00000017

Keystream block 24 = 7bd4f7e2 173eb542 a047f1b0 1f3d008c

Counter block T25  = aca4f50f 58000000 00000000 00000018

Keystream block 25 = 825fd522 f0e0b3b0 ccd4106d 39ddd88c

Counter block T26  = aca4f50f 58000000 00000000 00000019

Keystream block 26 = f930dc26 db0e6bce d465d457 b82fe7c2

Counter block T27  = aca4f50f 58000000 00000000 0000001a

Keystream block 27 = bc90c3f4 abc1072d 0f74300c 13106527

Counter block T28  = aca4f50f 58000000 00000000 0000001b

Keystream block 28 = 39da03e3 c5bf5152 b809045f ee778e01

Counter block T29  = aca4f50f 58000000 00000000 0000001c

Keystream block 29 = 3b1f75fe 95c81280 c2165b65 cf3c5fae

Counter block T30  = aca4f50f 58000000 00000000 0000001d

Keystream block 30 = 385138f8 c9f7d62e 07f8e4df e379d08d

Counter block T31  = aca4f50f 58000000 00000000 0000001e

Keystream block 31 = 06c8b899 06c71bb9 2e834ee7 e81cd109


Ciphertext = 5cb72c6e dc878f15 66e10253 afc364c9 fa540d91 4db94cbe e275d091 7ca6af0d

             77acb4ef 3bbe1a72 2b2ef5bd 1d4b8e2a a5024ec1 388a201e 7bce7920 aec61589

             5f763a55 64dcc4c4 82a2ee1d 8bfecc44 98eca83f bb75f9ab 530e0daf bede2fa5

             895b8299 1b6277c5 29e0f252 9d7f7960 6be96706 296dedfa 9d7412b6 16958cb5

             63c678c0 2825c30d 0aee77c4 c146d276 5412421a 808d13ce c819694c 75ad572e

             9b973d94 8b81a933 7c3b2a17 192e22c2 069f7ed1 162af44c dea81760 3665e807

             ce40c8e0 dd9d6394 dc6e3115 3fe1955c 47afb51f 2617ee0c 5e3b8ef1 ad7574ed

             343edc27 43cc94c9 90e1f1fd 264253c1 78dea739 c0befeeb cd9f9b76 d49c1015

             c9fecf50 e53b8b52 04dbcd3e ed863855 dabcdcc9 4b31e318 02156885 5c8b9e52

             a981957a 112827f9 78ba960f 1447911b 317b5511 fbcc7fb1 3ac153db 74251117

             e4861eb9 e83bffff c4eb7755 579038e5 7924b1f7 8b3e1ad9 0bab2a07 871b72db

             5eef96c3 34044966 db0c37ca fd1a89e5 646a3580 eb6465f1 21dce9cb 88d85b96

             cf23cccc d4280767 bee8eeb2 3d865246 1db64931 03003baf 89f5e182 61ea43c8

             4a92ebff ffe4909d c46c5192 f825f770 600b9602 c557b5f8 b431a79d 45977dd9

             c41b863d a9e142e9 0020cfd0 74d6927b 7ab3b672 5d1a6f3f 98b9c9da a8982aff

             06782800
```

C.2 128-EIA2

This section includes six test data sets; all are presented in hex, while the first is also presented in binary. Many intermediate computational values are included to assist implementers in tracing bugs. Some notation is taken from the specification of CMAC mode [17].

Bit ordering should be largely self explanatory, but in particular:

- The 5-bit BEARER is written in hex in a 'right aligned' form, i.e. as a two-hex-digit value in the range 00 to 1F inclusive, with BEARER [0] as the msb of the first digit.
- Similarly the single DIRECTION bit is written in hex in 'right aligned' form, i.e. the DIRECTION bit is the lsb of the hex digit.
- Where the length of the message, or of a message sub-block, is not a multiple of 32 bits, it is written in hex in a 'left aligned' form, i.e. the least significant few bits of the last word will be zero.

C.2.1 Test Set 1

```
Count-I   = (hex) 38a6f056

Count-I   = (bin) 00111000 10100110 11110000 01010110

Bearer    = (hex) 18

Bearer    = (bin) 11000

Direction = (hex) 0

Direction = (bin) 0

IK        = (hex) 2bd6459f 82c5b300 952c4910 4881ff48

IK        = (bin) 00101011 11010110 01000101 10011111 10000010 11000101 10110011 00000000

                  10010101 00101100 01001001 00010000 01001000 10000001 11111111 01001000

Length    = 58 bits

Message   = (hex) 33323462 63393840

Message   = (bin) 00110011 00110010 00110100 01100010 01100011 00111001 00111000 01


CMAC(K, M):

K         = (hex) 2bd6459f 82c5b300 952c4910 4881ff48

K         = (bin) 00101011 11010110 01000101 10011111 10000010 11000101 10110011 00000000

                  10010101 00101100 01001001 00010000 01001000 10000001 11111111 01001000

Mlen      = 122

M         = (hex) 38a6f056 c0000000 33323462 63393840

M         = (bin) 00111000 10100110 11110000 01010110 11000000 00000000 00000000 00000000

                  00110011 00110010 00110100 01100010 01100011 00111001 00111000 01
```

Subkey Generation:

```
L         = (hex) 6e426138 5adfc1fc b7c85f0c 469fb20c

L         = (bin) 01101110 01000010 01100001 00111000 01011010 11011111 11000001 11111100

                  10110111 11001000 01011111 00001100 01000110 10011111 10110010 00001100

K1        = (hex) dc84c270 b5bf83f9 6f90be18 8d3f6418
```

```
K1         = (bin) 11011100 10000100 11000010 01110000 10110101 10111111 10000011 11111001
                   01101111 10010000 10111110 00011000 10001101 00111111 01100100 00011000
K2         = (hex) b90984e1 6b7f07f2 df217c31 1a7ec8b7
K2         = (bin) 10111001 00001001 10000100 11100001 01101011 01111111 00000111 11110010
                   11011111 00100001 01111100 00110001 00011010 01111110 11001000 10110111
```

MAC Generation:

```
n          = 1
Mn*        = (hex) 38a6f056 c0000000 33323462 63393840
Mn*        = (bin) 00111000 10100110 11110000 01010110 11000000 00000000 00000000 00000000
                   00110011 00110010 00110100 01100010 01100011 00111001 00111000 01
Mn         = (hex) 81af74b7 ab7f07f2 ec134853 7947f0d7
Mn         = (bin) 10000001 10101111 01110100 10110111 10101011 01111111 00000111 11110010
                   11101100 00010011 01001000 01010011 01111001 01000111 11110000 11010111
C0         = (hex) 00000000 00000000 00000000 00000000
C0         = (bin) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
M1         = (hex) 81af74b7 ab7f07f2 ec134853 7947f0d7
M1         = (bin) 10000001 10101111 01110100 10110111 10101011 01111111 00000111 11110010
                   11101100 00010011 01001000 01010011 01111001 01000111 11110000 11010111
C1         = (hex) 118c6eb8 b775144b 0b831110 54c96eb6
C1         = (bin) 00010001 10001100 01101110 10111000 10110111 01110101 00010100 01001011
                   00001011 10000011 00010001 00010000 01010100 11001001 01101110 10110110

MACT       = (hex) 118c6eb8
MACT       = (bin) 00010001 10001100 01101110 10111000
```

C.2.2 Test Set 2

```
Count-I   = 398a59b4
Bearer    = 1a
Direction = 1
IK        = d3c5d592 327fb11c 4035c668 0af8c6d1
Length    = 64 bits
Message   = 484583d5 afe082ae


CMAC(K, M):
K         = d3c5d592 327fb11c 4035c668 0af8c6d1
Mlen      = 128
M         = 398a59b4 d4000000 484583d5 afe082ae
```

Subkey Generation:

```
L         = 9b71f299 132915d3 605211b5 e5df8632
K1        = 36e3e532 26522ba6 c0a4236b cbbf0ce3
K2        = 6dc7ca64 4ca4574d 814846d7 977e19c6
```

MAC Generation:

```
n         = 1
Mn*       = 398a59b4 d4000000 484583d5 afe082ae
Mn        = 0f69bc86 f2522ba6 88e1a0be 645f8e4d
C0        = 00000000 00000000 00000000 00000000
M1        = 0f69bc86 f2522ba6 88e1a0be 645f8e4d
C1        = b93787e6 493ff113 ad73d3e0 1e826d73

MACT      = b93787e6
```

C.2.3 Test Set 3

```
Count-I   = 36af6144
Bearer    = 18
Direction = 1
IK        = 7e5e9443 1e11d738 28d739cc 6ced4573
Length    = 254 bits
Message   = b3d3c917 0a4e1632 f60f8610 13d22d84 b726b6a2 78d802d1 eeaf1321 ba5929dc

CMAC(K, M):
K         = 7e5e9443 1e11d738 28d739cc 6ced4573
Mlen      = 318
M         = 36af6144 c4000000 b3d3c917 0a4e1632 f60f8610 13d22d84 b726b6a2 78d802d1
            eeaf1321 ba5929dc
```

Subkey Generation:

```
L         = d78b4628 35781e79 d2255f8d 309a60ef
K1        = af168c50 6af03cf3 a44abf1a 6134c159
K2        = 5e2d18a0 d5e079e7 48957e34 c2698235
```

MAC Generation:

```
n         = 3
Mn*       = eeaf1321 ba5929dc
Mn        = b0820b81 6fb95039 48957e34 c2698235
```

```
C0        = 00000000 00000000 00000000 00000000

M1        = 36af6144 c4000000 b3d3c917 0a4e1632

C1        = 3bb0e1d8 2cb96273 64a7cfd3 a52eed15

M2        = f60f8610 13d22d84 b726b6a2 78d802d1

C2        = e3a6446d fae7f10f e3e3320d a8e49955

M3        = b0820b81 6fb95039 48957e34 c2698235

C3        = 1f60b01d e05aa666 3bda32c6 1771e70b


MACT      = 1f60b01d
```

C.2.4 Test Set 4

```
Count-I   = c7590ea9

Bearer    = 17

Direction = 0

IK        = d3419be8 21087acd 02123a92 48033359

Length    = 511 bits

Message   = bbb05703 8809496b cff86d6f bc8ce5b1 35a06b16 6054f2d5 65be8ace 75dc851e

            0bcdd8f0 7141c495 872fb5d8 c0c66a8b 6da55666 3e4e4612 05d84580 bee5bc7e


CMAC(K, M):

K         = d3419be8 21087acd 02123a92 48033359

Mlen      = 575

M         = c7590ea9 b8000000 bbb05703 8809496b cff86d6f bc8ce5b1 35a06b16 6054f2d5

            65be8ace 75dc851e 0bcdd8f0 7141c495 872fb5d8 c0c66a8b 6da55666 3e4e4612

            05d84580 bee5bc7e
```

Subkey Generation:

```
L         = 054dd008 2d9ecd21 a3f32b0a a7369be4

K1        = 0a9ba010 5b3d9a43 47e65615 4e6d37c8

K2        = 15374020 b67b3486 8fccac2a 9cda6f90
```

MAC Generation:

```
n         = 5

Mn*       = 05d84580 bee5bc7e

Mn        = 10ef05a0 089e88f9 8fccac2a 9cda6f90

C0        = 00000000 00000000 00000000 00000000

M1        = c7590ea9 b8000000 bbb05703 8809496b

C1        = cb36ed77 e49bd772 ac410f25 eea31084

M2        = cff86d6f bc8ce5b1 35a06b16 6054f2d5
```

```
C2        = e44baf91 d48ba92c 542f3b14 a8a496d9

M3        = 65be8ace 75dc851e 0bcdd8f0 7141c495

C3        = c3542869 eed00692 e3b4ef1a 6b324aaf

M4        = 872fb5d8 c0c66a8b 6da55666 3e4e4612

C4        = 5054d998 92675b0f 989d3b0f 3c043c4e

M5        = 10ef05a0 089e88f9 8fccac2a 9cda6f90

C5        = 6846a2f0 a0b6be7a 4fb26a15 7e914c53


MACT      = 6846a2f0
```

C.2.5 Test Set 5

```
Count-I   = 36af6144

Bearer    = 0f

Direction = 1

IK        = 83fd23a2 44a74cf3 58da3019 f1722635

Length    = 768 bits

Message   = 35c68716 633c66fb 750c2668 65d53c11 ea05b1e9 fa49c839 8d48e1ef a5909d39

            47902837 f5ae96d5 a05bc8d6 1ca8dbef 1b13a4b4 abfe4fb1 006045b6 74bb5472

            9304c382 be53a5af 05556176 f6eaa2ef 1d05e4b0 83181ee6 74cda5a4 85f74d7a


CMAC(K, M):

K         = 83fd23a2 44a74cf3 58da3019 f1722635

Mlen      = 832

M         = 36af6144 7c000000 35c68716 633c66fb 750c2668 65d53c11 ea05b1e9 fa49c839

            8d48e1ef a5909d39 47902837 f5ae96d5 a05bc8d6 1ca8dbef 1b13a4b4 abfe4fb1

            006045b6 74bb5472 9304c382 be53a5af 05556176 f6eaa2ef 1d05e4b0 83181ee6

            74cda5a4 85f74d7a
```

Subkey Generation:

```
L         = 9df61c57 3c86acac 704db9d5 b0dea444

K1        = 3bec38ae 790d5958 e09b73ab 61bd480f

K2        = 77d8715c f21ab2b1 c136e756 c37a901e
```

MAC Generation:

```
n         = 7

Mn*       = 74cda5a4 85f74d7a

Mn        = 0315d4f8 77edffcb 4136e756 c37a901e

C0        = 00000000 00000000 00000000 00000000

M1        = 36af6144 7c000000 35c68716 633c66fb
```

```
C1        = 57c5a916 e19d7747 c2a69283 5eed0015

M2        = 750c2668 65d53c11 ea05b1e9 fa49c839

C2        = 7937651c b2c34e23 646b4396 f77bca0d

M3        = 8d48e1ef a5909d39 47902837 f5ae96d5

C3        = dfa3c570 d7b4dd08 2533b643 f82f646c

M4        = a05bc8d6 1ca8dbef 1b13a4b4 abfe4fb1

C4        = 7a8e64c0 eb34df52 e4236368 0f019ddd

M5        = 006045b6 74bb5472 9304c382 be53a5af

C5        = 3f5f08a2 5a6a8ba8 9a5dd816 626a26ef

M6        = 05556176 f6eaa2ef 1d05e4b0 83181ee6

C6        = 9fe7991a 50c5f542 e0bf0da0 9dec1456

M7        = 0315d4f8 77edffcb 4136e756 c37a901e

C7        = e657e182 5298f2fa ee2ca1e0 7373bc7e


MACT      = e657e182
```

C.2.6 Test Set 6

```
Count-I   = 36af6144

Bearer    = 18

Direction = 0

IK        = 6832a65c ff447362 1ebdd4ba 26a921fe

Length    = 383 bits

Message   = d3c53839 62682071 77656676 20323837 63624098 1ba6824c 1bfb1ab4 85472029

            b71d808c e33e2cc3 c0b5fc1f 3de8a6dc


CMAC(K, M):

K         = 6832a65c ff447362 1ebdd4ba 26a921fe

Mlen      = 447

M         = 36af6144 c0000000 d3c53839 62682071 77656676 20323837 63624098 1ba6824c

            1bfb1ab4 85472029 b71d808c e33e2cc3 c0b5fc1f 3de8a6dc
```

Subkey Generation:

```
L         = e50123c3 87e13fd6 8d8bf0d0 a4581685

K1        = ca024787 0fc27fad 1b17e1a1 48b02d8d

K2        = 94048f0e 1f84ff5a 362fc342 91605b9d
```

MAC Generation:

```
n         = 4

Mn*       = c0b5fc1f 3de8a6dc
```

```
Mn        = 54b17311 226c5987 362fc342 91605b9d

C0        = 00000000 00000000 00000000 00000000

M1        = 36af6144 c0000000 d3c53839 62682071

C1        = 263dd98f beccb69a 428e92d4 21fbed9e

M2        = 77656676 20323837 63624098 1ba6824c

C2        = 1838cb78 cb2d32dc ec486c79 d9007a19

M3        = 1bfb1ab4 85472029 b71d808c e33e2cc3

C3        = 5ebf1009 f663be7b 68373072 4c20271f

M4        = 54b17311 226c5987 362fc342 91605b9d

C4        = f0668c1e 4197300b 1243f834 25d06c25


MACT      = f0668c1e
```

C.2.7 Test Set 7

```
Count-I   = 7827fab2

Bearer    = 05

Direction = 1

IK        = 5d0a80d8 134ae196 77824b67 1e838af4

Length    = 2558 bits

Message   = 70dedf2d c42c5cbd 3a96f8a0 b11418b3 608d5733 604a2cd3 6aabc70c e3193bb5

            153be2d3 c06dfdb2 d16e9c35 7158be6a 41d6b861 e491db3f bfeb518e fcf048d7

            d5895373 0ff30c9e c470ffcd 663dc342 01c36add c0111c35 b38afee7 cfdb582e

            3731f8b4 baa8d1a8 9c06e811 99a97162 27be344e fcb436dd d0f096c0 64c3b5e2

            c399993f c77394f9 e09720a8 11850ef2 3b2ee05d 9e617360 9d86e1c0 c18ea51a

            012a00bb 413b9cb8 188a703c d6bae31c c67b34b1 b00019e6 a2b2a690 f02671fe

            7c9ef8de c0094e53 3763478d 58d2c5f5 b827a014 8c5948a9 6931acf8 4f465a64

            e62ce740 07e991e3 7ea823fa 0fb21923 b79905b7 33b631e6 c7d6860a 3831ac35

            1a9c730c 52ff72d9 d308eedb ab21fde1 43a0ea17 e23edc1f 74cbb363 8a2033aa

            a15464ea a733385d bbeb6fd7 3509b857 e6a419dc a1d8907a f977fbac 4dfa35ec


CMAC(K, M):

K         = 5d0a80d8 134ae196 77824b67 1e838af4

Mlen      = 2622

M         = 7827fab2 2c000000 70dedf2d c42c5cbd 3a96f8a0 b11418b3 608d5733 604a2cd3

            6aabc70c e3193bb5 153be2d3 c06dfdb2 d16e9c35 7158be6a 41d6b861 e491db3f

            bfeb518e fcf048d7 d5895373 0ff30c9e c470ffcd 663dc342 01c36add c0111c35

            b38afee7 cfdb582e 3731f8b4 baa8d1a8 9c06e811 99a97162 27be344e fcb436dd

            d0f096c0 64c3b5e2 c399993f c77394f9 e09720a8 11850ef2 3b2ee05d 9e617360

            9d86e1c0 c18ea51a 012a00bb 413b9cb8 188a703c d6bae31c c67b34b1 b00019e6

            a2b2a690 f02671fe 7c9ef8de c0094e53 3763478d 58d2c5f5 b827a014 8c5948a9
```

```
6931acf8 4f465a64 e62ce740 07e991e3 7ea823fa 0fb21923 b79905b7 33b631e6
c7d6860a 3831ac35 1a9c730c 52ff72d9 d308eedb ab21fde1 43a0ea17 e23edc1f
74cbb363 8a2033aa a15464ea a733385d bbeb6fd7 3509b857 e6a419dc a1d8907a
f977fbac 4dfa35ec
```

Subkey Generation:

```
L          = 9832e229 fbb93970 bcf7b282 3ee4fe5d
K1         = 3065c453 f77272e1 79ef6504 7dc9fc3d
K2         = 60cb88a7 eee4e5c2 f3deca08 fb93f87a
```

MAC Generation:

```
n          = 21
Mn*        = f977fbac 4dfa35ec
Mn         = 99bc730b a31ed02c f3deca08 fb93f87a
C0         = 00000000 00000000 00000000 00000000
M1         = 7827fab2 2c000000 70dedf2d c42c5cbd
C1         = 6c9b07c0 35b7a016 3aad1405 1f57f3e0
M2         = 3a96f8a0 b11418b3 608d5733 604a2cd3
C2         = ec9c6b75 1d027216 3412fad4 f01cebba
M3         = 6aabc70c e3193bb5 153be2d3 c06dfdb2
C3         = 3c83db67 ff87c86b 57ae4742 42c9816b
M4         = d16e9c35 7158be6a 41d6b861 e491db3f
C4         = e6e894ee 7e148494 44afcb75 9752e555
M5         = bfeb518e fcf048d7 d5895373 0ff30c9e
C5         = cbf27df1 0fd514f0 489dd303 d2dbee51
M6         = c470ffcd 663dc342 01c36add c0111c35
C6         = 6989143a 39de09ab 2680fe6c 41f0a7c1
M7         = b38afee7 cfdb582e 3731f8b4 baa8d1a8
C7         = fe4049fa 655ee010 49299c58 c91024ff
M8         = 9c06e811 99a97162 27be344e fcb436dd
C8         = 1e9dab32 48d5ee47 c7e3a420 6f18b17b
M9         = d0f096c0 64c3b5e2 c399993f c77394f9
C9         = 9da578a5 00a0c7f1 e825a4ca 71557055
M10        = e09720a8 11850ef2 3b2ee05d 9e617360
C10        = 4141c882 a23da353 2b11642a 85fea2bf
M11        = 9d86e1c0 c18ea51a 012a00bb 413b9cb8
C11        = 18467572 0bdfcb5b 6bb71899 a6cafcc7
M12        = 188a703c d6bae31c c67b34b1 b00019e6
C12        = 156a70e5 af77f9a4 74d08303 e8c0412a
```

```
M13       = a2b2a690 f02671fe 7c9ef8de c0094e53

C13       = dba504a1 26fa047f 8b8c295f 73e90a5c

M14       = 3763478d 58d2c5f5 b827a014 8c5948a9

C14       = ab1a2703 3472acc8 e36c221b b7a0e530

M15       = 6931acf8 4f465a64 e62ce740 07e991e3

C15       = 04ceffcd e7618885 43c7e837 0f3bce6d

M16       = 7ea823fa 0fb21923 b79905b7 33b631e6

C16       = 215ec3bf 5f3a303e 53db5269 e6c99fc2

M17       = c7d6860a 3831ac35 1a9c730c 52ff72d9

C17       = 8622e51b 45a660f3 d98fcf74 e5cc36b3

M18       = d308eedb ab21fde1 43a0ea17 e23edc1f

C18       = 6e998fa6 196d5a4c 1ded2973 c09c0f8c

M19       = 74cbb363 8a2033aa a15464ea a733385d

C19       = 1710bc91 22e54289 244a87ce 23438f41

M20       = bbeb6fd7 3509b857 e6a419dc a1d8907a

C20       = 3e18b029 a8ef18da b9968614 96552fd7

M21       = 99bc730b a31ed02c f3deca08 fb93f87a

C21       = f4cc8fa3 59e6e2e7 6e09c45d 6ea5e0de


MACT      = f4cc8fa3
```

C.2.8 Test Set 8

```
Count-I   = 296f393c

Bearer    = 0b

Direction = 1

IK        = b3120ffd b2cf6af4 e73eaf2e f4ebec69

Length    = 16448 bits

Message   = 00000000 00000000 01010101 01010101 e0958045 f3a0bba4 e3968346 f0a3b8a7

            c02a018a e6407652 26b987c9 13e6cbf0 83570016 cf83efbc 61c08251 3e21561a

            427c009d 28c298ef ace78ed6 d56c2d45 05ad032e 9c04dc60 e73a8169 6da665c6

            c48603a5 7b45ab33 221585e6 8ee31691 87fb0239 528632dd 656c807e a3248b7b

            46d002b2 b5c7458e b85b9ce9 5879e034 0859055e 3b0abbc3 eace8719 caa80265

            c97205d5 dc4bcc90 2fe18396 29ed7132 8a0f0449 f588557e 6898860e 042aecd8

            4b2404c2 12c9222d a5bf8a89 ef679787 0cf50771 a60f66a2 ee628536 57addf04

            cdde07fa 414e11f1 2b4d81b9 b4e8ac53 8ea30666 688d881f 6c348421 992f31b9

            4f8806ed 8fccff4c 9123b896 42527ad6 13b109bf 75167485 f1268bf8 84b4cd23

            d29a0934 925703d6 34098f77 67f1be74 91e708a8 bb949a38 73708aef 4a36239e

            50cc0823 5cd5ed6b be578668 a17b58c1 171d0b90 e813a9e4 f58a89d7 19b11042

            d6360b1b 0f52deb7 30a58d58 faf46315 954b0a87 26914759 77dc88c0 d733feff

            54600a0c c1d0300a aaeb9457 2c6e95b0 1ae90de0 4f1dce47 f87e8fa7 bebf77e1
```

dbc20d6b a85cb914 3d518b28 5dfa04b6 98bf0cf7 819f20fa 7a288eb0 703d995c

59940c7c 66de57a9 b70f8237 9b70e203 1e450fcf d2181326 fcd28d88 23baaa80

df6e0f44 35596475 39fd8907 c0ffd9d7 9c130ed8 1c9afd9b 7e848c9f ed38443d

5d380e53 fbdb8ac8 c3d3f068 76054f12 2461107d e92fea09 c6f6923a 188d53af

e54a10f6 0e6e9d5a 03d996b5 fbc820f8 a637116a 27ad04b4 44a0932d d60fbd12

671c11e1 c0ec73e7 89879faa 3d42c64d 20cd1252 742a3768 c25a9015 85888ece

e1e612d9 936b403b 0775949a 66cdfd99 a29b1345 baa8d9d5 400c9102 4b0a6073

63b013ce 5de9ae86 9d3b8d95 b0570b3c 2d391422 d32450cb cfae9665 2286e96d

ec1214a9 34652798 0a8192ea c1c39a3a af6f1535 1da6be76 4df89772 ec0407d0

6e4415be fae7c925 80df9bf5 07497c8f 2995160d 4e218daa cb02944a bf83340c

e8be1686 a960faf9 0e2d90c5 5cc6475b abc3171a 80a36317 4954955d 7101dab1

6ae81791 67e21444 b443a9ea aa7c91de 36d118c3 9d389f8d d4469a84 6c9a262b

f7fa1848 7a79e8de 11699e0b 8fdf557c b48719d4 53ba7130 56109b93 a218c896

75ac195f b4fb0663 9b379714 4955b3c9 327d1aec 003d42ec d0ea98ab f19ffb4a

f3561a67 e77c35bf 15c59c24 12da881d b02b1bfb cebfac51 52bc99bc 3f1d15f7

71001b70 29fedb02 8f8b852b c4407eb8 3f891c9c a733254f dd1e9edb 56919ce9

fea21c17 4072521c 18319a54 b5d4efbe bddf1d8b 69b1cbf2 5f489fcc 98137254

7cf41d00 8ef0bca1 926f934b 735e090b 3b251eb3 3a36f82e d9b29cf4 cb944188

fa0e1e38 dd778f7d 1c9d987b 28d132df b9731fa4 f4b41693 5be49de3 0516af35

78581f2f 13f561c0 66336194 1eab249a 4bc123f8 d15cd711 a956a1bf 20fe6eb7

8aea2373 361da042 6c79a530 c3bb1de0 c99722ef 1fde39ac 2b00a0a8 ee7c800a

08bc2264 f89f4eff e627ac2f 0531fb55 4f6d21d7 4c590a70 adfaa390 bdfbb3d6

8e46215c ab187d23 68d5a71f 5ebec081 cd3b20c0 82dbe4cd 2faca287 73795d6b

0c10204b 659a939e f29bbe10 88243624 429927a7 eb576dd3 a00ea5e0 1af5d475

83b2272c 0c161a80 6521a16f f9b0a722 c0cf26b0 25d5836e 2258a4f7 d4773ac8

01e4263b c294f43d ef7fa870 3f3a4197 46352588 7652b0b2 a4a2a7cf 87f00914

871e2503 9113c7e1 618da340 64b57a43 c463249f b8d05e0f 26f4a6d8 4972e7a9

05482414 5f91295c dbe39a6f 920facc6 59712b46 a54ba295 bbe6a901 54e91b33

985a2bcd 420ad5c6 7ec9ad8e b7ac6864 db272a51 6bc94c28 39b0a816 9a6bf58e

1a0c2ada 8c883b7b f497a491 71268ed1 5ddd2969 384e7ff4 bf4aab2e c9ecc652

9cf629e2 df0f08a7 7a65afa1 2aa9b505 df8b287e f6cc9149 3d1caa39 076e28ef

1ea028f5 118de61a e02bb6ae fc3343a0 50292f19 9f401857 b2bead5e 6ee2a1f1

91022f92 78016f04 7791a9d1 8da7d2a6 d27f2e0e 51c2f6ea 30e8ac49 a0604f4c

13542e85 b68381b9 fdcfa0ce 4b2d3413 54852d36 0245c536 b612af71 f3e77c90

95ae2dbd e504b265 733dabfe 10a20fc7 d6d32c21 ccc72b8b 3444ae66 3d65922d

17f82caa 2b865cd8 8913d291 a6589902 6ea13284 39723c19 8c36b0c3 c8d085bf

af8a320f de334b4a 4919b44c 2b95f6e8 ecf73393 f7f0d2a4 0e60b1d4 06526b02

2ddc3318 10b1a5f7 c347bd53 ed1f105d 6a0d30ab a477e178 889ab2ec 55d558de

ab263020 4336962b 4db5b663 b6902b89 e85b31bc 6af50fc5 0accb3fb 9b57b663

29703137 8db47896 d7fbaf6c 600add2c 67f936db 037986db 856eb49c f2db3f7d

```
            a6d23650 e438f188 4041b013 119e4c2a e5af37cc cdfb6866 0738b58b 3c59d1c0
            24843747 2aba1f35 ca1fb90c d714aa9f 635534f4 9e7c5bba 81c2b6b3 6fdee21c
            a27e347f 793d2ce9 44edb23c 8c9b914b e10335e3 50feb507 0394b7a4 a15c0ca1
            20283568 b7bfc254 fe838b13 7a2147ce 7c113a3a 4d65499d 9e86b87d bcc7f03b
            bd3a3ab1 aa243ece 5ba9bcf2 5f82836c fe473b2d 83e7a720 1cd0b96a 72451e86
            3f6c3ba6 64a6d073 d1f7b5ed 990865d9 78bd3815 d06094fc 9a2aba52 21c22d5a
            b996389e 3721e3af 5f05bedd c2875e0d faeb3902 1ee27a41 187cbb45 ef40c3e7
            3bc03989 f9a30d12 c54ba7d2 141da8a8 75493e65 776ef35f 97debc22 86cc4af9
            b4623eee 902f840c 52f1b8ad 658939ae f71f3f72 b9ec1de2 1588bd35 484ea444
            36343ff9 5ead6ab1 d8afb1b2 a303df1b 71e53c4a ea6b2e3e 9372be0d 1bc99798
            b0ce3cc1 0d2a596d 565dba82 f88ce4cf f3b33d5d 24e9c083 1124bf1a d54b7925
            32983dd6 c3a8b7d0

CMAC(K, M):
K          = b3120ffd b2cf6af4 e73eaf2e f4ebec69
Mlen       = 16512
M          = 296f393c 5c000000 00000000 00000000 01010101 01010101 e0958045 f3a0bba4
            e3968346 f0a3b8a7 c02a018a e6407652 26b987c9 13e6cbf0 83570016 cf83efbc
            61c08251 3e21561a 427c009d 28c298ef ace78ed6 d56c2d45 05ad032e 9c04dc60
            e73a8169 6da665c6 c48603a5 7b45ab33 221585e6 8ee31691 87fb0239 528632dd
            656c807e a3248b7b 46d002b2 b5c7458e b85b9ce9 5879e034 0859055e 3b0abbc3
            eace8719 caa80265 c97205d5 dc4bcc90 2fe18396 29ed7132 8a0f0449 f588557e
            6898860e 042aecd8 4b2404c2 12c9222d a5bf8a89 ef679787 0cf50771 a60f66a2
            ee628536 57addf04 cdde07fa 414e11f1 2b4d81b9 b4e8ac53 8ea30666 688d881f
            6c348421 992f31b9 4f8806ed 8fccff4c 9123b896 42527ad6 13b109bf 75167485
            f1268bf8 84b4cd23 d29a0934 925703d6 34098f77 67f1be74 91e708a8 bb949a38
            73708aef 4a36239e 50cc0823 5cd5ed6b be578668 a17b58c1 171d0b90 e813a9e4
            f58a89d7 19b11042 d6360b1b 0f52deb7 30a58d58 faf46315 954b0a87 26914759
            77dc88c0 d733feff 54600a0c c1d0300a aaeb9457 2c6e95b0 1ae90de0 4f1dce47
            f87e8fa7 bebf77e1 dbc20d6b a85cb914 3d518b28 5dfa04b6 98bf0cf7 819f20fa
            7a288eb0 703d995c 59940c7c 66de57a9 b70f8237 9b70e203 1e450fcf d2181326
            fcd28d88 23baaa80 df6e0f44 35596475 39fd8907 c0ffd9d7 9c130ed8 1c9afd9b
            7e848c9f ed38443d 5d380e53 fbdb8ac8 c3d3f068 76054f12 2461107d e92fea09
            c6f6923a 188d53af e54a10f6 0e6e9d5a 03d996b5 fbc820f8 a637116a 27ad04b4
            44a0932d d60fbd12 671c11e1 c0ec73e7 89879faa 3d42c64d 20cd1252 742a3768
            c25a9015 85888ece e1e612d9 936b403b 0775949a 66cdfd99 a29b1345 baa8d9d5
            400c9102 4b0a6073 63b013ce 5de9ae86 9d3b8d95 b0570b3c 2d391422 d32450cb
            cfae9665 2286e96d ec1214a9 34652798 0a8192ea c1c39a3a af6f1535 1da6be76
            4df89772 ec0407d0 6e4415be fae7c925 80df9bf5 07497c8f 2995160d 4e218daa
            cb02944a bf83340c e8be1686 a960faf9 0e2d90c5 5cc6475b abc3171a 80a36317
```

```
4954955d 7101dab1 6ae81791 67e21444 b443a9ea aa7c91de 36d118c3 9d389f8d
d4469a84 6c9a262b f7fa1848 7a79e8de 11699e0b 8fdf557c b48719d4 53ba7130
56109b93 a218c896 75ac195f b4fb0663 9b379714 4955b3c9 327d1aec 003d42ec
d0ea98ab f19ffb4a f3561a67 e77c35bf 15c59c24 12da881d b02b1bfb cebfac51
52bc99bc 3f1d15f7 71001b70 29fedb02 8f8b852b c4407eb8 3f891c9c a733254f
dd1e9edb 56919ce9 fea21c17 4072521c 18319a54 b5d4efbe bddf1d8b 69b1cbf2
5f489fcc 98137254 7cf41d00 8ef0bca1 926f934b 735e090b 3b251eb3 3a36f82e
d9b29cf4 cb944188 fa0e1e38 dd778f7d 1c9d987b 28d132df b9731fa4 f4b41693
5be49de3 0516af35 78581f2f 13f561c0 66336194 1eab249a 4bc123f8 d15cd711
a956a1bf 20fe6eb7 8aea2373 361da042 6c79a530 c3bb1de0 c99722ef 1fde39ac
2b00a0a8 ee7c800a 08bc2264 f89f4eff e627ac2f 0531fb55 4f6d21d7 4c590a70
adfaa390 bdfbb3d6 8e46215c ab187d23 68d5a71f 5ebec081 cd3b20c0 82dbe4cd
2faca287 73795d6b 0c10204b 659a939e f29bbe10 88243624 429927a7 eb576dd3
a00ea5e0 1af5d475 83b2272c 0c161a80 6521a16f f9b0a722 c0cf26b0 25d5836e
2258a4f7 d4773ac8 01e4263b c294f43d ef7fa870 3f3a4197 46352588 7652b0b2
a4a2a7cf 87f00914 871e2503 9113c7e1 618da340 64b57a43 c463249f b8d05e0f
26f4a6d8 4972e7a9 05482414 5f91295c dbe39a6f 920facc6 59712b46 a54ba295
bbe6a901 54e91b33 985a2bcd 420ad5c6 7ec9ad8e b7ac6864 db272a51 6bc94c28
39b0a816 9a6bf58e 1a0c2ada 8c883b7b f497a491 71268ed1 5ddd2969 384e7ff4
bf4aab2e c9ecc652 9cf629e2 df0f08a7 7a65afa1 2aa9b505 df8b287e f6cc9149
3d1caa39 076e28ef 1ea028f5 118de61a e02bb6ae fc3343a0 50292f19 9f401857
b2bead5e 6ee2a1f1 91022f92 78016f04 7791a9d1 8da7d2a6 d27f2e0e 51c2f6ea
30e8ac49 a0604f4c 13542e85 b68381b9 fdcfa0ce 4b2d3413 54852d36 0245c536
b612af71 f3e77c90 95ae2dbd e504b265 733dabfe 10a20fc7 d6d32c21 ccc72b8b
3444ae66 3d65922d 17f82caa 2b865cd8 8913d291 a6589902 6ea13284 39723c19
8c36b0c3 c8d085bf af8a320f de334b4a 4919b44c 2b95f6e8 ecf73393 f7f0d2a4
0e60b1d4 06526b02 2ddc3318 10b1a5f7 c347bd53 ed1f105d 6a0d30ab a477e178
889ab2ec 55d558de ab263020 4336962b 4db5b663 b6902b89 e85b31bc 6af50fc5
0accb3fb 9b57b663 29703137 8db47896 d7fbaf6c 600add2c 67f936db 037986db
856eb49c f2db3f7d a6d23650 e438f188 4041b013 119e4c2a e5af37cc cdfb6866
0738b58b 3c59d1c0 24843747 2aba1f35 ca1fb90c d714aa9f 635534f4 9e7c5bba
81c2b6b3 6fdee21c a27e347f 793d2ce9 44edb23c 8c9b914b e10335e3 50feb507
0394b7a4 a15c0ca1 20283568 b7bfc254 fe838b13 7a2147ce 7c113a3a 4d65499d
9e86b87d bcc7f03b bd3a3ab1 aa243ece 5ba9bcf2 5f82836c fe473b2d 83e7a720
1cd0b96a 72451e86 3f6c3ba6 64a6d073 d1f7b5ed 990865d9 78bd3815 d06094fc
9a2aba52 21c22d5a b996389e 3721e3af 5f05bedd c2875e0d faeb3902 1ee27a41
187cbb45 ef40c3e7 3bc03989 f9a30d12 c54ba7d2 141da8a8 75493e65 776ef35f
97debc22 86cc4af9 b4623eee 902f840c 52f1b8ad 658939ae f71f3f72 b9ec1de2
1588bd35 484ea444 36343ff9 5ead6ab1 d8afb1b2 a303df1b 71e53c4a ea6b2e3e
9372be0d 1bc99798 b0ce3cc1 0d2a596d 565dba82 f88ce4cf f3b33d5d 24e9c083
```

```
            1124bf1a d54b7925 32983dd6 c3a8b7d0
```

Subkey Generation:

```
L          = 2c645dcd 72114961 d8b9c864 7aac2c5b
K1         = 58c8bb9a e42292c3 b17390c8 f55858b6
K2         = b1917735 c8452587 62e72191 eab0b16c
```

MAC Generation:

```
n          = 129
Mn*        = 1124bf1a d54b7925 32983dd6 c3a8b7d0
Mn         = 49ec0480 3169ebe6 83ebad1e 36f0ef66
C0         = 00000000 00000000 00000000 00000000
M1         = 296f393c 5c000000 00000000 00000000
C1         = 2c174eee b856df54 a2e3ce41 116181e0
M2         = 01010101 01010101 e0958045 f3a0bba4
C2         = 7a923db9 b053f844 9e706b27 378aeae0
M3         = e3968346 f0a3b8a7 c02a018a e6407652
C3         = 59d30ebc 8eb2314c 74fe3a04 1a248463
M4         = 26b987c9 13e6cbf0 83570016 cf83efbc
C4         = 78db898b 6396784c 34f8edbd e7a747c5
M5         = 61c08251 3e21561a 427c009d 28c298ef
C5         = 7c29e481 44ac6afa 3aca8a4a 7208ce99
M6         = ace78ed6 d56c2d45 05ad032e 9c04dc60
C6         = 7220fde3 3a769298 c9406349 6ad867d3
M7         = e73a8169 6da665c6 c48603a5 7b45ab33
C7         = 46e63f6e c6529a3b 2a7aa97c 0e280443
M8         = 221585e6 8ee31691 87fb0239 528632dd
C8         = 79803306 ad490c46 3d971205 dc99a211
M9         = 656c807e a3248b7b 46d002b2 b5c7458e
C9         = 4d74cec4 f07795ab f6127db4 529dfb57
M10        = b85b9ce9 5879e034 0859055e 3b0abbc3
C10        = a6eb9d1e 93820f49 d9c5f9e1 760cb686
M11        = eace8719 caa80265 c97205d5 dc4bcc90
C11        = 8f95155b d32ad9a3 463e905d 7ba480ee
M12        = 2fe18396 29ed7132 8a0f0449 f588557e
C12        = 6f120bf0 e6f4c66f a5c67815 65133712
M13        = 6898860e 042aecd8 4b2404c2 12c9222d
C13        = db74500e 895db74a ef3b3b87 25087f2b
M14        = a5bf8a89 ef679787 0cf50771 a60f66a2
```

```
C14       = f5879d17 7c0ddf7d 5772993a c137aeab
M15       = ee628536 57addf04 cdde07fa 414e11f1
C15       = b18a88a1 bceb93e0 a4b7ae95 4479bbfe
M16       = 2b4d81b9 b4e8ac53 8ea30666 688d881f
C16       = 7d75c4a5 e87bff2f 07471eb4 46fcdb73
M17       = 6c348421 992f31b9 4f8806ed 8fccff4c
C17       = b3456ccb e8f3e8d7 33568c84 f89d2145
M18       = 9123b896 42527ad6 13b109bf 75167485
C18       = b5363e85 edabc25d bd1a400d 5952742e
M19       = f1268bf8 84b4cd23 d29a0934 925703d6
C19       = 55abea1b 574ea033 45df9cd1 46f1c8e9
M20       = 34098f77 67f1be74 91e708a8 bb949a38
C20       = 8efc00fd 5d245efc de807875 cd46423d
M21       = 73708aef 4a36239e 50cc0823 5cd5ed6b
C21       = aa07abd7 b26d40b0 53945cfa 6aafab45
M22       = be578668 a17b58c1 171d0b90 e813a9e4
C22       = 4739c2bb 17ae5960 7ac250e2 c4c172fa
M23       = f58a89d7 19b11042 d6360b1b 0f52deb7
C23       = eda48d2b 146feccf 11c45d3b 2aac4c37
M24       = 30a58d58 faf46315 954b0a87 26914759
C24       = 4dbbb4e3 9e344d41 d05ca472 50186527
M25       = 77dc88c0 d733feff 54600a0c c1d0300a
C25       = ecda3d93 5776d708 42c9c5da 9a09dbe3
M26       = aaeb9457 2c6e95b0 1ae90de0 4f1dce47
C26       = 58a010aa f0149da7 5dfe9049 4676b663
M27       = f87e8fa7 bebf77e1 dbc20d6b a85cb914
C27       = d611b8cb bb9fb2ac f82aa88b fd6aab42
M28       = 3d518b28 5dfa04b6 98bf0cf7 819f20fa
C28       = a23131a6 d7352c69 e9790a6b 26b0292a
M29       = 7a288eb0 703d995c 59940c7c 66de57a9
C29       = 9026e0dd c60dc7fe 3ff024e4 5c853be8
M30       = b70f8237 9b70e203 1e450fcf d2181326
C30       = af09e79e 54d8c2e1 85b08d12 d638d687
M31       = fcd28d88 23baaa80 df6e0f44 35596475
C31       = f7bc7632 8b116b03 f5d1fd78 3f4d866d
M32       = 39fd8907 c0ffd9d7 9c130ed8 1c9afd9b
C32       = 0c2a4710 a2362a1f 7967fd45 1a7d188d
M33       = 7e848c9f ed38443d 5d380e53 fbdb8ac8
C33       = df3fc64e ff5998be 926a71d8 7836cf38
M34       = c3d3f068 76054f12 2461107d e92fea09
```

```
C34       = 11133bc0 6cdef5b2 0ba5cf12 b293ea83
M35       = c6f6923a 188d53af e54a10f6 0e6e9d5a
C35       = fe95113c c42ac4c4 bd53dfcb 41d01f1a
M36       = 03d996b5 fbc820f8 a637116a 27ad04b4
C36       = fbd5a26b 824d7a62 bdcad592 0ef8d4c8
M37       = 44a0932d d60fbd12 671c11e1 c0ec73e7
C37       = e75a94c8 e5b631b8 6e0f1153 f88b87aa
M38       = 89879faa 3d42c64d 20cd1252 742a3768
C38       = 773a8452 8fb77154 baaa0445 d517de8f
M39       = c25a9015 85888ece e1e612d9 936b403b
C39       = b53b90f0 6dce6530 593171f8 42eb5ab7
M40       = 0775949a 66cdfd99 a29b1345 baa8d9d5
C40       = 2d211e99 76cad436 d37bb281 74fd9aaf
M41       = 400c9102 4b0a6073 63b013ce 5de9ae86
C41       = 71f3983e 65f0af4d 028c1308 6488de12
M42       = 9d3b8d95 b0570b3c 2d391422 d32450cb
C42       = 0d292597 f79f9c95 f213724a 55e54437
M43       = cfae9665 2286e96d ec1214a9 34652798
C43       = 9b3ba456 072cdaa2 5bc5dae7 ab5e5c36
M44       = 0a8192ea c1c39a3a af6f1535 1da6be76
C44       = 0a3b8e65 0bf406a9 267783f1 69979a3e
M45       = 4df89772 ec0407d0 6e4415be fae7c925
C45       = 6a6cb8da bfaca611 7b7f1996 b83d4c92
M46       = 80df9bf5 07497c8f 2995160d 4e218daa
C46       = 6ed66263 70b356c4 bea4e69b fa281190
M47       = cb02944a bf83340c e8be1686 a960faf9
C47       = 65cf4cda 156b2025 b5b43852 022b0211
M48       = 0e2d90c5 5cc6475b abc3171a 80a36317
C48       = 96cff0a9 6e209fd5 065c9f34 e0edc899
M49       = 4954955d 7101dab1 6ae81791 67e21444
C49       = 61158848 8fb6a12b a2a155bc fa279420
M50       = b443a9ea aa7c91de 36d118c3 9d389f8d
C50       = 79a1892a 63751231 f45163bb cb8a7729
M51       = d4469a84 6c9a262b f7fa1848 7a79e8de
C51       = 25c71838 32d36692 22379a7b a086716c
M52       = 11699e0b 8fdf557c b48719d4 53ba7130
C52       = 466dbaf4 10f27161 202bd3e2 ce7fc5f3
M53       = 56109b93 a218c896 75ac195f b4fb0663
C53       = adcb04f6 86696807 38756fa3 7a350ccc
M54       = 9b379714 4955b3c9 327d1aec 003d42ec
```

```
C54     = 802a2d59 0b3a457a f449ba39 f8bad584
M55     = d0ea98ab f19ffb4a f3561a67 e77c35bf
C55     = b6bbd86d 5e708389 d18413f9 ddd9a92a
M56     = 15c59c24 12da881d b02b1bfb cebfac51
C56     = ff010e37 0ad1420e df6a5276 81b9f685
M57     = 52bc99bc 3f1d15f7 71001b70 29fedb02
C57     = a7af152e b0c0dc25 d96c9792 672c098e
M58     = 8f8b852b c4407eb8 3f891c9c a733254f
C58     = 957bc801 eaabe60c 27193122 a94cccb8
M59     = dd1e9edb 56919ce9 fea21c17 4072521c
C59     = 3b6d3712 3ea45568 15a4c417 3f903fc3
M60     = 18319a54 b5d4efbe bddf1d8b 69b1cbf2
C60     = 656e7869 42ef502b f5838dc4 44a89253
M61     = 5f489fcc 98137254 7cf41d00 8ef0bca1
C61     = 934b5a02 5051d909 a9d84ab2 547853c6
M62     = 926f934b 735e090b 3b251eb3 3a36f82e
C62     = b667b4da 06f5670f c014bb27 09e6e18c
M63     = d9b29cf4 cb944188 fa0e1e38 dd778f7d
C63     = 88033db1 446aaa10 a348ddaa d7d80d16
M64     = 1c9d987b 28d132df b9731fa4 f4b41693
C64     = 52d29028 818fae29 dad8c1fb 124d173f
M65     = 5be49de3 0516af35 78581f2f 13f561c0
C65     = b6131b03 2cc9c6ae 96051b5d 68aa7659
M66     = 66336194 1eab249a 4bc123f8 d15cd711
C66     = 58fbdb68 61d57ded 89977624 977ce584
M67     = a956a1bf 20fe6eb7 8aea2373 361da042
C67     = b9929b5e 371a0fb6 357c864d 4ea36d30
M68     = 6c79a530 c3bb1de0 c99722ef 1fde39ac
C68     = 198a06eb 2c013cab eadb6627 d555e3a6
M69     = 2b00a0a8 ee7c800a 08bc2264 f89f4eff
C69     = d1f0a42a b3045545 8e69a513 14825bfc
M70     = e627ac2f 0531fb55 4f6d21d7 4c590a70
C70     = 6b8c1b1a 03286dde f4ecf569 66f264d0
M71     = adfaa390 bdfbb3d6 8e46215c ab187d23
C71     = 082fe1f5 61373b7b 048b92ed 3b36c1d5
M72     = 68d5a71f 5ebec081 cd3b20c0 82dbe4cd
C72     = cd304dc4 682e63df 49b7da3b 1e780f3a
M73     = 2faca287 73795d6b 0c10204b 659a939e
C73     = 596f4ba2 4a20bb10 a9fa3124 6a7488b9
M74     = f29bbe10 88243624 429927a7 eb576dd3
```

```
C74      = 776ca237 97bc8e6b bca6eafd 8409dfe3

M75      = a00ea5e0 1af5d475 83b2272c 0c161a80

C75      = 828637a1 8145e141 83f331c6 606b7d86

M76      = 6521a16f f9b0a722 c0cf26b0 25d5836e

C76      = d7791efa bc262f54 835ec67c 7a224aff

M77      = 2258a4f7 d4773ac8 01e4263b c294f43d

C77      = af53bb31 351481e9 7a71d208 f603161e

M78      = ef7fa870 3f3a4197 46352588 7652b0b2

C78      = d4022c6e 13ea8576 e2828b8a 71889135

M79      = a4a2a7cf 87f00914 871e2503 9113c7e1

C79      = 934e9389 7d051877 7e33d2b5 51d450ba

M80      = 618da340 64b57a43 c463249f b8d05e0f

C80      = 0d505c6e 3820f48f 2d9d7965 7fda8c62

M81      = 26f4a6d8 4972e7a9 05482414 5f91295c

C81      = 7e83e4a2 e028cb71 aa4d49c3 77cb6878

M82      = dbe39a6f 920facc6 59712b46 a54ba295

C82      = e60a012c 3604a26b fcbd8bb8 ada3fa25

M83      = bbe6a901 54e91b33 985a2bcd 420ad5c6

C83      = 3b571f1e 45fc0552 6ac062f6 e38133b9

M84      = 7ec9ad8e b7ac6864 db272a51 6bc94c28

C84      = 64c12b59 f3f996cf aa4600f0 bbe782c7

M85      = 39b0a816 9a6bf58e 1a0c2ada 8c883b7b

C85      = 6d697d70 41a532be 99db1d5e 1802416e

M86      = f497a491 71268ed1 5ddd2969 384e7ff4

C86      = e13200d9 02b60040 c8d432e3 c6476faf

M87      = bf4aab2e c9ecc652 9cf629e2 df0f08a7

C87      = bb96999a e4f1f5cb 9f6c2787 1215a092

M88      = 7a65afa1 2aa9b505 df8b287e f6cc9149

C88      = f2ede003 89c33765 4d195eeb ceda25e7

M89      = 3d1caa39 076e28ef 1ea028f5 118de61a

C89      = bfa3ef0f 3171e7fa 90b5b1b8 e1a002d6

M90      = e02bb6ae fc3343a0 50292f19 9f401857

C90      = 56e2b617 3161c6c2 1e122148 86ecd966

M91      = b2bead5e 6ee2a1f1 91022f92 78016f04

C91      = d3a15f8e 6390dafe fc41cab0 472a7670

M92      = 7791a9d1 8da7d2a6 d27f2e0e 51c2f6ea

C92      = 5b666f14 2c224401 655c48e8 d1b2c12e

M93      = 30e8ac49 a0604f4c 13542e85 b68381b9

C93      = 4413e8b8 94bee1f2 05e193ee b695ab3d

M94      = fdcfa0ce 4b2d3413 54852d36 0245c536
```

C94 = 7e0693cb ed077fa8 2944064c ffc7d5d6

M95 = b612af71 f3e77c90 95ae2dbd e504b265

C95 = d25164b5 d9efcd07 17be88f0 17990efd

M96 = 733dabfe 10a20fc7 d6d32c21 ccc72b8b

C96 = 9e2abf1e 5f8ebdf4 2fb41ae7 d4eb6973

M97 = 3444ae66 3d65922d 17f82caa 2b865cd8

C97 = d7fe8071 8577524b 01297cf3 ae68a829

M98 = 8913d291 a6589902 6ea13284 39723c19

C98 = 0c6be895 d9e858a7 e2500452 42e2686e

M99 = 8c36b0c3 c8d085bf af8a320f de334b4a

C99 = 3629aeb3 673b422d 4aea4a5c 5a935941

M100 = 4919b44c 2b95f6e8 ecf73393 f7f0d2a4

C100 = 6cc0142b e8455f69 67284dc0 dd708f02

M101 = 0e60b1d4 06526b02 2ddc3318 10b1a5f7

C101 = d2839043 25718658 fac2fb23 59d3994f

M102 = c347bd53 ed1f105d 6a0d30ab a477e178

C102 = a5b5a2bf 19ec33b3 d2296d4a 3735981e

M103 = 889ab2ec 55d558de ab263020 4336962b

C103 = e97eb2ee e9769c3d ea6ad1bb ea079a88

M104 = 4db5b663 b6902b89 e85b31bc 6af50fc5

C104 = 042f1f1c 59a41204 1484dd2b 426eb392

M105 = 0accb3fb 9b57b663 29703137 8db47896

C105 = 45e15f74 bb550567 a80a5dac acc18ebb

M106 = d7fbaf6c 600add2c 67f936db 037986db

C106 = 9e285b68 8a3338f8 dc2e12de d3a89153

M107 = 856eb49c f2db3f7d a6d23650 e438f188

C107 = 48f6e6c3 0b1448b7 a94983d3 1416029d

M108 = 4041b013 119e4c2a e5af37cc cdfb6866

C108 = a4645c35 b9a4f509 89704523 0e98fac1

M109 = 0738b58b 3c59d1c0 24843747 2aba1f35

C109 = f8ec48ec 33ad7364 20ea077f 16be98b8

M110 = ca1fb90c d714aa9f 635534f4 9e7c5bba

C110 = 8de31e96 1bb879e2 ca169749 51afab6f

M111 = 81c2b6b3 6fdee21c a27e347f 793d2ce9

C111 = f602eab6 e1373191 fc30b633 8cd82741

M112 = 44edb23c 8c9b914b e10335e3 50feb507

C112 = 762c51e6 d30a4eab 869c8827 0d698121

M113 = 0394b7a4 a15c0ca1 20283568 b7bfc254

C113 = e1db681b 5fb862fc b1c3747f ab057c1c

M114 = fe838b13 7a2147ce 7c113a3a 4d65499d

```
C114      = e77d4ba4 812e0730 4eb1ee0e c233685d

M115      = 9e86b87d bcc7f03b bd3a3ab1 aa243ece

C115      = 177fd714 1f206a6f 06940efd a023309f

M116      = 5ba9bcf2 5f82836c fe473b2d 83e7a720

C116      = c738f59b 0715dded 2efe635d a073b5a3

M117      = 1cd0b96a 72451e86 3f6c3ba6 64a6d073

C117      = c99dbfa3 ebd3f018 bba8b961 96818130

M118      = d1f7b5ed 990865d9 78bd3815 d06094fc

C118      = eebd79e4 c7378d33 3941a3c5 45ee8d37

M119      = 9a2aba52 21c22d5a b996389e 3721e3af

C119      = dbdce382 e9abef5d 39f309ad a6ce7e8c

M120      = 5f05bedd c2875e0d faeb3902 1ee27a41

C120      = 7f851259 1a77d8a5 2f146735 6ebec181

M121      = 187cbb45 ef40c3e7 3bc03989 f9a30d12

C121      = 8e423a41 34eca7b9 f8a1c48e 6fbc50ec

M122      = c54ba7d2 141da8a8 75493e65 776ef35f

C122      = b6e40968 80bfc03f c7aa655b c0e12a25

M123      = 97debc22 86cc4af9 b4623eee 902f840c

C123      = 3a1a64aa b9addbd6 eb3ad3b1 1f2fe168

M124      = 52f1b8ad 658939ae f71f3f72 b9ec1de2

C124      = 1559a703 6187d461 52dbf04d 4bac3ca0

M125      = 1588bd35 484ea444 36343ff9 5ead6ab1

C125      = 16136377 e935b0fd e2c2ab4e 1718b30e

M126      = d8afb1b2 a303df1b 71e53c4a ea6b2e3e

C126      = 995211d4 8695b1a2 a59b377d d2829f31

M127      = 9372be0d 1bc99798 b0ce3cc1 0d2a596d

C127      = e8c5844a c73c27d1 3b0b6df9 3142fdaa

M128      = 565dba82 f88ce4cf f3b33d5d 24e9c083

C128      = 64c755f6 43c48ee6 1e5af291 ea4df86f

M129      = 49ec0480 3169ebe6 83ebad1e 36f0ef66

C129      = ebd5ccb0 b61ca905 29138303 f3377d22


MACT      = ebd5ccb0
```

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2007-05	SA3#47				Initial version contains commented Table of Contents with references to TS 33.102 and TR 33.821	-	0.0.0
2007-07	SA3#48				Inclusion of content based on S3-070524, S3-070525 and S3-070526	0.0.0	0.1.0
2007-10	SA3#49				Inclusion of content based on S3-070770, S3-070776, S3-070801, S3-070743, S3-070766	0.1.0	0.2.0
2007-12	SA3#49bis				Inclusion of content based on S3-071021, S3-070963, S3-070951, S3-071015, S3-070968, S3-070923, S3-070922, S3-070919, S3-070960, S3-070953, S3-070958	0.2.0	0.3.0
2008-02	SA3#50				Inclusion of content based on S3-080193; S3-080047; S3-080044; S3-080139; S3-080059; S3-080082; S3-080085; S3-080155; S3-080205; S3-080053; S3-080057; S3-080170; S3-080144; S3-080165; S3-080079; S3-080068;S3-080135; S3-080168; S3-080083	0.3.0	0.4.0
2008-03	SA#39				Presented for information at SA	0.4.0	1.0.0
2008-04	SA3#51				Additions based on S3-080418, 457, 367, 316, 390, 391,325, 490 (414), 466, 380, 364, 402,318, 314, 407, 504	1.0.0	1.1.0
2008-05					MCC preparation for approval	1.1.0	2.0.0
2008-06	SA#40	SP-080257			SA#40 Approval	2.0.0	8.0.0
2008-09	SA#41	SP-080487	0010	1	KeNB forward security simplification	8.0.0	8.1.0
2008-09	SA#41	SP-080653	0040	-	IRAT related changes (merge of CRs 14,24,27)	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0016	1	CR-33401: Security Context Selection on IRAT handover to E-UTRAN	8.0.0	8.1.0
2008-09	SA#41	SP-080653	0041	-	KeNB derivation when activating cached security context and AS SMC clarifications (merge of CR0011 and CR0015)	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0006	1	CR 33.401: Correction of text on security context and authentication data	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0009	1	CR 33.401 :Ensuring security context freshness on handover from an SGSN towards MME	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0012	1	CR-33401: NAS SMC handling clarifications	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0023	1	CR: Algorithm selection cleanup	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0022	1	CR: Editorial cleanup of TS 33.401	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0021	1	CR: Removal of editor's note related to key change on-the-fly	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0013	1	CR-33401: Replay UE capabilities in the NAS SMC Command message	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0007	1	CR 33.401: Completing the specification on air interface ciphering and integrity algorithms inputs	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0005	2	CS key derivation in SRVCC	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0039	-	Note on SNID IP binding	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0038	-	CR on KSI security context desynchronization	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0035	1	NAS integrity tag term consistent	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0017	1	Key freshness during mobility from E-UTRAN to UTRAN/GERAN	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0018	-	Distribution of authentication data from HSS to serving network	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0025	1	CR: Additional inputs to EPS Key Derivation Function (KDF)	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0032	2	Clarification on MSIN_IMEI confidentiality protected limitation	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0029	1	NAS key re-keying	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0030	-	EPS key hierarchy	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0001	1	SAE: backhaul link management plane protection	8.0.0	8.1.0
2008-09	SA#41	SP-080487	0026	1	CR: Add requirement to have replay protection for NAS and RRC	8.0.0	8.1.0
2008-10	--	--	--	--	MCC editorial amendment (9.1.2 missing 'Cached context')	8.1.0	8.1.1
2008-12	SA#42	SP-080738	42	1	Specification of security algorithms for EPS	8.1.1	8.2.0
2008-12	SA#42	SP-080738	43	-	KDFs for EPS shall not be negotiated	8.1.1	8.2.0
2008-12	SA#42	SP-080738	44	-	Removal of editor's notes that are resolved or are related to new functionality	8.1.1	8.2.0
2008-12	SA#42	SP-080738	45	-	KeNB Derivation During inter-RAT TAU	8.1.1	8.2.0
2008-12	SA#42	SP-080738	47	-	Inter-RAT change from GERAN/UTRAN to E-UTRAN with mapped context	8.1.1	8.2.0
2008-12	SA#42	SP-080738	48	1	Storage of EPS NAS security context in the UE	8.1.1	8.2.0
2008-12	SA#42	SP-080738	51	-	Clarfication in definitions	8.1.1	8.2.0
2008-12	SA#42	SP-080738	52	-	Correction of definition and usage of Key Set Identifier (KSI) in EPS	8.1.1	8.2.0
2008-12	SA#42	SP-080738	53	-	Correction of handling of EPS security contexts	8.1.1	8.2.0
2008-12	SA#42	SP-080738	54	1	Correction of storage of security contexts during state transitions, handling of mapped contexts	8.1.1	8.2.0
2008-12	SA#42	SP-080738	55	1	Introducing the generic term 'eKSI' for the Key Set Identifier in E-UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	56	-	Correction of definition of GUTI in EPS	8.1.1	8.2.0
2008-12	SA#42	SP-080738	57	-	Transfering unused AVs	8.1.1	8.2.0
2008-12	SA#42	SP-080738	60	2	Corrections to the KDF input parameters	8.1.1	8.2.0
2008-12	SA#42	SP-080738	61	-	RLF recovery procedure	8.1.1	8.2.0
2008-12	SA#42	SP-080738	62	-	Addition of missing requirements to drop messages with wrong or missing MAC	8.1.1	8.2.0
2008-12	SA#42	SP-080738	63	-	NAS uplink and downlink ciphering	8.1.1	8.2.0
2008-12	SA#42	SP-080738	68	-	Correction and addition of the security features in SRVCC.	8.1.1	8.2.0
2008-12	SA#42	SP-080738	69	-	Correction of idle mode mobility from E-UTRAN to UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	70	-	Correction of idle mode mobility from UTRAN to E-UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	71	-	Correction of handover procedure from E-UTRAN to UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	72	-	Correction of definition and usage of Key Set Identifier (KSI) in EPS	8.1.1	8.2.0
2008-12	SA#42	SP-080738	73	-	EPS algorithm selection and bidding down attack and added MME Algorithm selection with a prioritized list	8.1.1	8.2.0
2008-12	SA#42	SP-080738	74	-	Correction of Handover from UTRAN to E-UTRAN and activation of cached context	8.1.1	8.2.0
2008-12	SA#42	SP-080738	75	-	Removal of editor's notes related to section 6 and clarification of ASME	8.1.1	8.2.0
2008-12	SA#42	SP-080738	77	-	Inter-RAT change from GERAN/UTRAN to E-UTRAN with mapped context	8.1.1	8.2.0
2008-12	SA#42	SP-080738	78	-	Clarification on MME requirement in UTRAN-eUTRAN handover	8.1.1	8.2.0
2008-12	SA#42	SP-080738	79	-	Correction of idle mode mobility from UTRAN to E-UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	80	-	Storage of cached EPS NAS security contexts after security interworking from UTRAN to E-UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	81	-	E-UTRAN key identification	8.1.1	8.2.0
2008-12	SA#42	SP-080738	82	-	Clarification on User Data Protection	8.1.1	8.2.0
2008-12	SA#42	SP-080738	83	-	Corrections to section 6.2 describing the key hierarchy	8.1.1	8.2.0
2008-12	SA#42	SP-080738	84	-	Correction of text on handling of security capabilities in handover from E-UTRAN to UTRAN (section 9.2.1)	8.1.1	8.2.0
2008-12	SA#42	SP-080738	85	-	Correction of text on activation of security in E-UTRAN and consequences for Handover from E-UTRAN	8.1.1	8.2.0
2008-12	SA#42	SP-080738	87	-	Harmonising clauses 9.2.2 on HO from UTRAN and 10.3.2 on HO from GERAN with clause 7.2.9 on key-change-on-the fly	8.1.1	8.2.0
2008-12	SA#42	SP-080738	88	-	Correction of S1/X2 transport protection	8.1.1	8.2.0
2008-12	SA#42	SP-080738	89	-	Removal of editor"s notes on security requirements on eNodeB	8.1.1	8.2.0
2008-12	SA#42	SP-080738	90	-	Requirements on secure environment within eNB	8.1.1	8.2.0
2008-12	SA#42	SP-080738	91	-	Corrections to Section 7.2.3 on E-UTRAN key lifetime	8.1.1	8.2.0
2008-12	SA#42	SP-080738	92	-	E-UTRAN handover key derivations correction	8.1.1	8.2.0
2008-12	SA#42	SP-080738	64	1	Corrections to security procedures for interworking between E-UTRAN and GERAN and change of subclause titles	8.1.1	8.2.0
2008-12	--	-	--	-	Editorial correction	8.2.0	8.2.1
2009-03	SA#43	SP-090129	118	-	Preventing mobility into E-UTRAN for UE with SIM access	8.2.1	8.3.0
2009-03	SA#43	SP-090129	124	-	Removal of editors note in clause 5.4 'Other security features'	8.2.1	8.3.0
2009-03	SA#43	SP-090129	127	-	Correction on dropping AS and NAS SMC when detecting IP failure for SMC Command	8.2.1	8.3.0

2009-03	SA#43	SP-090129	131	-	Clarification on Token Calculation for RCR procedure	8.2.1	8.3.0
2009-03	SA#43	SP-090129	138	-	Removal of Editor"s note about PDCP COUNT threshold value	8.2.1	8.3.0
2009-03	SA#43	SP-090129	93	1	Correction of EPS NAS Security Context definition	8.2.1	8.3.0
2009-03	SA#43	SP-090129	166	-	NAS COUNT for mapped security context	8.2.1	8.3.0
2009-03	SA#43	SP-090129	117	1	Storage of the EPS NAS security context in non-volatile memory of the UE	8.2.1	8.3.0
2009-03	SA#43	SP-090129	116	1	Handling of security contexts in transition to EMM-DEREGISTERED state	8.2.1	8.3.0
2009-03	SA#43	SP-090129	97	1	Security context for transition to EMM DEREGISTERED	8.2.1	8.3.0
2009-03	SA#43	SP-090129	114	1	Definition of UE security capabilities	8.2.1	8.3.0
2009-03	SA#43	SP-090129	125	1	Clarification on ciphering for GUTI confidentiality	8.2.1	8.3.0
2009-03	SA#43	SP-090129	136	1	AKA when NAS COUNT about to wrap around	8.2.1	8.3.0
2009-03	SA#43	SP-090129	95	1	Key derivation figures	8.2.1	8.3.0
2009-03	SA#43	SP-090129	102	1	Rekeying of the entire EPS key hierarchy	8.2.1	8.3.0
2009-03	SA#43	SP-090129	121	1	Various corrections related to algorithm definitions, NAS token and P-TMSI signature	8.2.1	8.3.0
2009-03	SA#43	SP-090129	145	1	Adding EARFCN-DL to KeNB* derivation	8.2.1	8.3.0
2009-03	SA#43	SP-090129	128	1	Forward security corrections related to R3-083569 (S3-09xxxx)	8.2.1	8.3.0
2009-03	SA#43	SP-090129	130	1	KeNB re-keying corrections	8.2.1	8.3.0
2009-03	SA#43	SP-090129	135	-	Removal of editor"s note on key-change-on-the-fly for possible RLF	8.2.1	8.3.0
2009-03	SA#43	SP-090129	108	1	Multi-KeNB* forwarding for RRCConnectionRe-establishment	8.2.2	8.3.0
2009-03	SA#43	SP-090129	143	1	Definition of NULL ciphering	8.2.1	8.3.0
2009-03	SA#43	SP-090129	164	1	Correction of definition of direction bit DIR	8.2.1	8.3.0
2009-03	SA#43	SP-090129	103	1	NAS COUNT start value	8.2.1	8.3.0
2009-03	SA#43	SP-090129	154	1	Correction of NAS COUNTs reset to the start value	8.2.1	8.3.0
2009-03	SA#43	SP-090129	96	1	Algorithm selection clarification	8.2.1	8.3.0
2009-03	SA#43	SP-090129	129	1	Verification of NONCEU	8.2.1	8.3.0
2009-03	SA#43	SP-090129	133	1	Removal of 4LSB and eKSI in the AS SMC	8.2.1	8.3.0
2009-03	SA#43	SP-090129	134	-	Removal of editor"s note related 4LSB NAS downlink COUNT in the HO COMMAND	8.2.1	8.3.0
2009-03	SA#43	SP-090129	156	1	Correcting the EPS security context fetching between MMEs	8.2.1	8.3.0
2009-03	SA#43	SP-090129	153	1	Correction of description	8.2.1	8.3.0
2009-03	SA#43	SP-090129	153	1	Correction of MAC-I	8.2.1	8.3.0
2009-03	SA#43	SP-090129	105	1	Transparent container in UTRAN to E-UTRAN handover	8.2.1	8.3.0
2009-03	SA#43	SP-090129	106	1	UE keys for E-UTRAN and GERAN	8.2.1	8.3.0
2009-03	SA#43	SP-090129	115	1	Handling of UE capabilities in Idle Mode Mobility and Handover to E-UTRAN	8.2.1	8.3.0
2009-03	SA#43	SP-090129	155	1	Adding definition of native contexts and clarifying the mapped and current definitions	8.2.1	8.3.0
2009-03	SA#43	SP-090129	104	1	I-RAT handover E-UTRAN and UTRAN	8.2.1	8.3.0
2009-03	SA#43	SP-090129	158	1	Some corrections to the UTRAN to E-UTRAN handover procedure	8.2.1	8.3.0
2009-03	SA#43	SP-090129	132	1	Selection of key set and security activation on Handover from GERAN/UTRAN to E-UTRAN	8.2.1	8.3.0
2009-03	SA#43	SP-090129	147	1	Correction of security requirements on eNodeB	8.2.1	8.3.0
2009-03	SA#43	SP-090129	119	-	Replacing the cached security context with the mapped security context and setting START value in SRVCC	8.2.1	8.3.0
2009-03	SA#43	SP-090129	162	-	Handling of integrity protection failures	8.2.1	8.3.0
2009-03	SA#43	SP-090129	120	1	Clarification on security contexts and correction of protection of TAU requests in E-UTRAN , Handling of UE capabilities in Idle Mode Mobility	8.2.1	8.3.0
2009-03	SA#43	SP-090129	113	1	Clarification for KSIsgsn	8.2.1	8.3.0
2009-03	SA#43	SP-090129	99	2	Making CK" and IK"	8.2.1	8.3.0
2009-03	SA#43	SP-090136	161	-	Add reference to ciphering indicator feature specification	8.2.1	8.3.0
2009-03					Editorial corrections	8.3.0	8.3.1
2009-06	SA#44	SP-090268	167	1	Storage and handling of EPS NAS security context	8.3.1	8.4.0
2009-06	SA#44	SP-090268	168	1	UE security capability handling at S1 HO	8.3.1	8.4.0
2009-06	SA#44	SP-090268	172	1	Correction of TAU procedure after IRAT Handover to E-UTRAN	8.3.1	8.4.0
2009-06	SA#44	SP-090268	173	-	Correction of UE EPS security capability default set	8.3.1	8.4.0
2009-06	SA#44	SP-090268	174	1	Resolving ed note on PLMN ID encoding and correction of KDF	8.3.1	8.4.0
2009-06	SA#44	SP-090268	176	1	Update of abbreviations list	8.3.1	8.4.0
2009-06	SA#44	SP-090268	177	1	Definition of forward security	8.3.1	8.4.0
2009-06	SA#44	SP-090268	178	2	Removal of NAS keys storage	8.3.1	8.4.0
2009-06	SA#44	SP-090268	181	1	Corrections for idle mode procedure in EUTRAN	8.3.1	8.4.0
2009-06	SA#44	SP-090268	184	-	Corrections for Attach procedure	8.3.1	8.4.0
2009-06	SA#44	SP-090268	188	-	The correction on input key of Null ciphering algorithm	8.3.1	8.4.0
2009-06	SA#44	SP-090268	190	-	Clarification on Multiple KeNB*s Preparation in Section 7.4.3	8.3.1	8.4.0
2009-06	SA#44	SP-090268	191	2	The definition of eKSI format	8.3.1	8.4.0
2009-06	SA#44	SP-090268	197	-	Forward security chaining modificarion	8.3.1	8.4.0
2009-06	SA#44	SP-090268	200	-	Changing SGSN to the enhanced MSC server in SRVCC	8.3.1	8.4.0
2009-06	SA#44	SP-090268	202	2	Correction of KeNB derivations	8.3.1	8.4.0
2009-06	SA#44	SP-090268	203	1	Correction of EPS AS Security Mode Command Procedure	8.3.1	8.4.0
2009-06	SA#44	SP-090268	204	4	Correction of EPS AS/NAS Security Context definition	8.3.1	8.4.0
2009-06	SA#44	SP-090268	205	-	Keys to be used in HO from E-UTRAN to UTRAN and GERAN	8.3.1	8.4.0
2009-06	SA#44	SP-090268	206	1	Editorial corrections	8.3.1	8.4.0
2009-06	SA#44	SP-090268	207	1	Correction on KeNB refresh	8.3.1	8.4.0
2009-06	SA#44	SP-090268	208	1	Correction on Key Identification	8.3.1	8.4.0

2009-06	SA#44	SP-090268	209	-	Keys to be used in HO from UTRAN and GERAN to E-UTRAN	8.3.1	8.4.0
2009-06	SA#44	SP-090268	210	1	NH derivation rules for re-keying	8.3.1	8.4.0
2009-06	SA#44	SP-090268	211	1	RRC Connection Re-establishment - security corrections	8.3.1	8.4.0
2009-06	SA#44	SP-090268	216	1	Randomness requirements on nonces for LTE	8.3.1	8.4.0
2009-06	SA#44	SP-090268	214	2	Clarifications to calculating the initial K_eNB	8.3.1	8.4.0
2009-06	SA#44	SP-090268	218	1	Modification of the title of 8.1.1	8.3.1	8.4.0
2009-06	SA#44	SP-090268	217	1	Additional security requirement on eNB	8.3.1	8.4.0
2009-06	SA#44	SP-090268	219	1	Clarification on the requirement of AS algorithm selection procedure during intra-eNB handover	8.3.1	8.4.0
2009-06	SA#44	SP-090268	220	2	SAE: Clarification on eNodeB management plane protection	8.3.1	8.4.0
2009-06	SA#44	SP-090268	224	-	Correction of distribution of authentication data	8.3.1	8.4.0
2009-06	SA#44	SP-090268	225	-	Correction of key setting during AKA	8.3.1	8.4.0
2009-06	SA#44	SP-090268	227	-	Resolution of an Editor''s note on signaling messages, which are not confidentiality protected	8.3.1	8.4.0
2009-06	SA#44	SP-090268	228	1	Rules on concurrent runs of security procedures	8.3.1	8.4.0
2009-06	SA#44	SP-090268	231	1	Update of NAS SMC figure	8.3.1	8.4.0
2009-06	SA#44	SP-090268	232	1	An operator option to use NAS signalling confidentialty protection	8.3.1	8.4.0
2009-06	SA#44	SP-090268	233	-	Inconsistent text in section 9.2.2.2 related to HO from UTRAN to E-UTRAN	8.3.1	8.4.0
2009-06	SA#44	SP-090268	252	-	Clarification on key identification and KeNB derivation	8.3.1	8.4.0
2009-06	SA#44	SP-090268	254	1	Correction of eKSI usage in AS and other corrections of 33.401	8.3.1	8.4.0
2009-06	SA#44	SP-090268	255	-	Correction of security context transfer over S10	8.3.1	8.4.0
2009-06	SA#44	SP-090268	256	-	Test vectors for 128-EEA2 and 128-EIA2	8.3.1	8.4.0
2009-06	SA#44	SP-090466	212	2	Clarification on the requirement of AS algorithm selection procedure during intra-eNB handover	8.3.1	8.4.0

History

Document history		
V8.1.1	January 2009	Publication
V8.2.1	January 2009	Publication
V8.3.1	April 2009	Publication
V8.4.0	July 2009	Publication