NETMANIAS
www.netmanias.com

NMC
CONSULTING GROUP
www.nmcgroups.com

# LTE Security II: NAS and AS Security

Table of Contents



Once LTE authentication is completed, UE and MME share the same $K_{ASME}$. This document describes NAS and AS security setup procedures in which NAS and AS security keys are generated based on $K_{ASME}$, and how control messages and user packets are securely delivered thereafter. Then, it discusses security contexts to be stored in EPS entities as a result of the NAS and AS security setup, followed by a summary of the security keys used in LTE.

**October 14, 2014**

**(Initial Release: May 23, 2011)**

www.netmanias.com

**NMC Consulting Group (tech@netmanias.com)**

**About NMC Consulting Group**

NMC Consulting Group is an advanced and professional network consulting company, specializing in IP network areas (e.g., FTTH, Metro Ethernet and IP/MPLS), service areas (e.g., IPTV, IMS and CDN), and wireless network areas (e.g., Mobile WiMAX, LTE and Wi-Fi) since 2002.

## Netmanias LTE Technical Documents

Visit http://www.netmanias.com/en/ to view and download more technical documents.

| Index | Topic | Document Title | Document presented here |
|---|---|---|---|
| 1 | Network Architecture | LTE Network Architecture: Basic | |
| 2 | Identification | LTE Identification I: UE and ME Identifiers | |
| 3 | | LTE Identification II: NE and Location Identifiers | |
| 4 | | LTE Identification III: EPS Session/Bearer Identifiers | |
| 5 | **Security** | LTE Security I: LTE Security Concept and LTE Authentication | |
| **6** | | **LTE Security II: NAS and AS Security** | **O** |
| 7 | QoS | LTE QoS: SDF and EPS Bearer QoS | |
| 8 | EMM | LTE EMM and ECM States | |
| 9 | | Eleven EMM Cases in an EMM Scenario | |
| 10 | | LTE EMM Procedure 1. Initial Attach – Part 1. Case of Initial Attach | |
| 11 | | LTE EMM Procedure 1. Initial Attach – Part 2. Call Flow of Initial Attach | |
| 12 | | LTE EMM Procedure 2. Detach | |
| 13 | | LTE EMM Procedure 3. S1 Release | |
| 14 | | LTE EMM Procedure 4. Service Request | |
| 15 | | LTE EMM Procedure 5. Periodic TAU | |
| 16 | | LTE EMM Procedure 6. Handover without TAU – Part 1. Overview of LTE Handover | |
| 17 | | LTE EMM Procedure 6. Handover without TAU – Part 2. X2 Handover | |
| 18 | | LTE EMM Procedure 6. Handover without TAU – Part 3. S1 Handover | |
| 19 | | LTE EMM Procedure 7. Cell Reselection without TAU | |
| 20 | | LTE EMM Procedure 8 & 9. Handover and Cell Reselection with TAU | |
| 21 | | LTE EMM Procedure 10 & 11. Move to Another City and Attach | |
| 22 | PCC | LTE Policy and Charging Control (PCC) | |
| 23 | Charging | LTE Charging I: Offline | |
| 24 | | LTE Charging II: Online (TBD) | |
| 25 | IP Address Allocation | LTE: IP Address Allocation Schemes I: Basic | |
| 26 | | LTE: IP Address Allocation Schemes II: A Case for Two Cities | |

## Abbreviations

| | |
|---|---|
| AES | Advanced Encryption Standard |
| AKA | Authentication and Key Agreement |
| AS | Access Stratum |
| ASME | Access Security Management Entity |
| AuC | Authentication Center |
| AV | Authentication Vector |
| CK | Cipher Key |
| DRB | Data Radio Bearer |
| EEA | EPS Encryption Key |
| EIA | EPS Integrity Key |
| eNB | Evolved Node B |
| EPS | Evolved Packet System |
| E-UTRAN | Evolved Universal Terrestrial Radio Access Network |
| HSS | Home Subscriber Server |
| IK | Integrity Key |
| IMSI | International Mobile Subscriber Identity |
| KDF | Key Derivation Function |
| KSI | Key Set Identifier |
| LTE | Long Term Evolution |
| MAC | Message Authentication Code |
| MAC-I | Message Authentication Code for Integrity |
| MME | Mobility Management Entity |
| NAS | Non Access Stratum |
| NAS-MAC | Message Authentication Code for NAS for Integrity |
| NCC | Next hop Chaining Counter |
| NH | Next Hop |
| PDCP | Packet Data Convergence Protocol |
| RRC | Radio Resource Control |
| SRB | Signaling Radio Bearer |
| UE | User Equipment |
| UP | User Plane |
| USIM | Universal Subscriber Identity Module |
| ZUC | Zu Chongzhi |

## I. Introduction

In LTE Security I [1], Part I of the LTE Security technical document, we have discussed LTE authentication based on EPS AKA procedure and learned a UE and an MME get to share the $K_{ASME}$ when authenticated. In this document, we will explain NAS and AS security setup procedures to be performed based on $K_{ASME}$, and how data are transmitted in user and control planes after the security setup procedures.

Chapter II herein will explain NAS security setup procedure and how NAS messages are sent and received after the procedure. Chapter III will cover AS security setup procedure and how RRC messages and IP packets are transmitted thereafter. Chapter IV will provide a description of EPS security contexts and security data to be set in EPS entities (UE, eNB, MME and HSS). Finally, Chapter V will summarize all the security keys covered in the LTE Security technical document (LTE Security I and II).

Before we move on to security setup procedures, we will look in the protocol stacks where NAS and AS security are actually applied to. Figure 1 shows the protocol stacks related to NAS and AS security setup.

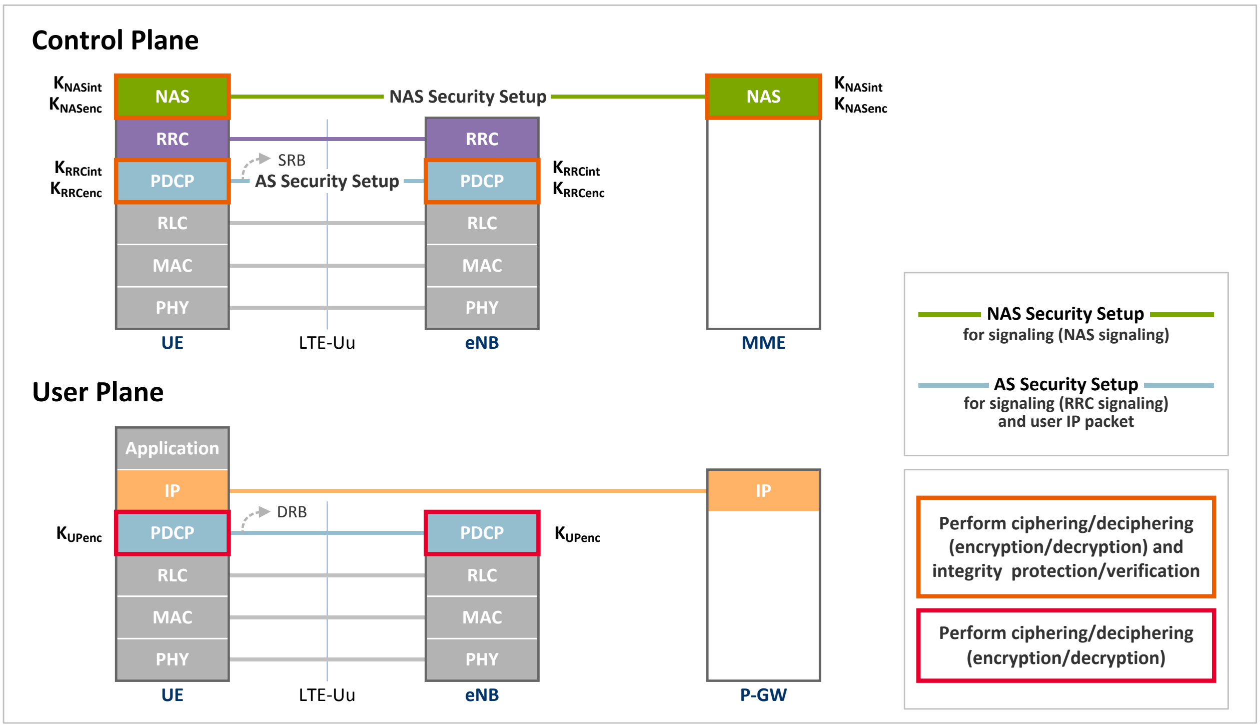



**Figure 1. Protocol stacks for security setup**

- **NAS Security**: The purpose of NAS security is to securely deliver NAS signaling messages between a UE and an MME in the control plane using NAS security keys. The NAS security keys are derived from $K_{ASME}$ and new keys are generated every time EPS AKA is performed (every time a new $K_{ASME}$ is generated). After the NAS security setup is completed, the UE and the MME get to share a NAS encryption key ($K_{NASenc}$) and a NAS integrity key ($K_{NASint}$), which are used in encryption and integrity protection, respectively, of NAS messages before transmitting.
- **AS Security**: The purpose of AS security is to securely deliver RRC messages between a UE and an eNB in the control plane and IP packets in the user plane using AS security keys. The AS security keys are derived

from $K_{eNB}$ and new keys are generated every time a new radio link is established (that is, when RRC state moves from idle to connected)[1]. After the AS security setup is completed, the UE and the eNB get to share an RRC integrity key ($K_{RRCint}$), RRC encryption key ($K_{RRCenc}$) and user plane encryption key ($K_{UPenc}$). Encryption and integrity protection using these keys are performed at the PDCP layer. $K_{RRCint}$ and $K_{RRCenc}$ are used to securely deliver RRC messages in the control plane through an SRB (Signaling Radio Bearer) over radio links. The RRC messages are encrypted using $K_{RRCenc}$ and integrity protected using $K_{RRCint}$ at the PDCP layer before being sent. $K_{UPenc}$ is used to securely deliver IP packets in the user plane through a DRB (Data Radio Bearer) over radio links. The IP packets are encrypted using $K_{UPenc}$ at the PDCP layer before being sent.

## II. NAS Security

A detailed description of the NAS security previously mentioned in LTE Security I [1] will be given below. A NAS security setup procedure consists of NAS signaling, between a UE and an MME, by a **Security Mode Command** message that the MME sends to the UE and a **Security Mode Command** message that the UE sends to the MME. Descriptions of the NAS security setup procedure by NAS messages and how NAS messages are delivered thereafter will be provided in Sections 2.1 and 2.2, respectively.

### 2.1 NAS Security Setup

**(1) Delivering a Security Mode Command message**

Figure 2 shows how a **Security Mode Command** message is delivered during the NAS security setup procedure. The MME, by sending a **Security Mode Command** message to the UE, informs the UE that it is authenticated by the network and the NAS security setup procedure for secure message delivery between them is initiated. The **Security Mode Command** message is integrity protected and then sent to the UE, which then derives NAS security keys (a cipher key and an integrity key) and verifies the integrity of the message using the integrity key.

A simplified LTE authentication procedure that precedes the NAS security setup procedure is shown as ❶ and ❷ in Figure 2 [1]. The same $K_{ASME}$ is shared by the UE and the MME as a result of the LTE authentication. We will explain the NAS security setup procedure presuming the MME allocates a $KSI_{ASME}$ to identify $K_{ASME}$ as 1 ("001").

1 During handover, a new key is generated even when RRC state is active. However, since security during handover is out of the scope of this document, it is not covered herein.

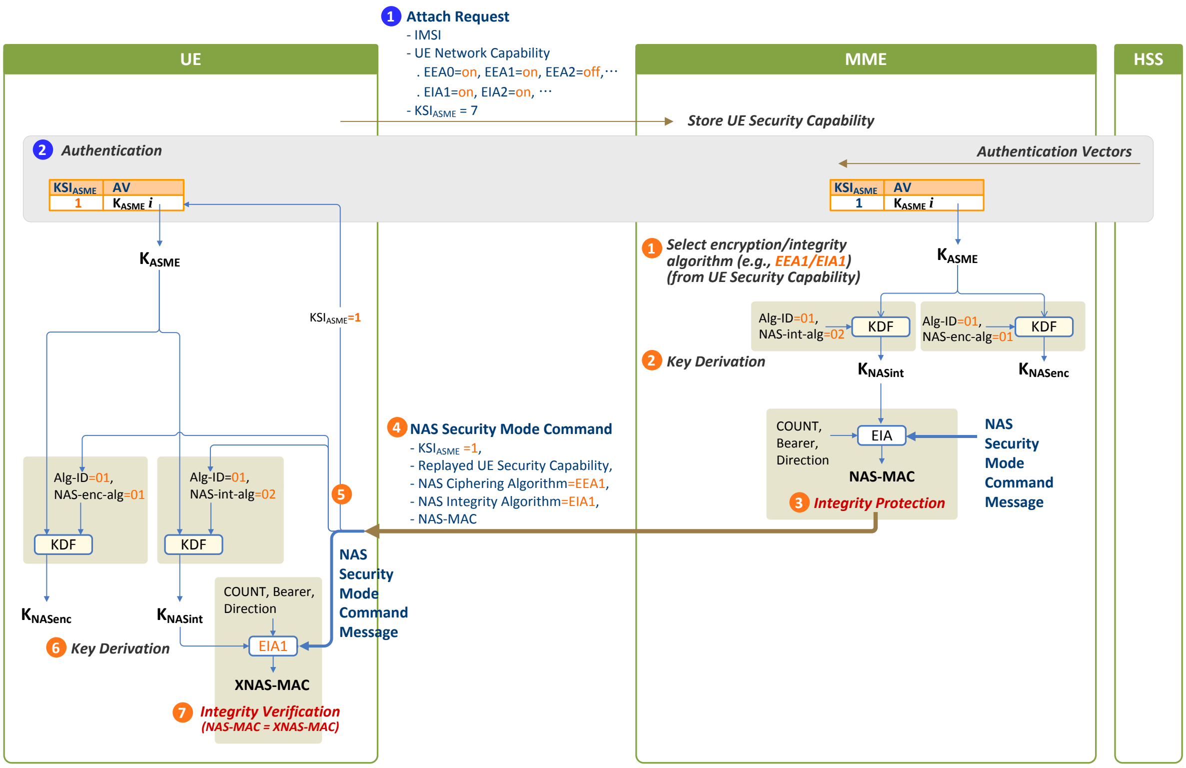



**Figure 2. NAS security setup: Delivery of a Security Mode Command message**

## 1 [MME] Selecting security algorithms

The MME selects ciphering and integrity algorithm to be applied to NAS messages based on UE Network Capability information included in the received **Attach Request** message from the UE. Figure 2 shows an example of selecting EEA1 for an encryption algorithm and EIA1 for an integrity algorithm, i.e., SNOW 3G algorithm (see LTE Security I [1]).

## 2 [MME] Deriving NAS security keys

The MME derives $K_{NASint}$ and $K_{NASenc}$ from $K_{ASME}$ using the algorithm IDs and algorithm distinguishers of the selected security algorithms. Table 1 lists algorithm IDs and algorithm distinguishers [2].

- $K_{NASint}$ = KDF($K_{ASME}$, NAS-int-alg, Alg-ID)
- $K_{NASenc}$ = KDF($K_{ASME}$, NAS-enc-alg, Alg-ID)

**Table 1. Security algorithm IDs and algorithm distinguishers [2]**

| Algorithm ID | Description | Value |
|---|---|---|
| 128-EEA0 | Null ciphering algorithm | 0000 |
| 128-EEA1 | SNOW 3G | 0001 |
| 128-EEA2 | AES | 0010 |
| 128-EEA3 | ZUC (optional) | 0011 |
| 128-EIA1 | SNOW 3G | 0001 |
| 128-EIA2 | AES | 0010 |
| 128-EIA3 | ZUC (optional) | 0011 |

| Algorithm Distinguisher | Value |
|---|---|
| NAS-enc-alg | 0x01 |
| NAS-int-alg | 0x02 |
| RRC-enc-alg | 0x03 |
| RRC-int-alg | 0x04 |
| UP-enc-alg | 0x05 |
| UP-int-alg[2] | 0x06 |
| | |

[2] It is applied when using relay nodes. As relay is out of the scope of this document, user plane integrity algorithms are not discussed herein.

### ③ [MME] Generating NAS-MAC for integrity protection

The MME forms a **Security Mode Command** message to send to the UE and calculates **NAS-MAC** (Message Authentication Code for NAS for Integrity) using the selected EIA algorithm (EIA1) with input parameters such as the **Security Mode Command** message and $K_{NASint}$ derived in ②. Figure 3 shows how **NAS-MAC** is calculated using the following EIA algorithm input parameters [2]:

- Count: 32-bit downlink NAS count
- Message: NAS message, i.e., **Security Mode Command** message herein
- Direction: 1-bit direction of the transmission, 0 for uplink and 1 for downlink (set to 1 herein)
- Bearer[3]: 5-bit bearer ID, constant value (set to 0)
- $K_{NASint}$: 128-bit NAS integrity key

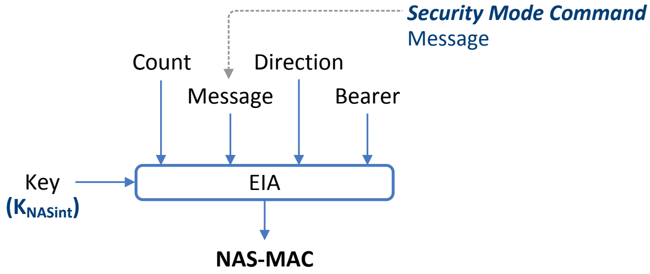



**Figure 3. Calculation of NAS-MAC [2]**

### ④ [UE ← MME] Sending a Security Mode Command message

The MME attaches the **NAS-MAC** calculated in ③ to the **Security Mode Command** message and sends it to the UE. Here the message is integrity protected but not ciphered. Message parameters used are as follows:

- $KSI_{ASME}$: 3-bit value associated with a $K_{ASME}$, used to identify the $K_{ASME}$ ($KSI_{ASME}$=1 herein)
- Replayed UE Security Capability: UE Security Capability included in the UE Network Capability in the **Attach Request** message sent by UE, indicates which security algorithms are supported by the UE
- NAS Ciphering Algorithm: NAS ciphering algorithm selected by the MME, EEA1 herein
- NAS Integrity Algorithm: NAS integrity algorithm selected by the MME, EIA1 herein

### ⑤ [UE] Setting $K_{ASME}$ identifier ($KSI_{ASME}$)

When the UE receives a **Security Mode Command** message from the MME, it sets $KSI_{ASME}$ in the message as its $KSI_{ASME}$ and uses it as an identifier of the current $K_{ASME}$.

### ⑥ [UE] Deriving NAS security keys

The UE, recognizing the NAS security algorithm that the MME selected, derives $K_{NASint}$ and $K_{NASenc}$ from $K_{ASME}$ using the algorithm IDs and the algorithm distinguishers (see Table 1).

[3] As there is only one NAS signaling connection between a UE and an MME, technically no bearer is needed. However, it was included here so that the same input parameters are used in calculating both NAS MAC (NAS-MAC) and AS MAC (MAC-I).

### ❼ [UE] Verifying the integrity of the Security Mode Command message

The UE checks the integrity of the received **Security Mode Command** message by verifying the **NAS-MAC** included in the message. It recognizes the NAS integrity algorithm selected by the MME is EIA1 and calculates **XNAS-MAC**, a message authentication code, by using the selected EIA1 algorithm with the **Security Mode Command** message and $K_{NASint}$ derived in ❻. Figure 4 shows how **XNAS-MAC** is calculated using the same EIA input parameters as in ❸ [2]. The UE verifies the integrity of the message by examining whether the **XNAS-MAC** calculated by itself matches the **NAS-MAC** calculated by the MME. If they match, it is guaranteed that the **Security Mode Command** message has not been manipulated (e.g., inserted or replaced) on the way.

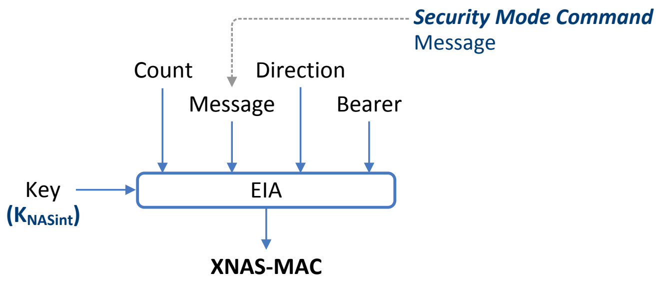



**Figure 4. Calculation of XNAS-MAC [2]**

### (2) Delivering a Security Mode Complete message

Figure 5 illustrates how a **Security Mode Complete** message is delivered during the NAS security setup procedure. The UE, by sending a **Security Mode Complete** message to the MME, informs the MME that the same NAS security keys as MME's are derived in the UE and that the integrity of the **Security Mode Command** message is verified. The **Security Mode Complete** message is ciphered and integrity protected for transmission.

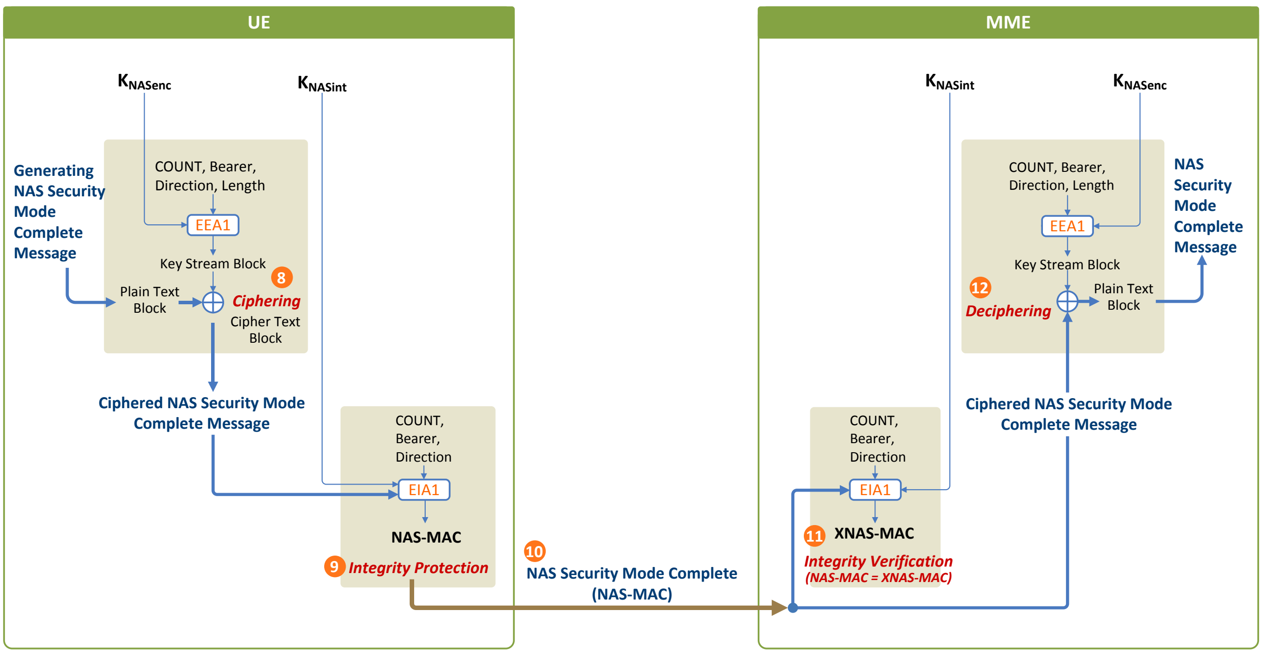



**Figure 5. NAS security setup: Delivery of a Security Mode Complete message**

## 8 [UE] Encrypting the message using the selected encryption algorithm (EEA1)

The UE forms and encrypts the **Security Mode Complete** message to be sent to the MME. The ciphered **Security Mode Complete** message (Cipher Text Block) is derived by performing bitwise XOR between the **Security Mode Complete** message (Plane Text Block) and the encryption key stream (Key Stream Block) generated using EEA1 algorithm with NAS encryption key ($K_{NASenc}$). Figure 6 shows how NAS messages are encrypted [2]. EEA algorithm input parameters used to generate the key stream block are as follows:

- Count: 32-bit uplink NAS count
- Bearer: 5-bit bearer ID, constant value (set to 0)
- Direction: 1-bit direction of the transmission, 0 for uplink and 1 for downlink (set to 0 herein)
- Length: the length of the key stream to be generated by the encryption algorithm
- $K_{NASenc}$: 128-bit NAS cipher key

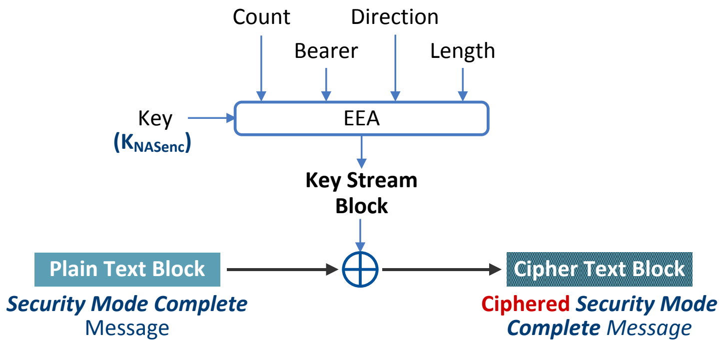



**Figure 6. Encryption of NAS message by the sender (UE) [2]**

## 9 [UE] Generating NAS-MAC for integrity protection

The UE calculates **NAS-MAC** using EIA algorithm (EIA1) with the ciphered **Security Mode Complete** message and $K_{NASint}$. Figure 3a shows how **NAS-MAC** is calculated using the following EIA algorithm input parameters:

- Count: 32-bit uplink NAS count
- Message: NAS message, **Security Mode Complete** message herein
- Direction: 1-bit direction of the transmission, 0 for uplink and 1 for downlink (set to 0 herein)
- Bearer: 5-bit bearer ID, constant value (set to 0)
- $K_{NASint}$: 128-bit NAS integrity key

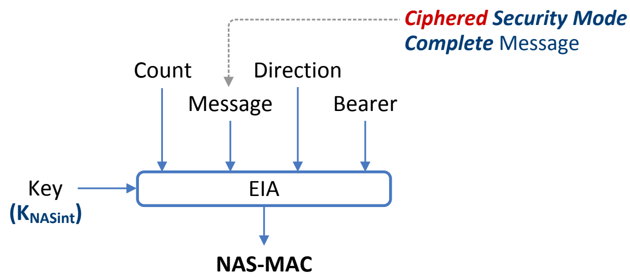



**Figure 3a. Calculation of NAS-MAC for the Ciphered Security Mode Complete message**

### ⑩ [UE → MME] Sending the Security Mode Complete message

The UE attaches the **NAS-MAC** calculated in ⑨ to the **Security Mode Complete** message and sends it to the MME. Here the message is integrity protected and ciphered, and all the NAS messages that the UE sends to the MME hereafter are securely delivered.

### ⑪ [MME] Verifying the Integrity of the Security Mode Complete message

The MME checks the integrity of the received **Security Mode Complete** message by verifying **NAS-MAC** included in the message. MME calculates **XNAS-MAC**, a message authentication code, by using the selected EIA1 algorithm with the **Security Mode Complete** message and $K_{NASint.}$ Figure 4a shows how **XNAS-MAC** is calculated using the same EIA input parameters as in ⑨. The MME verifies the integrity of the message by examining whether the **XNAS-MAC** calculated by itself matches the **NAS-MAC** calculated by the UE. If they match, it is guaranteed that the **Security Mode Complete** message has not been manipulated on the way.

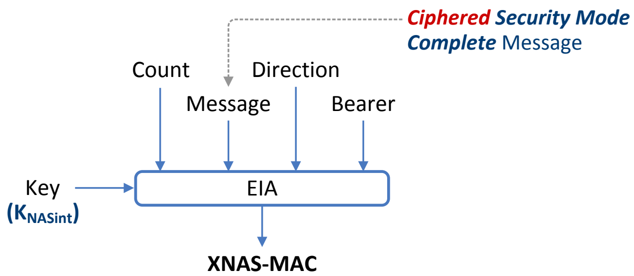



**Figure 4a. Calculation of XNAS-MAC for the Ciphered Security Mode Complete message**

### ⑫ [MME] Decrypting of the Security Mode Complete message

After successful verification of the **Security Mode Complete** message, the MME decrypts the message using EEA algorithm (EEA1). Then the **Security Mode Complete** message, the original message generated by the UE, is derived through XOR between the ciphered **Security Command Complete** message and the key stream block. Figure 7 illustrates how the message is decrypted using the same EEA algorithm input parameters as in ⑧.

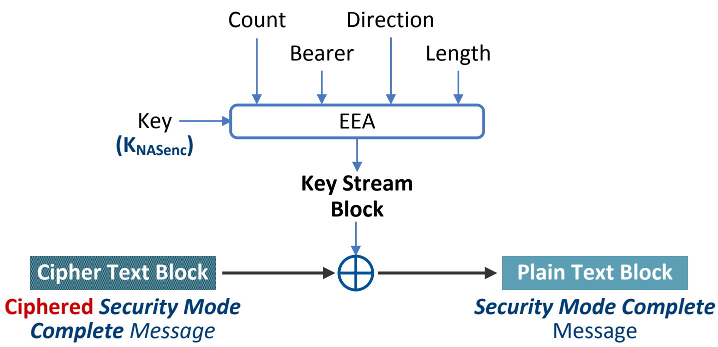



**Figure 7. Decryption of the NAS message by the receiver (MME) [2]**

## 2.2 After NAS Security Setup

Once the NAS security setup is completed as in Section 2.1, all the NAS messages between the UE and the MME thereafter are encrypted and integrity protected before being sent. Figure 8 shows how NAS messages are delivered between the UE and the MME after the NAS security setup.

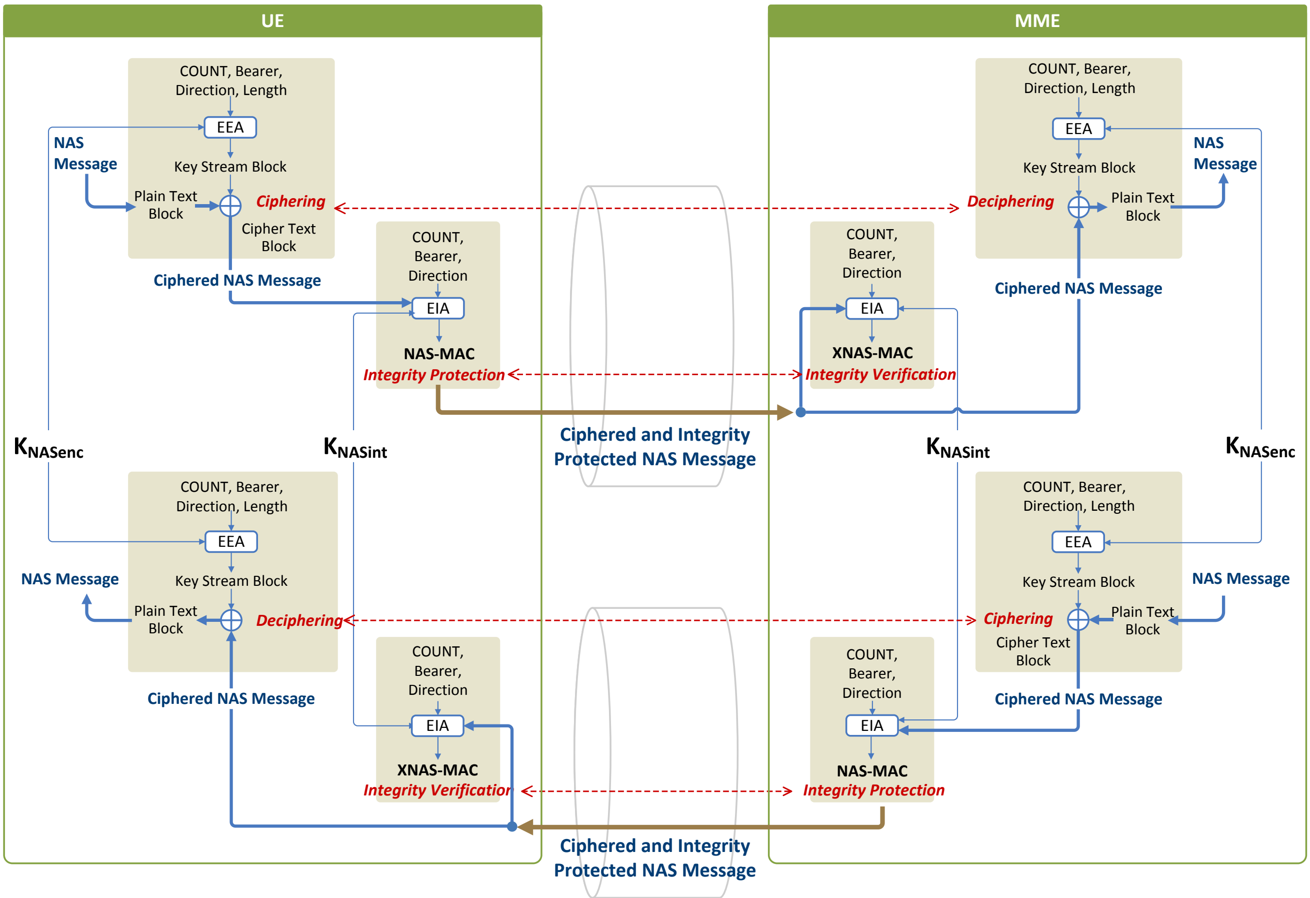
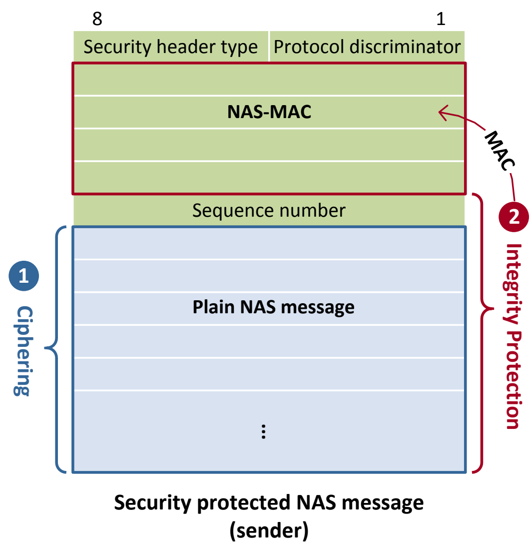



**Figure 8. Ciphering and integrity protection of the NAS Messages after the NAS security setup**

When NAS messages are being sent, they are encrypted first and then integrity protected before being sent. The original NAS messages are first encrypted using an encryption key ($K_{NASenc}$) and then integrity protected by including **NAS-MAC** calculated using an integrity key ($K_{NASint}$) so that the messages are delivered as encrypted and integrity protected.

When received, however, the NAS messages are integrity verified first and then decrypted, which is in the opposite order of what has been done when they were sent. That is, the integrity of the NAS messages is verified first by comparing the **XNAS-MAC** calculated using the integrity key ($K_{NASint}$) and the received **NAS-MAC**, and then the messages are decrypted to get the original NAS messages.



## III. AS Security

A detailed description of the AS security previously mentioned in LTE Security I [1] will be given below. An AS security setup procedure consists of RRC signaling, between a UE and an eNB, by a **Security Mode Command** message that the eNB sends to the UE and a **Security Mode Complete** message that the UE sends to the eNB. Descriptions of the AS security setup procedure by RRC signaling and how RRC messages in the control plane and IP packets in the user plane are transmitted thereafter will be provided in Sections 3.1 and 3.2, respectively.

### 3.1 AS Security Setup

**(1)** shows how a **Security Mode Command** message is delivered and **(2)** demonstrates how a **Security Mode Complete** message is delivered.

**(1) Delivering a Security Mode Command message**

Figure 9 and 10 are illustrations of how a **Security Mode Command** message is delivered during the AS security setup procedure. The Figures show how the message is processed at the eNB and at the UE, respectively. First, Figure 9 shows how the eNB derives AS security keys and delivers the **Security Mode Command** message to the UE. $K_{eNB}$, an AS security base key, is derived from $K_{ASME}$ and the eNB derives AS security keys from $K_{eNB}$. Since $K_{ASME}$ is not delivered to the eNB, the MME derives $K_{eNB}$ from $K_{ASME}$ and forwards it to the eNB, which then derives AS security keys based on the forwarded $K_{eNB}$.

1 and 2 show the LTE authentication procedure (see [1] for the detail operation).

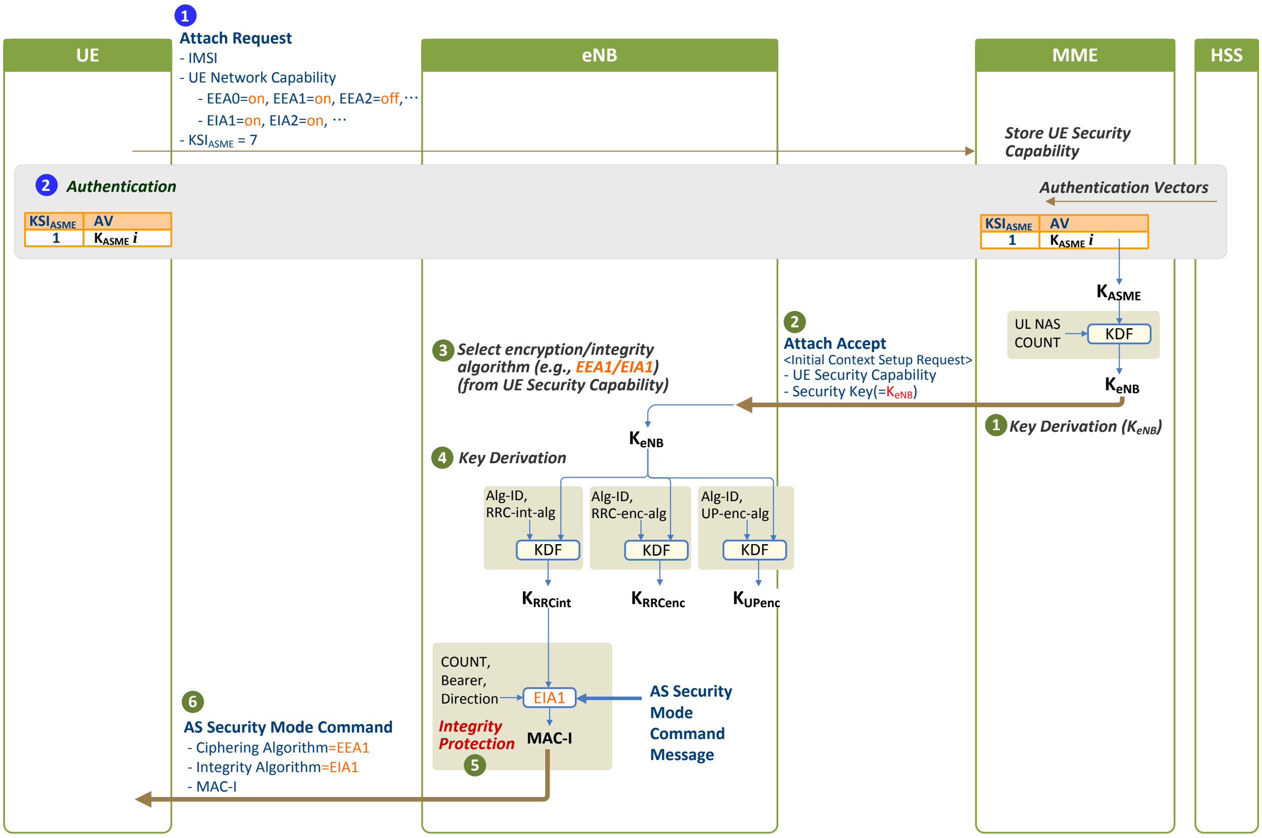



**Figure 9. AS security setup: Generating and sending a Security Mode Command message**

**❶ [MME] Deriving $K_{eNB}$**

The MME derives $K_{eNB}$ using a key derivation function with $K_{ASME}$ and UL NAS Count.

**❷ [eNB ← MME] Forwarding $K_{eNB}$**

The MME forwards the **Attach Accept** message to the UE as a response to the **Attach Request** message in blue ❶. This NAS message is delivered through an **Initial Context Setup Request** message, an S1 signaling message between the eNB and the MME. Message parameters used are as follows:

- UE Security Capability: security algorithms selected by the MME in the UE Network Capability in the **Attach Request** message sent by the UE
- Security Key: 256-bit $K_{eNB}$

**❸ [eNB] Selecting security algorithms**

The eNB selects ciphering and integrity algorithms to be applied to RRC messages and IP packets based on the UE Security Capability information included in the received **Initial Context Setup Request** message from the MME. Figure 9 shows an example of selecting EEA1 for an encryption algorithm and EIA1 for an integrity algorithm.

**❹ [eNB] Deriving AS Security Keys**

The eNB derives $K_{RRCint}$, $K_{RRCenc}$ and $K_{UPenc}$ from $K_{eNB}$ using the algorithm IDs and algorithm distinguishers of the selected security algorithms (see Table 1).

- $K_{RRCint}$ = KDF($K_{eNB}$, RRC-int-alg, Alg-ID)
- $K_{RRCenc}$ = KDF($K_{eNB}$, RRC-enc-alg, Alg-ID)
- $K_{UPenc}$ = KDF($K_{eNB}$, UP-enc-alg, Alg-ID)

**❺ [eNB] Generating MAC-I for integrity protection**

The eNB forms a **Security Mode Command** message to send to the UE and calculates **MAC-I** (Message Authentication Code for Integrity) using the selected EIA algorithm (EIA1) with $K_{RRCint}$ derived in ❹. Calculation of **MAC-I** is illustrated in Figure 3 and the EIA input parameters used are as follows:

- Count: 32-bit downlink PDCP count
- Message: RRC message, i.e., **Security Mode Command** message herein
- Direction: 1-bit direction of the transmission, 0 for uplink and 1 for downlink (set to 1 herein)
- Bearer: 5-bit radio bearer ID
- $K_{NASint}$: 128-bit AS integrity key

**❻ [UE ← eNB] Sending Security Mode Command message**

The eNB attaches the **MAC-I** calculated in ❺ to the **Security Mode Command** message and sends it to the UE. Here the message is integrity protected but not ciphered. Message parameters used are as follows:

- AS Ciphering Algorithm: AS ciphering algorithm selected by eNB, EEA1 herein
- AS Integrity Algorithm: AS integrity algorithm selected by eNB, EIA1 herein

Figure 10 shows how the UE derives AS keys from the **Security Mode Command** message received from the eNB and verifies the integrity of the message.

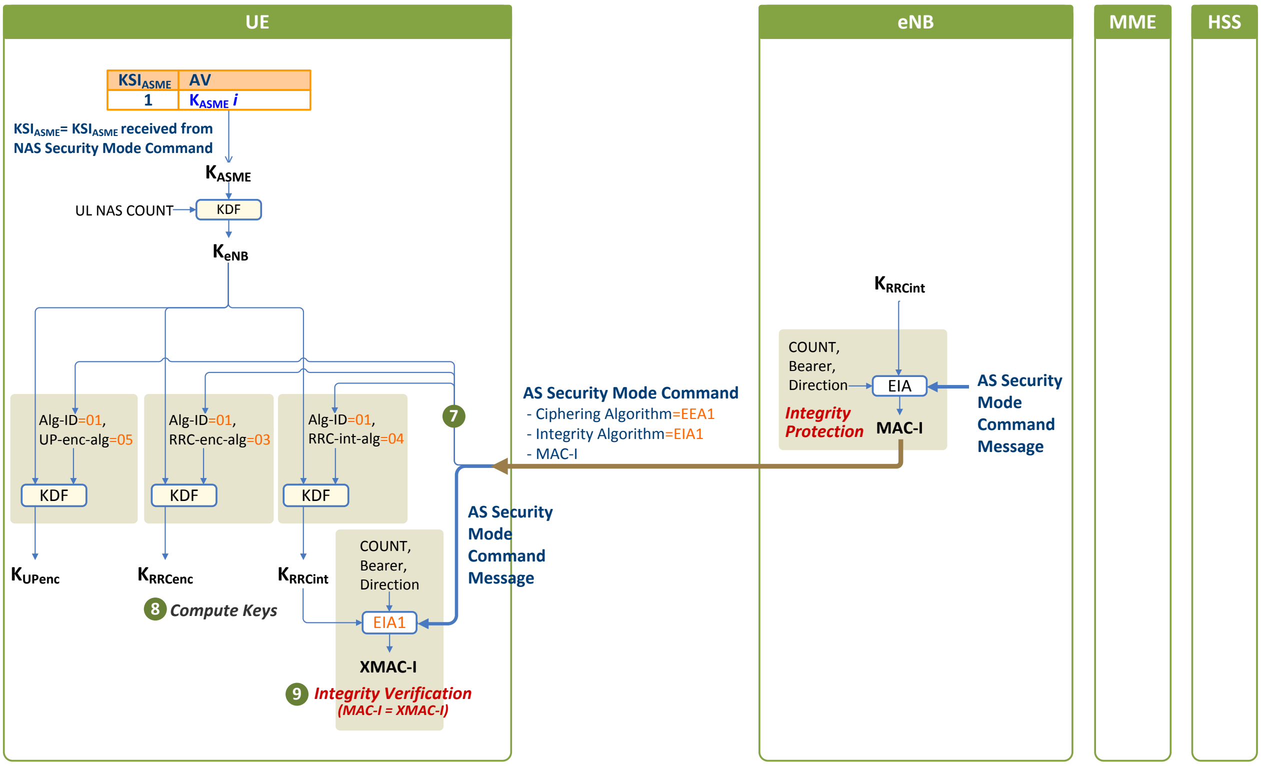



**Figure 10. AS security setup: Receiving a Security Mode Command message**

## ❼ [UE] Identifying security algorithms: EEA1, EIA1

The UE identifies which AS encryption and integrity algorithms are selected by the eNB when it receives the **Security Mode Command** message from the eNB. Figure 10 shows an example of selecting EEA1 and EIA1.

## ❽ [UE] Deriving AS security keys

The UE derives $K_{RRCint}$, $K_{RRCenc}$ and $K_{UPenc}$ from $K_{eNB}$ using the algorithm IDs and the algorithm distinguishers of the identified security algorithms (see Table 1).

## ❾ [UE] Verifying the integrity of the Security Mode Command message

The UE verifies the **MAC-I** included in the **Security Mode Command** message using the integrity key ($K_{RRCint}$) derived in ❽. During this verification, it is checked whether the **XMAC-I** calculated by UE matches the **MAC-I** calculated by the eNB. If they match, it is guaranteed that the **Security Mode Command** message has not been manipulated on the way. Calculation of **XMAC-I** is illustrated in Figure 4 and the same EIA input parameters used in ❺ are used.

**(2) Delivering a Security Mode Complete message**

Figure 11 shows how a **Security Mode Complete** message is delivered during the AS security setup procedure. The UE, by sending the **Security Mode Complete** message to the eNB, informs the eNB that the same AS security keys as eNB's are derived in the UE and that the integrity of the **Security Mode Command** message is verified. Now, the **Security Mode Complete** message is delivered as integrity protected.

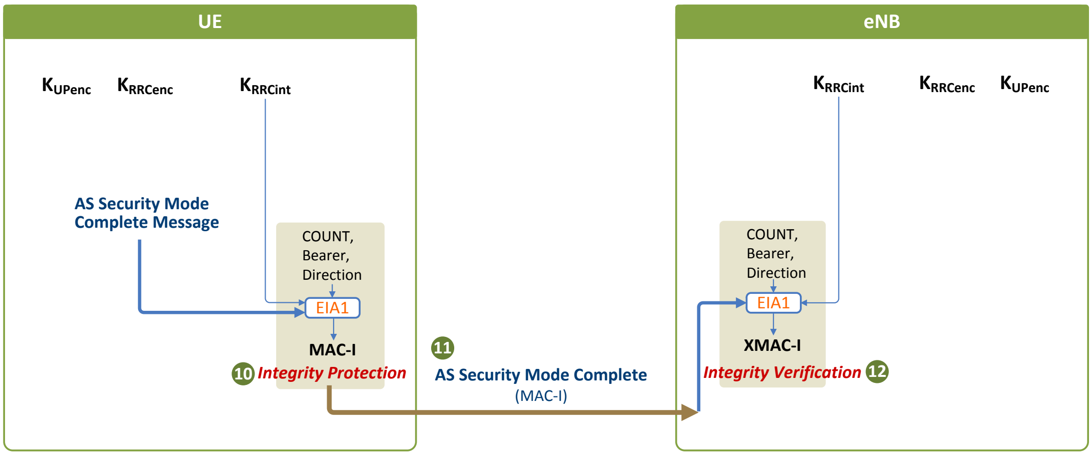



**Figure 11. AS security setup: Delivery of a Security Mode Complete message**

**⑩ [UE] Generating MAC for integrity protection**

The UE calculates **MAC-I** using EIA algorithm (EIA1) with the **Security Mode Complete** message and $K_{RRCint}$. Calculation of **MAC-I** is illustrated in Figure 3 and the same EIA input parameters used in ⑤ are used.

**⑪ [UE → eNB] Sending the Security Mode Complete message**

The UE attaches the **MAC-I** calculated in ⑩ to the **Security Mode Complete** message and sends it to the eNB. Here the message is integrity protected.

**⑫ [eNB] Verifying the integrity of the Security Mode Complete message**

The eNB checks the integrity of the received **Security Mode Complete** message by verifying the **MAC-I** included in the message. The eNB calculates **XMAC-I**, a message authentication code, by using the selected EIA1 algorithm with the **Security Mode Complete** message and $K_{RRCint}$. The eNB verifies the integrity of the message by examining whether the **XMAC-I** calculated by itself matches the **MAC-I** calculated by the UE. If they match, it is guaranteed that the **Security Mode Complete** message has not been manipulated on the way.

### 3.2 After AS Security Setup

Once the AS security setup is completed as in Section 3.1, all the RRC messages delivered between UE and eNB thereafter are integrity protected and encrypted and all the IP packets are encrypted before being sent. Figure 12 shows how RRC messages and IP packets are delivered between the UE and the eNB after the AS Security setup.

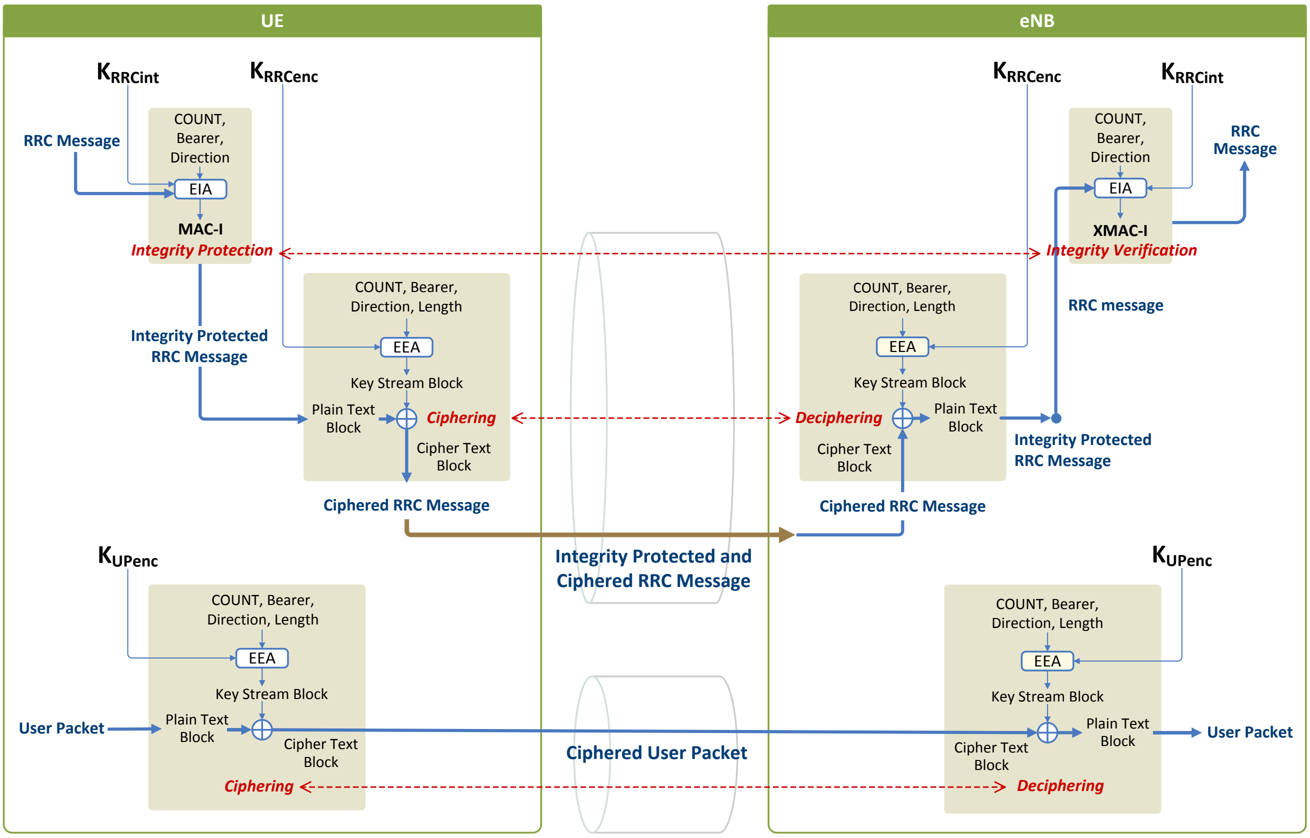
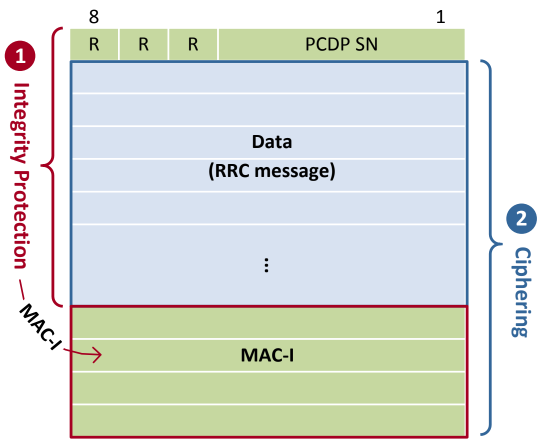



**Figure 12. Integrity protection and ciphering of RRC messages and ciphering of user packets after the AS security setup**

When RRC messages are being sent, they are integrity protected first and then encrypted before being sent, unlike NAS messages were. The original RRC messages are first integrity protected including **MAC-I** calculated using the integrity key ($K_{RRCint}$) and then they are encrypted using the encryption key ($K_{RRCenc}$). That way, the messages are delivered as integrity protected and encrypted.




**Security protected RRC message (sender)**

When received, however, RRC messages are decrypted first and then integrity verified, which is in the opposite order of what has been done when they were sent. That is, the messages are decrypted first using $K_{RRCenc}$ to get the integrity protected RRC messages, and then the integrity of the RRC messages is verified by comparing the **XMAC-I** calculated using the integrity key ($K_{RRCint}$) and the received **MAC-I** to confirm the original RRC messages.

User packets are encrypted but not integrity protected. The user packets encrypted by a sender using the encryption key ($K_{UPenc}$) are decrypted by the receiver using the same encryption key ($K_{UPenc}$) to get the original user packets.

## IV. Security Context

So far, we have discussed the LTE authentication procedure (in LTE Security I [1]) and NAS security setup and AS security setup procedures (in Chapter II and Chapter III herein). Data relating to security that has been set in the EPS entities during these procedures is called an EPS security context, which can be either a NAS security context or an As security context. A NAS security context can be one of the two types, "full native" or "partial native". A NAS security context is called as "partial native" after EPS AKA is performed and before the first SMC (Security Mode Command) procedure begins. A partial native EPS NAS security context is transformed into a full native after the SMC procedure is completed. Table 2 lists these EPS security contexts[4].

**Table 2. EPS security contexts**

| Partial Native EPS NAS Security Context | Full Native EPS NAS Security Context | EPS AS Security Context |
|---|---|---|
| UE Security Capability | UE Security Capability | UE Security Capability |
| $K_{ASME}$ | $K_{ASME}$ | $K_{eNB}$ |
| $KSI_{ASME}$ | $KSI_{ASME}$ | |
| UL NAS Count | UL NAS Count | UL NAS Count |
| DL NAS Count | DL NAS Count | DL NAS Count |
| | EIA ID | EIA ID |
| | EEA ID | EEA ID |
| | $K_{NASint}$ | $K_{RRCint}$ |
| | $K_{NASenc}$ | $K_{RRCenc}$ |
| | | $K_{UPenc}$ |

Figure 13 displays the key LTE security data stored in EPS entities as a result of the EPS AKA and NAS/AS security setup procedures. It shows how each security data is generated (e.g. provisioning, calculated by itself) and the data transfer flow (→) indicating from which data each security data is delivered.

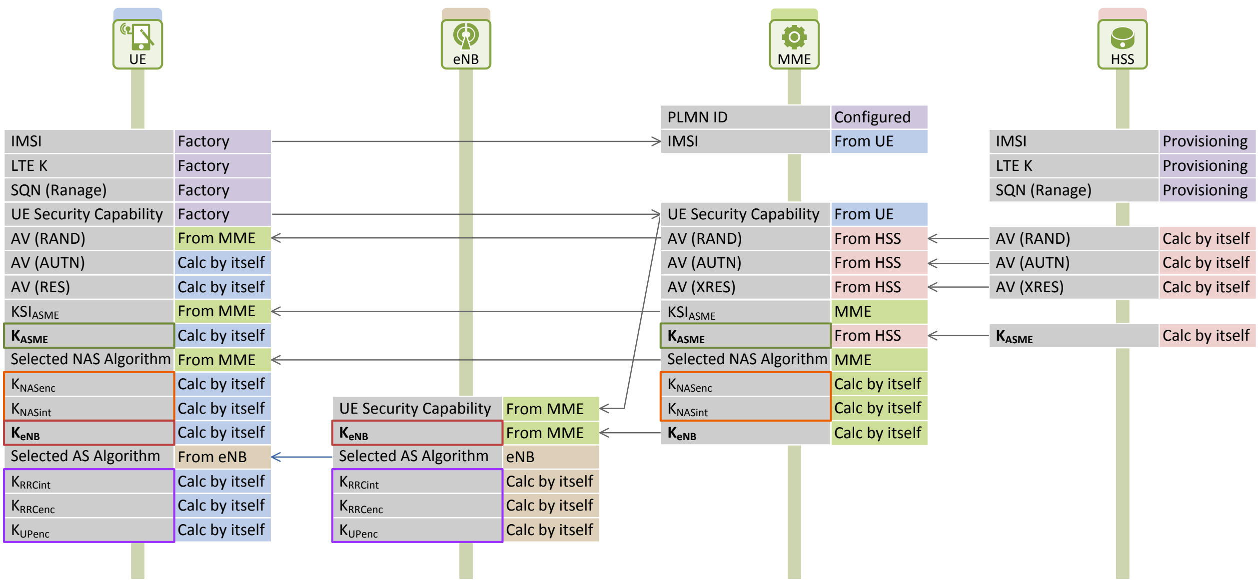



**Figure 13. Security data in EPS entities**

4 As handover security is beyond the scope of this document, handover-related data (NH, NCC, $K_{eNB}$*) are not included herein.

## V. Closing

In the LTE Security technical documents, i.e., LTE Security I [1] and LTE Security II, we have covered some of the key LTE security technologies, including EPS AKA-based LTE authentication, NAS and AS setup procedures, and security data in EPS entities. We have learned that LTE security keys have their own hierarchy, which are separated and used for different purpose. The top-level key is K, an LTE key, and it has a permanent value stored in USIM and HSS (AuC). From this K, CK and IK are derived and then $K_{ASME}$ is derived from CK and IK. NAS keys ($K_{NASint}$, $K_{NASenc}$) and $K_{eNB}$ are derived from $K_{ASME}$. And from $K_{eNB}$, AS security keys ($K_{RRCint}$, $K_{RRCenc}$, $K_{UPenc}$) are derived. We have also found that different keys are derived from a UE, eNB or MME depending on whether they are intended for the NAS level or the AS level, for the control plane or the user plane, and for ciphering or integrity check, or which algorithms are used. Table 3 lists all the LTE securities keys that have covered so far.

**Table 3. LTE security keys**

| Key | Length | Location | Derived from | Description |
|---|---|---|---|---|
| K | 128 bits | USIM, HSS/AuC | - | EPS master key |
| CK | 128 bits | USIM, HSS/AuC | K | Cipher key |
| IK | 128 bits | USIM, HSS/AuC | K | Integrity key |
| $K_{ASME}$ | 256 bits | UE, MME, HSS | CK, IK | MME base key |
| $K_{eNB}$ | 256 bits | UE, eNB, MME | $K_{ASME}$ | eNB base key |
| $K_{NASint}$ | 128/256 bits | UE, MME | $K_{ASME}$ | Integrity key for NAS messages |
| $K_{NASenc}$ | 128/256 bits | UE, MME | $K_{ASME}$ | Encryption key for NAS messages |
| $K_{RRCint}$ | 128/256 bits | UE, eNB | $K_{eNB}$ | Integrity key for RRC messages on SRB |
| $K_{RRCenc}$ | 128/256 bits | UE, eNB | $K_{eNB}$ | Encryption key for RRC messages on SRB |
| $K_{UPenc}$ | 128/256 bits | UE, eNB | $K_{eNB}$ | Encryption key for user packets on DRB |

## References

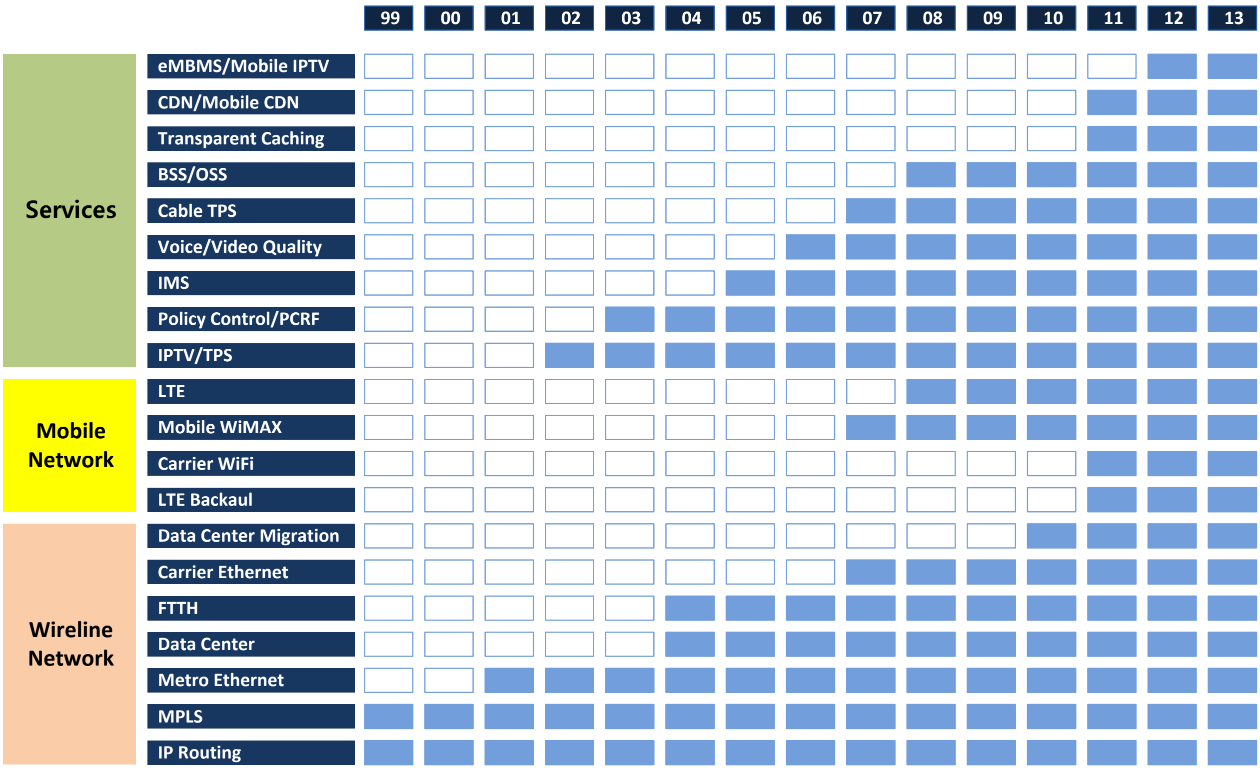
[1] Netmanias Technical Document, "LTE Security I: LTE Security Concept and LTE Authentication", July 2013, http://www.netmanias.com/en/?m=view&id=techdocs&no=5902

[2] 3GPP TS 33.401, "3GPP System Architecture Evolution (SAE); Security architecture".

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

[4] NMC Consulting Group Confidential Internal Report, "E2E LTE Network Design", August 2010.

## Netmanias Research and Consulting Scope




**Visit http://www.netmanias.com to view and download more technical documents.**

**About NMC Consulting Group**

NMC Consulting Group is an advanced and professional network consulting company, specializing in IP network areas (e.g., FTTH, Metro Ethernet and IP/MPLS), service areas (e.g., IPTV, IMS and CDN), and wireless network areas (e.g., Mobile WiMAX, LTE and Wi-Fi) since 2002.