

Source: Nokia, Siemens Networks
Title: Key refresh in SAE/LTE
Agenda item: SAE/LTE security
Document for: Discussion and decision

1 Introduction

In SAE/LTE we have encryption and integrity protection for NAS and RRC, and encryption for UP. Encryption and integrity protection relies on cryptographic algorithms, cryptographic keys, and on counters or sequence numbers. When the same algorithm, key and counter are used with different messages then an attacker could exploit this fact to break security, at least in one of the following two ways:

- For integrity protection, an attacker could replace one message with the other, and the receiver would not be able to notice it.
- For confidentiality protection with a stream cipher, the key stream will be the same, and the attacker can obtain the XOR of the two messages, which gives him a strong advantage in finding out the cleartext messages.

Any solution to this problem must ensure that the solution is efficiently embedded in other procedures of the mobile communication system.

In UMTS, the problem is addressed by using the parameters START and FRESH, cf. TS 33.102. The parameter START serves two purposes:

A) to ensure that ciphering and integrity key streams do not repeat (by using START to initialize HFN/SN)

B) to count how much traffic has been encrypted / integrity protected with the same keys so that these keys can be changed when a certain THRESHOLD is reached.

We propose to decouple the mechanisms for achieving A) and B) in LTE because

- in LTE there are three different protocols to be protected (RRC, UP and NAS) with two different network termination points with very different traffic characteristics;
- monitoring of counter values would have to be done for each protocol separately;
- key refresh procedures would have to be run separately for every protocol, or the counter values would have to be aggregated into a START value somehow in the UE and in a single network entity;
- START value(s) would have to be transmitted from the UE to the relevant network entities.
- hence, a coupling of solutions to A) and B) may complicate the solutions for SAE/LTE.
- As Rel-99 USIMs shall be re-usable for LTE, additional START values and THRESHOLD values for LTE cannot be stored on the USIM. START values and THRESHOLD values may be stored in non-volatile memory on the ME. As a consequence an extra THRESHOLD value for LTE cannot be set by the home operator and distributed with the USIM or managed on the USIM over the air. But the THRESHOLD value on the USIM for UMTS may not be suitable for LTE. A THRESHOLD value stored in the ME would have to be standardised or configured into the ME by the operator, both of which may be difficult. Therefore UE control of key change due to amount of traffic may be difficult to manage.

We do not deal with a solution to problem B here. But we would like to point out that, when problem B is decoupled from problem A, there is no need any more for keeping any counters synchronised between UE and network for the purposes of solving B), so the solution may be easier to achieve than in UMTS. Given the difficulty for the home operator to configure a THRESHOLD value in the UE, as explained above, it may even be worth thinking about controlling key usage only on the network side, which may still provide a good level of security. (The UE has to trust the visited network in many respects anyhow) There may still be a need for counters related to limiting key usage, e.g.

to limit the number of key derivations from one parent key. However, it will be possible to store these counters on the network side only, as in case of changes of network entities, keys are changed as well.

We propose to solve problem A) within SAE/LTE by using fresh keys for RRC, UP and NAS after handover, state transitions from detach to active, or idle to active, and idle mode mobility. Whenever a fresh key is used for the first time, all counters pertaining to the related security association are reset to zero. As a consequence, counter values do not have to be stored and reused after handover, state transitions and idle mode mobility. In our proposal, if a key is to be refreshed, the UE and a suitable network entity each generate a nonce and derive a fresh key with the help of these nonces and a suitable parent key. This ensures that fresh keys depend only on their parent key but not on old keys of the same hierarchy level. In the rest of the document we make detailed proposals for how UE and the network entities can exchange nonces upon handover, idle mode mobility and state transitions.

2 Proposal

2.1 Input parameters to key derivation functions

Throughout the rest of the document we assume the following input parameters for the different key derivation functions (KDF) for the keys in the SAE/LTE key hierarchy:

- 1) $K_{UPenc} = \text{KDF}(K_{eNB} \parallel \text{"UP encryption"} \parallel [\text{other}])$
- 2) $K_{RRCenc} = \text{KDF}(K_{eNB} \parallel \text{"RRC encryption"} \parallel [\text{other}])$
- 3) $K_{RRCint} = \text{KDF}(K_{eNB} \parallel \text{"RRC integrity"} \parallel [\text{other}])$
- 4) $K_{eNB} = \text{KDF}(K_{ASME} \parallel eNB\text{ Id} \parallel \text{Nonce}_{UE} \parallel \text{Nonces}_{SN} \parallel [\text{other}])$
- 5) $K_{NASenc} = \text{KDF}(K_{ASME} \parallel \text{Nonce}_{UE} \parallel \text{Nonces}_{SN} \parallel \text{"NAS encryption"} \parallel [\text{other}])$
- 6) $K_{NASint} = \text{KDF}(K_{ASME} \parallel \text{Nonce}_{UE} \parallel \text{Nonces}_{SN} \parallel \text{"NAS integrity"} \parallel [\text{other}])$
- 7) $K_{ASME} = \text{KDF}(CK \parallel IK \parallel PLMN\text{-Id} \parallel \text{"LTE access"} \parallel [\text{other}])$

Note that the input parameters shown in the key derivation functions here only cover the parameters relevant for the purpose of key refresh. Additional parameters may have to be added for other purposes as discussed in Section 7.4.7 or TR 33.821 (e.g. other node identifiers or algorithm identifiers...)

For the keys, K_{UPenc} , K_{RRCenc} , K_{RRCint} , a variant of the key derivation functions in a handover situation which would make the key derivation process in handover slightly more efficient but would require different key derivation functions for the above keys on handover compared to state transitions. . This is discussed further below.

2.2 Key refresh on state transitions

According to TR 25.813, (Section 5.5.2) or TS 36.300 (Annex 2) a security context is established during the two state transitions detached to active and idle to active. State transitions from detached to idle are not currently considered by the RAN groups. We therefore do not consider detached to idle transitions in this section.

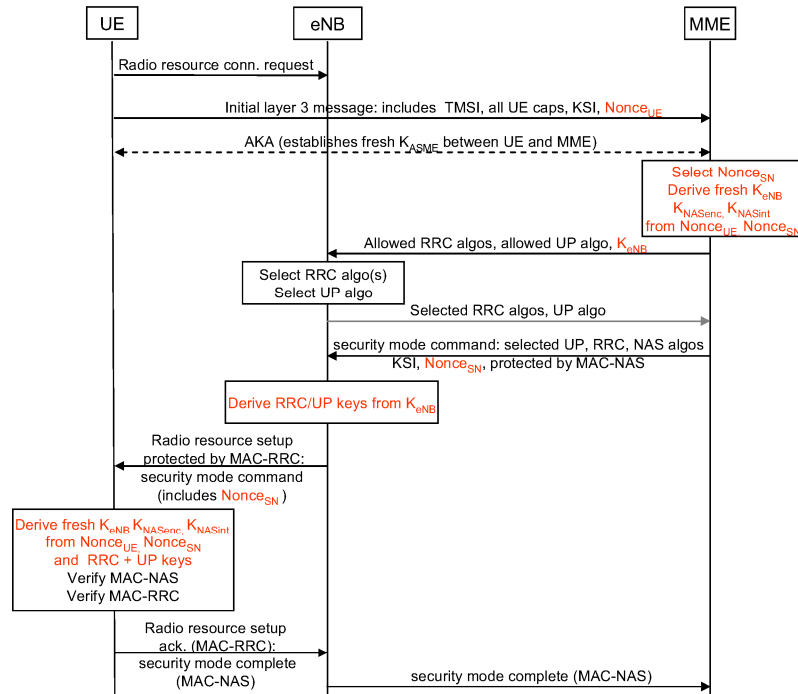
A side remark: According to TR 25.813 (v. 7.10, Section 5.5.2), a new AKA run is performed on detached to active transitions, while on idle to active transitions no AKA occurs. But AKA is not strictly required if the key K_{ASME} (cf. TR 33.821, section 7.4) is still available in UE and MME. This is the case when the UE is in idle mode. We think that the decision when to run AKA should be left to the operator as much as possible. Therefore, in idle to active transition, AKA should be optional. In order to stay future proof in case of changes to the above current assumptions, we here consider transitions from idle to active with an optional AKA run.

2.2.1 Key refresh on idle to active transition

We propose that, on idle to active transitions without AKA, K_{NASenc} , K_{NASint} , K_{eNB} , K_{RRCenc} , K_{RRCint} , K_{UPenc} are refreshed with the help of an exchange of nonces between UE and MME. In case MME requests a new AKA during the idle to

active transition, K_{ASME} is additionally refreshed by the AKA. When K_{ASME} is refreshed then, strictly speaking, the nonces would not be needed to obtain fresh lower layer keys, but we propose to use them also in this case to have a uniform handling of key derivation. In both cases, the $Nonce_{UE}$ is selected by UE and included in the initial layer 3 message. The $Nonce_{SN}$ is generated by MME and included in the security mode command message sent to UE via eNB. As the security mode command is already protected by the new NAS integrity key derived using both nonces, there is no need to include $Nonce_{UE}$ in the security mode command for confirmation.

In the following Figure 1 the key refresh procedure is detailed based on one of the security mode command procedures (Alternative 5) included in the current version of TR 33.821 and suggested as the best current alternative in a companion contribution. The same key refresh procedure could, however, also be used if another SMC procedure was selected.



- **Figure 1: Key re-fresh on idle to active** UE includes $Nonce_{UE}$ in the initial layer 3 message. This message is protected with the old NAS keys
- MME optionally requests a new run of AKA to refresh K_{ASME}
- MME selects $Nonce_{SN}$ and includes it in security mode command. MME derives K_{eNB} , K_{NASenc} , and K_{NASint} from $Nonce_{UE}$, $Nonce_{SN}$ and the K_{ASME} generated during the last AKA run, as described in 4), 5) and 6) of section 2.1.
- MME transfers K_{eNB} to eNB and includes $Nonce_{SN}$ in the security mode command sent to eNB. The security mode command is integrity protected with the new NAS key (NAS-MAC).
- eNB derives the RRC and UP keys from the received fresh K_{eNB} , as in 1), 2), 3) of section 2.1.
- eNB includes the security mode command message in the radio resource setup message. This message is protected with the new RRC integrity key (RRC-MAC).
- UE derives K_{NASenc} , K_{NASint} , and K_{eNB} from K_{ASME} , $Nonce_{UE}$, and $Nonce_{SN}$ as in 4), 5), and 6) of section 2.1. UE derives K_{RRCenc} , K_{RRCint} , K_{UPenc} from the fresh K_{eNB} according to 1), 2), and 3) of section 2.1 respectively

2.2.2 Key refresh on detached to active transition

The current assumption in TR 23.813 (Section 5.5.2) is that in detached mode no security context is stored. As a consequence, on detached to active transitions, a new authentication and key agreement has to take place. In this case,

the initial layer 3 message is sent unprotected but the rest of the key derivation is handled in the same way as on idle to active transitions with AKA.

2.3 Key refresh on idle mode mobility

From the discussions in RAN and SA2 it is currently not clear whether an RRC connection has to be setup for TA registration requests (e.g. due to idle mode mobility). But this is irrelevant for our discussion, as TA registration requests are sufficiently integrity protected by the NAS keys only (see TR 25.813, Section 5.4.1).

According to TR 23.882, V.18.0, Section 7.7 a new AKA run may optionally be executed after idle mode mobility. If no new authentication takes place the previously used K_{ASME} is transferred from the old to the new MME and reused after idle mode mobility, possibly after some transformation. If authentication takes place, a new K_{ASME} is agreed upon between UE and the new MME during AKA. In both cases, the target MME has a K_{ASME} available. Fresh NAS keys are derived with the help of nonces exchanged between UE and the new MME. Figure 2 illustrates the procedure:

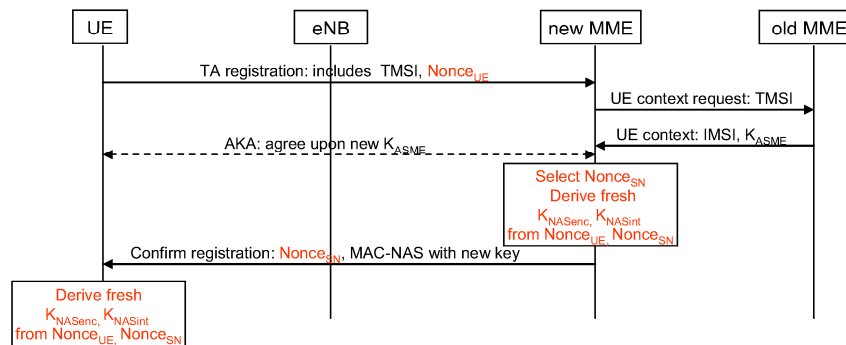


Figure 2: Key re-fresh on idle mode mobility

- UE includes $Nonce_{UE}$ in the TA registration request sent to the new MME, protected with the old NAS keys.
- The new MME requests UE context from the old MME including the TMSI
- Old MME sends the UE context to the new MME, including IMSI and K_{ASME}
- Optionally, UE and the new MME agree upon a fresh K_{ASME} with the help of AKA
- The new MME selects a new $Nonce_{SN}$ and derives K_{NASenc} , K_{NASint} as in 5) and 6) of section 2.1.
- The new MME sends the confirm registration message to UE including $Nonce_{SN}$, protected by the new NAS integrity key
- UE derives fresh K_{NASenc} , K_{NASint} as in 5) and 6) of section 2.1 and starts protection with the new NAS keys

2.4 Key refresh on handover

The goal of refreshing keys during handover is to guarantee that UP and RRC key streams and counters against replay protection do not repeat in case the same encryption or integrity algorithms is used after handover and that weak algorithms used by the source eNB do not influence target eNB and vice versa.

As a general requirement of handover procedures within an LTE network, UE shall not have to be able to distinguish between intra and inter MME handover procedures. As a consequence, the key refresh mechanism on intra MME handover and inter MME handover shall be as similar as possible and shall not require different behaviour from the UE.

2.4.1 Key refresh on Intra MME handover

According to 25.813, v 7.10, Section 9.1.5, on intra MME handover the source eNB sends a handover request to the target eNB. The target eNB replies with a handover response. The handover response includes information required by UE. The source eNB includes this information in the handover command it sends to UE.

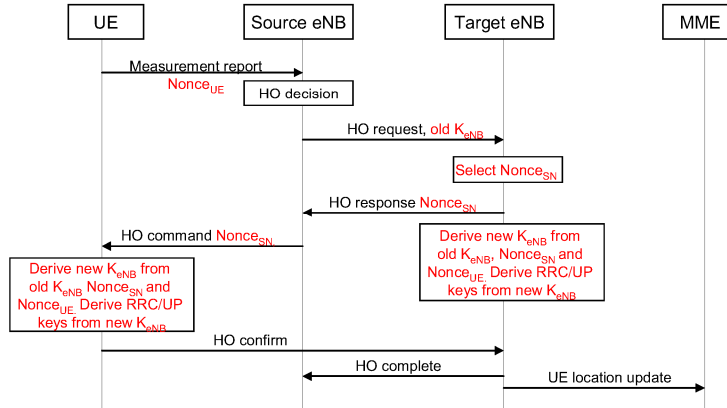


Figure 5: Key re-fresh on intra MME handover

K_{RRCenc} , K_{RRCint} , K_{UPenc} are refreshed on intra MME handover with the help of nonce exchanged between UE and eNB:

- UE transfers $Nonce_{UE}$ to source eNB in a suitable message, e.g. measurement report
- Source eNB transfers K_{eNB} to target eNB in the handover request message
- Target eNB selects a short $Nonce_{eNB}$ and includes it in the response message sent to source eNB. As target eNB chooses C-RNTI at this point and includes it in the handover command it may be used as part of the $Nonce_{SN}$. This may reduce the handover command message length.
- Target eNB derives a new K_{eNB} from $Nonce_{SN}$, $Nonce_{UE}$ and old K_{eNB} by $K_{eNB_new} = KDF(K_{eNB_old} || Nonce_{UE} || Nonce_{SN})$ and derives K_{RRCenc} , K_{RRCint} , K_{UPenc} from the K_{eNB_new} as in 1), 2), and 3) of section 2.1.
- Source eNB includes $Nonce_{eNB}$ in the handover command sent to UE (protected by the RRC integrity key shared between UE and source eNB)
- UE derives the new K_{eNB} , K_{RRCenc} , K_{RRCint} , K_{UPenc} in the same way as the target eNB

A variant of the key derivation functions would be possible here, which would make the key derivation process slightly more efficient, at the expense of a uniform handling of key derivation. This variant differs from the above as follows: the target eNB does not derive a new K_{eNB} , but uses the two nonces $Nonce_{UE}$ and $Nonce_{SN}$ in deriving K_{RRCenc} , K_{RRCint} , K_{UPenc} from K_{eNB} .

2.4.2 Key refresh on Inter MME handover

According to 23.882, v 1.18, Section 7.15 inter MME handover does either not occur at all (due to S1 flexible nature) or is executed with involvement of a target MME. We assume here that in the latter case, the handover command and handover confirm messages are exchanged between UE and the source eNB in the same way as on intra MME handover such that inter and intra MME handover are indistinguishable for the UE. It is ffs if this assumption holds.

On inter MME handover as on intra MME handover, the old K_{eNB} is transferred to the target eNB. A new fresh K_{eNB} is derived from the old K_{eNB} , and K_{RRCenc} , K_{RRCint} , K_{UPenc} are refreshed with the help of the new K_{eNB} . The proposed procedure is detailed in Figure 6:

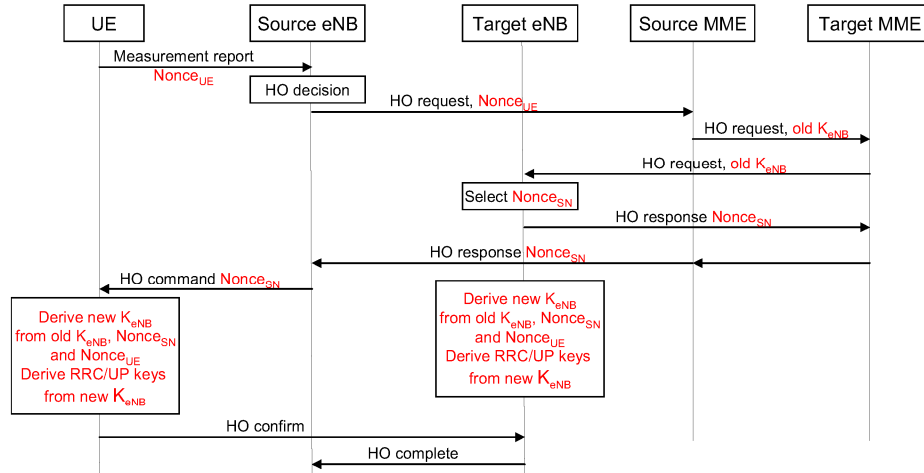


Figure 6: Key refresh on inter MME handover

- UE transfers Nonce_{UE} to source eNB in a suitable message, e.g. a measurement report
- Source MME transfers the old K_{eNB} to the target MME in the handover request
- Target MME includes the old K_{eNB} in the handover request sent to target eNB
- Target eNB includes Nonce_{SN} in the handover response to target MME. As target eNB chooses C-RNTI randomly at this point anyway, the C-RNTI shall be part of Nonce_{SN} ¹
- Target eNB derives the new K_{eNB} by $K_{\text{eNB_new}} = \text{KDF}(K_{\text{eNB_old}} \parallel \text{Nonce}_{\text{UE}} \parallel \text{Nonce}_{\text{SN}})$ and derives K_{RRCenc} , K_{RRCint} , K_{UPenc} from new K_{eNB} as in 1), 2), and 3) of Section 2.1
- Target MME includes Nonce_{SN} in the handover response to source MME which sends it to source eNB
- Source eNB sends the handover command message to UE including Nonce_{SN} . This message is protected with the old RRC integrity key.
- UE derives new K_{eNB} from Nonce_{SN} , Nonce_{UE} and old K_{eNB} by $K_{\text{eNB_new}} = \text{KDF}(K_{\text{eNB_old}} \parallel \text{Nonce}_{\text{UE}} \parallel \text{Nonce}_{\text{SN}})$ derives K_{RRCenc} , K_{RRCint} , K_{UPenc} from $K_{\text{eNB_new}}$ as in 1), 2), and 3) of Section 2.1

The same variant of the key derivation functions mentioned for intra-MME handover would be possible here also.

3 Conclusion

We propose that SA3 adopts the proposed procedures as working assumption on how to ensure key stream freshness within SAE/LTE and includes the above text in TR 33.821.

¹ If C-RNTI is long enough, it can be used as Nonce_{SN} directly. Otherwise the target eNB shall use additional random bits for Nonce_{SN}