2

UMTS Security Features in Release 1999

2.1 Access Security to UMTS

Radio access technology will change from TDMA (Time Division Multiple Access) to WCDMA (Wideband Code Division Multiple Access) when the Third Generation (3G) mobile networks are introduced. Despite this shift, requirements for access *security* will not change. It is an absolute prerequisite of UMTS (Universal Mobile Telecommunications System) that end-users of the system are *authenticated* (i.e., the identity of each subscriber is verified): nobody wants to pay for fraudulent calls that are made by others.

The *confidentiality* of voice calls is protected in the Radio Access Network (RAN), as is the confidentiality of transmitted user data. This means that the user has control over choosing the parties he or she wants to communicate with. Users also want to *know* that confidentiality protection is really applied and so *visibility* of applied security mechanisms is needed. *Privacy* of a user's whereabouts is generally appreciated; most of the time an average citizen does not care whether it is possible to trace where he or she is, but if persistent tracking occurs the user would rightly be irritated. Similarly, precise information about the location of people would be useful to burglars. The privacy of user data is another issue that is critical during transfer through the network (privacy and confidentiality are largely synonymous in this presentation).

UMTS *accessibility* is clearly important for subscribers who are paying for it, but network operators consider *reliability* of network functionality to be equally important: they need control within network functions to be effective. This is guaranteed by the *integrity* of radio network signalling, which checks that all control messages have been created by authorized elements of the network. In general, integrity checking protects against any manipulation of a message (e.g., insertion, deletion or substitution).

The most important ingredient in providing security for network operators and subscribers is *cryptography*, which consists of various techniques that have their

roots in the science and art of *secret writing*. It is sometimes useful to make communication deliberately incomprehensible (i.e., using ciphers or, synonymously, encryption). This is the most effective way to protect communications against eavesdroppers. Cryptographic issues are thoroughly discussed in Part II.

In the present chapter, we go through the security features introduced in the first release of the 3GPP system specifications (Release 1999).

2.1.1 Mutual authentication

There are three entities involved in the authentication mechanism of the UMTS system:

- Home Environment (HE);
- Serving Network (SN);
- terminal, more specifically USIM (Universal Subscriber Identity Module), typically in a smart card.

The basic idea is that the SN checks the subscriber's identity (as in GSM—Global System for Mobile communications) by a *challenge-and-response* technique while the terminal checks that the SN has been authorized by the home network to do so. The latter part is unique to UMTS (not available with GSM) and through it the terminal can check that it is connected to a legitimate network.

The mutual authentication protocol itself does not prevent the active attack scenario of Figure 1.1, but in combination with other security mechanisms it guarantees that the active attacker cannot get any real benefit out of the situation. The only possible gain for the attacker is to be able to disturb the connection (but an attacker could also do this by means of radio-jamming). At the moment no protocol method can circumvent such an attack.

The cornerstone of the authentication mechanism is a *master key* or a subscriber authentication key K, which is shared between the USIM of the user and the home network database, Authentication Centre (AuC). The key is permanently kept secret and has a length of 128 bits. The key K is never transferred from these two locations (i.e., the user has no knowledge of the master key).

Apart from mutual authentication, keys for encryption and integrity checking are also derived. These are temporary keys (with the same length of 128 bits) and are derived from the permanent key K during every authentication event. It is a basic principle in cryptography to keep the use of permanent keys to a minimum and, instead, derive temporary keys from it for protection of bulk data.

We now describe the Authentication and Key Agreement (AKA) mechanism at a general level. The design of the mechanism was begun by combining two different

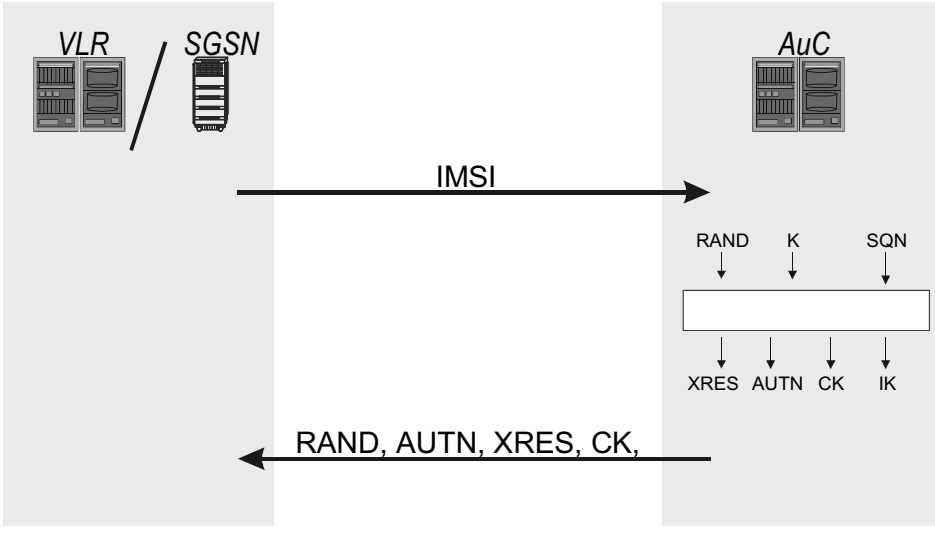

Figure 2.1 Authentication data request and authentication data response
VLR = Visitor Location Register; SGSN = Serving GPRS Support Node; AuC = Authentication Centre; IMSI = International Mobile Subscriber Identity; RAND = random number; SQN = sequence number; XRES = expected response; AUTN = authentication token; CK = Cipher Key; IK = Integrity Key; GPRS = General Packet Radio Service

authentication mechanisms: GSM's authentication and key agreement mechanism [29] and a generic authentication mechanism based on sequence numbers specified in an ISO standard [63].

The authentication procedure begins when the user is identified in the SN. Identification occurs when the identity of the user (i.e., permanent identity International Mobile Subscriber Identity (IMSI), or temporary identity Temporary Mobile Subscriber Identity (TMSI), or Packet TMSI (P-TMSI)), has been transmitted to the VLR (Visitor Location Register) or SGSN (Serving GPRS Support Node). Then the VLR or SGSN sends an *authentication data request* to the AuC in the home network.

The AuC contains the master key of each user and, based on the knowledge of IMSI, the AuC is able to generate *authentication vectors* for the user. The generation process contains executions of several cryptographic algorithms, which are described in more detail in Chapter 8. The generated vectors are sent back to the VLR/SGSN in the *authentication data response*. This process is depicted in Figure 2.1. These control messages are carried on the MAP (Mobile Application Part) protocol.

In the SN, one authentication vector is needed for each authentication instance (i.e., for each run of the authentication procedure). This means that the (potentially-long distance) signalling between SN and AuC is not needed for every authentication event and that in principle this signalling can be done independently of user actions after initial registration. Indeed, the VLR/SGSN may fetch new authentication vectors from AuC well before the number of stored vectors runs out.

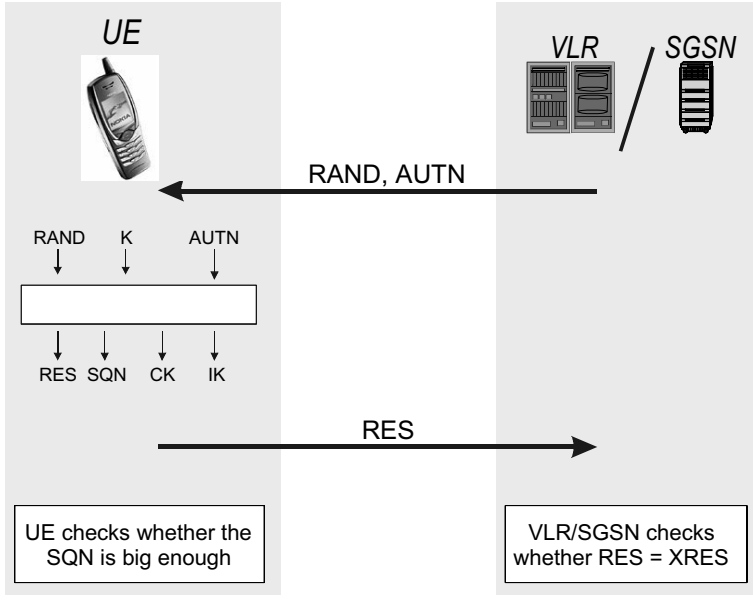

Figure 2.2 User authentication request and user authentication response
UE = User Equipment; VLR = Visitor Location Register; SGSN = Serving GPRS Support Node; RAND = random number; AUTN = authentication token; RES = user response; SQN = sequence number; CK = Cipher Key; IK = Integrity Key; XRES = expected response; GPRS = General Packet Radio Service

The SN (VLR or SGSN) sends a *user authentication request* to the terminal, containing two parameters from the authentication vector, called RAND and AUTN. These parameters are transferred to the USIM, which exists inside a tamper-resistant environment (i.e., in the Universal Integrated Circuit Card—UICC). The USIM contains the master key K and, using it with the RAND (random number) and AUTN (authentication token) parameters along with other input values, USIM carries out a computation that resembles the generation of authentication vectors in AuC. This process also involves running several algorithms, just as in the corresponding AuC computation. The result of the computation gives the USIM the ability to verify whether the AUTN parameter:

- was indeed generated in AuC;
- was not sent beforehand to the USIM.

In the positive case, the computed RES parameter is sent back to the VLR/SGSN as part of the *user authentication response*. Now, the VLR/SGSN is able to compare the user response (RES) with the expected response (XRES), which is part of the authentication vector. If they match, authentication ends positively. This part of the process is depicted in Figure 2.2.

The keys for Radio Access Network (RAN) encryption and integrity protection (namely, Cipher Key (CK) and Integrity Key (IK)) are created as a by-product in the authentication process. These temporary keys are included in the authentication vector and, thus, are transferred to the VLR/SGSN. These keys are later transferred to the Radio Network Controller (RNC) in the RAN when encryption and integrity protection start. Respectively, the USIM is able to compute the CK and IK after it has obtained the RAND (and verified it through the AUTN). Temporary keys are subsequently transferred from USIM to the Mobile Equipment (ME) where the encryption and integrity protection algorithms are implemented.

In the following sections we take a more detailed look at the mechanisms needed for authentication and key agreement.

2.1.1.1 Authentication vector generation

We now take a closer look at the generation of authentication vectors in the AuC. An illustration of the process is given in Figure 2.3. The process begins by picking an appropriate sequence number (SQN). Roughly speaking, what is required is that SQNs are chosen in ascending order. A more detailed description about how to create SQNs is given in Section 2.1.1.3. The purpose of the SQN is to provide the user (or more technically the USIM) with proof that the generated authentication vector is *fresh* (i.e., it has not been used before in an earlier run of authentication). In parallel with the choice of SQN, a 128-bit long RAND is generated. This is a

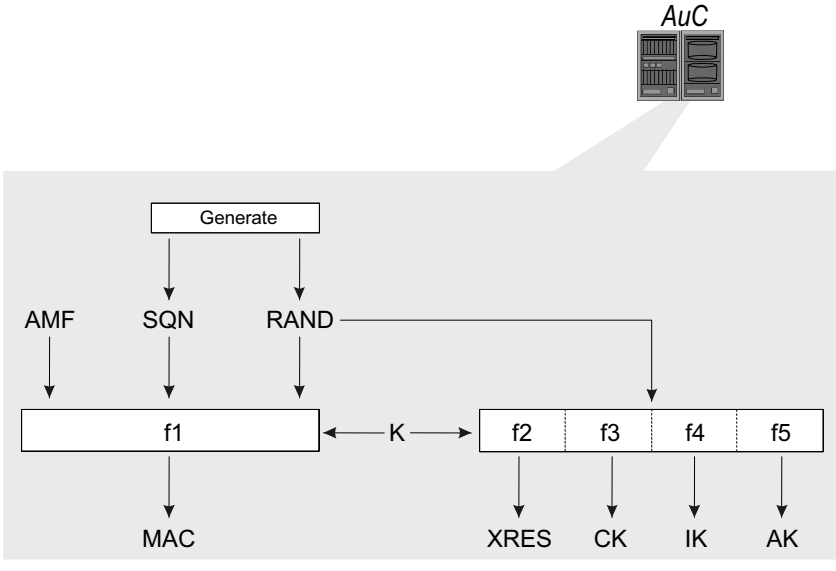

Figure 2.3 Authentication vector generation
AuC = Authentication Centre; AMF = Authentication Management Field; SQN = sequence number; RAND = random number; MAC = Message Authentication Code; XRES = expected response; CK = Cipher Key; IK = Integrity Key; AK = Anonymity Key

demanding task in itself, but in this presentation we just assume that a cryptographic pseudorandom generator is in use that is able to produce large amounts of unpredictable output bits, when a good physical random source is available to produce smaller amounts of random bits that can be used as an input (seed) for the pseudorandom generator.

The key concept in authentication vector computation is a mathematical function, called *one-way function*, which is relatively easy to compute but practically impossible to invert. In other words, as long as we have input parameters there exists a fast algorithm to compute output parameters, but if the output parameters are not known, then there exist no efficient algorithms to deduce any input that would produce the output. Of course, there is a simple algorithm, called the exhaustive search algorithm, that can be used to find the correct input by trying all possible choices until one gives the requisite output. However, this algorithm quickly becomes extremely inefficient as the length of input increases.

In total, five one-way functions are used to compute the authentication vector. These functions are denoted by f1, f2, f3, f4 and f5. The function f1 differs from the other four in that it takes four input parameters: master key K, RAND, SQN and finally an administrative Authentication Management Field (AMF). All other functions from f2 to f5 only take K and RAND as inputs. The requirement of the one-way property is common to all functions f1–f5. They can all be built around the same *core* function. However, it is essential that they differ from each other in a fundamental way so that the output of one function reveals no information about the outputs of the other functions. The output of f1 is Message Authentication Code (MAC) (64 bits) and the outputs of f2, f3, f4 and f5 are, respectively, XRES (32–128 bits), CK (128 bits), IK (128 bits) and AK (64 bits). The authentication vector consists of the parameters RAND, XRES, CK, IK and the authentication token (AUTN). The last one is obtained by concatenating three different parameters: SQN added bit by bit to AK, AMF and MAC. All of the functions involved in the AKA procedure are studied in detail in Chapter 8 of this book.

2.1.1.2 Authentication on the USIM side

We now take a closer look into the handling of authentication on the USIM side (illustrated in Figure 2.4). The same functions f1–f5 are involved on this side but in a slighty different order. The function f5 has to be computed before the f1, since f5 is used to conceal the SQN. This concealment is needed in order to prevent eavesdroppers from getting information about the user identity through the SQN. The output of the function f1 is marked XMAC (or XMAC-A) on the user side. This is compared with the MAC received from the network as part of the parameter AUTN. If there is a match it implies RAND and AUTN have been created by some entity that knows K (i.e., the AuC of the user's home network).

Of course, there is still the possibility that some attacker who has recorded an

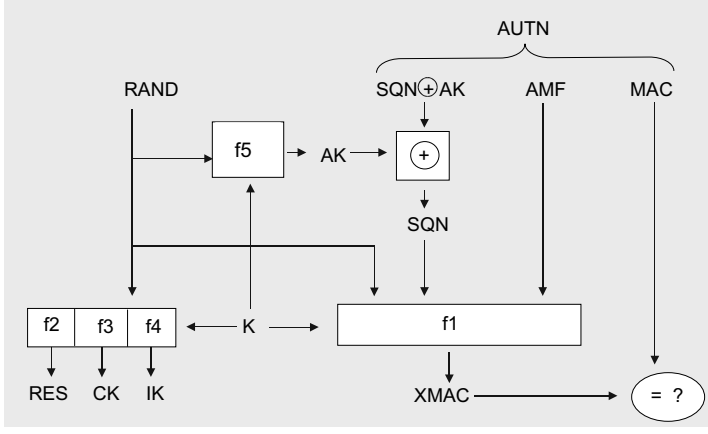

Figure 2.4 Authentication handling in USIM
AUTN = authentication token; RAND = random number; SQN = sequence number; AK = Anonymity Key; AMF = Authentication Management Field; MAC = Message Authenication Code; RES = user response; CK = Cipher Key; IK = Integrity Key; XMAC = expected MAC

earlier authentication event could ascertain the RAND and AUTN. However, as mentioned above, the SQN protects against this threat. The USIM should simply check that it has not seen the same SQN before, and the easiest way to do this is to require that SQNs appear in ascending order. It is also possible for the USIM to allow some SQNs to arrive out of order (e.g., by maintaining a shortlist of the greatest SQNs received so far). In the next sections we will take a closer look at this issue.

Since the transfer of authentication vectors from the AuC and the actual use of these vectors for authentication are done somewhat independently, there are several reasons why it is possible that authentication vectors may be used in a different order from which they were originally generated. The most obvious reason for this is because of the fact that mobility management functions for the CS (Circuit Switched) and PS (Packet Switched) domain are independent of each other, implying that authentication vectors are fetched to the VLR and SGSN independently of each other and that the vectors are also used independently.

The choice of algorithm (f1–f5) is in principle operator-specific, because they are only used in the AuC and in the USIM and the same home operator controls both of these entities. An example set of algorithms (called MILENAGE) exists in the Third Generation Partnership Project (3GPP) specification TS 35.206 [24] (these algorithms are discussed thoroughly in Chapter 8).

2.1.1.3 SQN generation in the AuC

SQN management is also operator-specific in principle. There are two basic strategies at work in creating SQNs: each user may have an individual SQN, or SQN

generation may be based on a global counter (e.g., universal time). A combination of these two strategies is also possible in which the most significant part of the SQN is user-specific but the least significant part is based on a global counter.

In the 3GPP specification 33.102 [9] there is an informative annex C that describes three different options for generating SQNs. Because this part of the specification is only for informative purposes, the network operator is also free to choose some other way of generating SQNs while remaining fully compliant with 3GPP standards. However, it has been observed in practice that excessive diversity inside one standard tends to lead in the long run to interoperability problems of some sort or another. This observation is by no means limited to security mechanisms.

Let us discuss this important issue a bit further. There is a widely-held agreement inside the 3GPP that different optional functionalities for the same purpose in the same standard should be avoided if ever possible. In standardization specification work a decision often has to be made between two (or more) proposed solutions that have equal technical merits but are simply different ways of achieving the same goal. An easy way out from such a situation is to allow different options in the standard. At first sight it may look like the only penalty that has to be paid for such a compromise decision is the risk that some elements in the system may have to contain redundant, duplicate functionality. However, when viewed in depth there is the much bigger issue of future specifications. It may happen that a new functionality is designed on top of the old functionality for which several implementation options were allowed. As a consequence it becomes difficult to stop these options from being available to the new functionality, which may also be dependent on a number of other old functionalities that again may well contain several options. So, it is difficult to keep the design of the new functionality simple in such cases.

This general concern certainly applies to our context because the UICC manufacturers and AuC manufacturers are usually (if not always) different companies. Also, the issue with future standards has emerged, as the AKA mechanism has been introduced into new contexts in 3GPP Releases 5 and 6 (see Chapter 3). For these reasons, it can be anticipated that the example mechanisms for SQN management presented in [9] are likely to be adopted widely in practice.

Let us now give a brief description of the example mechanisms. However, for full details see annex C in [9]. The SQN for a certain user contains two concatenated parts: SQN = SEQ | IND. The least significant part (5 bits) IND is used to allow effective mechanisms in the USIM side to verify the freshness of the SQN parameter. The general rule is that the IND value is incremented by one for each new authentication vector to be generated. This increment is understood cyclically (i.e., when the IND parameter reaches the maximal value then the next value to be chosen is zero).

It is possible, although not usually the case, that the AuC gets information about the type of node requesting the vector (e.g., whether it is a MSC/VLR or a SGSN). When this happens, it may be useful to differentiate the range of IND values allocated to nodes in each different domain. For instance, IND values that are

Table 2.1 Partitioning the IND value space (an example)

Access to domain (AV sent to)	IND value range
CS domain (MSC/VLR)	0–9
PS domain (SGSN)	10–19
IMS domain (S-CSCF)	20–24
WLAN domain	25–28
Other domains	29–31

AV = Authentication Vector; IND =least significant part of SQN; CS = Circuit Switched; MSC/VLR = Mobile Switching Centre/Visitor Location Register; PS = Packet Switched; SGSN = Serving GPRS Support Node; IMS = IP Multimedia CN Subsystem; IP = Internet Protocol; CN = Core Network; S-CSCF = Serving Call Session Control Function; WLAN = Wireless LAN; GPRS = General Packet Radio Service; LAN = Local Area Network

even are allocated to nodes in the CS domain and odd IND values are allocated to nodes in the PS domain. Consequently, in this example, for two consecutive authentication vectors allocated to the same domain, the difference in IND value would be 2 instead of 1. As mentioned earlier, according to the more recent releases of 3GPP standards, authentication vectors may also be consumed in other domains (e.g., by IMS (IP Multimedia Core Network Subsystem) (Release 5) or by an interworking WLAN (Wireless Local Area Network) system (Release 6)). For effective handling of SQNs in even these cases, it may be useful to introduce a more fine-grained partition of the IND value space. How this partition should exactly be done is highly dependent on the structure of the network in question and the optimal partition probably changes as the network evolves. To elaborate on this a bit more, let us look at the way of partitioning the IND value space as given in Table 2.1.

Authentication Vectors (AVs) may be sent to the destination in *batches*. This reduces the number of times the AuC has to be accessed. At the same time, there is an increased probability that AVs are consumed in a different order than they were generated. Typically, all AVs in a batch share the same value of SEQ and only differ in the value of IND. There are three different strategies used to generate the value of SEQ:

1. SEQ is an individual counter and its current value is maintained in a database independently for each user.

2. SEQ is based on a global counter and for each user a deviation from the global counter, called DIF (difference), is maintained in a database. Ideally the DIF value is 0 for all users, but because of synchronization errors (see Section 2.1.1.5) it may have to be updated for some users.

3. SEQ has two parts: SEQ = SEQ1 | SEQ2, where SEQ1 is an individual counter and SEQ2 is based on a global counter (GLC) that represents universal time. The value of SEQ is maintained in the database for each user in this case as well.

All three ways of generating SEQ are described in [9]. Here we only present the third strategy because the other two can be seen (more or less) as extreme cases of the combined case 3.

A suggested length for parameter SEQ2 is 24 bits, leaving 19 bits for the individual counter SEQ1 as IND consists of 5 bits (the length of a SQN is fixed at 48 bits). The GLC also consists of 24 bits. Using time units of 1 second, the GLC would wrap around once in 194 days. This ensures that almost all users would be authenticated at least once during each GLC period.

The idea is to keep the most significant part (SEQ1) constant until SEQ2 wraps around. The latter correlates heavily with GLC, and therefore the wrap-around would typically happen once in 194 days. We cannot assume that SEQ2 = GLC because it is possible that two batches of authentication vectors are fetched for the same user exactly at the same time (or at least during the same time unit of GLC). Remember that different domains consume and fetch AVs from the AuC independently of each other, and therefore fetching can certainly happen simultaneously. If another fetch occurs during the same time unit as the previous fetch then SEQ2 is incremented by 1 anyway. As a result, SEQ2 would become temporarily greater than GLC in the second fetch.

We can safely assume that the GLC clock rate is on average faster than the authentication frequency for any one specific user. Therefore, even if the SEQ2 temporarily overtakes the GLC, the latter catches up fairly quickly.

Let us assume that the AuC gets a request for a batch of AVs. First, the previous value of SEQ is retrieved from the database (for this particular user) and then the previous value of SEQ2 is compared with the current GLC value. We have three possible cases (see also Figure 2.5):

1. SEQ2 < GLC. This is the usual case, with the new SEQ2 value set to be equal to the current GLC value while the SEQ1 value remains unchanged;

2. GLC < SEQ2, but any difference is small (or even zero). This is the case discussed above (i.e., the previous generation of an AV and, consequently, the previous update of the SEQ2 value have happened very recently or there have been many updates in the very recent past). Here SEQ2 is incremented by 1 and SEQ1 remains unchanged unless there is a wrap-around of SEQ2 as a result of this increment, in which case the SEQ1 is also incremented by 1;

3. GLC < SEQ2, and the difference is large. This is the case where a wrap-around of GLC has occurred since the generation of the last batch of AVs. Here the new

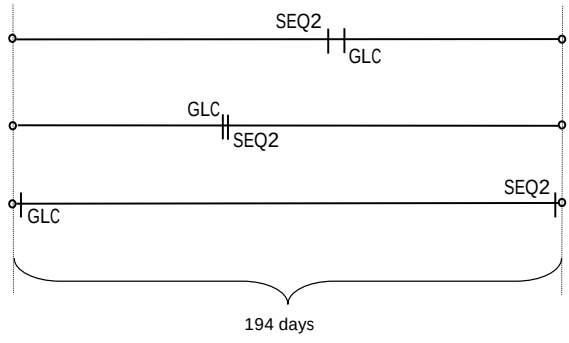

Figure 2.5 SEQ2 update cases
SEQ2 = least significant part of SEQ; GLC = global counter

SEQ2 value is set to be equal to the current GLC value while the SEQ1 value is incremented by 1.

Obviously a precise threshold value is needed to differentiate between cases 2 and 3 and a recommended value is given in [9]: a difference is deemed small if (and only if) it is smaller than 2^{16}. Note that if many, almost simultaneous AVs were accidentally categorized to class 3 instead of class 2, then the only disadvantage is that SEQ1 would be incremented unnecessarily. This does not, however, lead to SQN values being reused. In the opposite case, where the user is inactive for a very long period (say, almost a full period of 194 days) and, as a consequence, AV generation falls into class 2 instead of class 3, there is a small risk that some SQN values could be reused, even though this is never supposed to happen. Anyway, there is no *security* risk involved, since reusing SQN values only implies that network authentication may fail on the USIM side and recovery from this kind of situation is guaranteed by the resynchronization procedure.

2.1.1.4 SQN checking in USIM

As mentioned earlier, SQNs exist in the AKA concept for one reason: they allow the USIM to check whether the authentication challenge has been received before. In addition to the highest SQN received so far, the USIM maintains an array of other received SQNs. The array is indexed by the elements within the range of the IND parameter. In our example case where the length of the IND parameter is 5 bits, consequently there are 32 elements in the array. The element in the array indexed by I is the highest SQN received so far with IND = I.

Let us suppose the USIM receives an SQN = SEQ || IND. This can be compared with the element in the maintained array that is indexed by the value of IND.

If the received SQN is greater than the value in the array, then the SQN is accepted, the network authentication succeeds and the element in the array is

replaced by the received value SQN. Clearly, it is enough just to store the SEQ part, because the IND part is indicated by the index of the element in the array.

On the other hand, if the received value is smaller or equal to the value in the array, then the received SQN is not accepted and the network authentication fails. The array element is not changed, but, instead, a resynchronization procedure is initiated.

In addition to the comparison mentioned above, the received SQN value is also checked against the highest SQN value stored in the USIM. The purpose of this check is to prevent arbitrarily-big jumps in SQN values. Therefore, the received SQN value is not accepted unless it increases the highest SQN value by at most a value of Δ. If this check is not done, there is a small chance that the USIM gets into a situation where it has accepted and stored such large values of SQN that the only way out is to do a wrap-around of these counters. The limit value Δ has to be chosen carefully in order to make sure that normal jumps in SQN are not considered abnormal. An example value $\Delta = 2^{28}$ is given in [9].

2.1.1.5 Synchronization of SQNs

The mutual authentication mechanism is based on two parameters that are stored in both the AuC and USIM: a static master key K and a dynamic SQN. It is vital that these parameters are kept synchronized on both sides. For the static K this is easy, but it is possible for dynamic SQNs to get out of synchronization for some reason. As a consequence, authentication would fail. A specific *resynchronization procedure* is used in this case (see Figure 2.6). By using the master key K as the basis for secure communication, the USIM informs the AuC of its current (highest) SQN value.

The AUTS parameter is delivered during resynchronization. It contains two parts: the sequence number of the USIM concealed by AK and a message authentication code MAC-S computed by another one-way function f1* from the input parameters SQN, K, RAND and AMF. The last two parameters are obtained from the failed authentication event. The one-way function f1* has to be different from f1 because, otherwise, already recorded AUTN parameters could in principle be accepted as valid AUTS parameters in the resynchronization and an attacker could at least disturb the authentication process. When the AuC receives the AUTS parameter, it carries out the following steps:

1. The $\mathrm{SQN}_{\mathrm{USIM}}$ is computed from AUTS.
2. Based on the value of $\mathrm{SQN}_{\mathrm{USIM}}$, the AuC checks whether the next authentication vector would be acceptable to the USIM—
   a if YES, then the process continues from step 4;
   b if NO, then

3. The AuC checks whether the the MAC-S value in AUTS is correct—
   a if YES, then the value SQN_{AuC} is reset to SQN_{USIM} and the process continues from step 4;
   b if NO, then SQN_{AuC} is not reset but the process continues anyway from step 4.
4. The AuC sends a batch of fresh AVs to the VLR/SGSN.

Note that new AVs are also sent to the SN node when the MAC-S value is either not checked (2a) or failed the check (3b). However, a general cryptographic principle states that no action should be taken when a message authentication code turns out to be false. Nevertheless, in our case the sending of new AVs is justified for the following reasons:

- if the AUTS parameter was computed by the genuine USIM, the UE would then try to get access to the network again after the first attempt has failed, resulting in a new AV being needed from the AuC in any event;
- if the AUTS parameter was computed and sent to an attacker (for whatever reason), the attacker might try to access the network again, but once more a new AV is probably fetched from the AuC.

Equipped with the new AVs, the VLR/SGSN is able to authenticate the UE in case he or she tries to get access again. If the UE repeatedly indicates network authentication failures by sending more AUTS values, the two most probable reasons are:

- there is something wrong with the computations or data in the USIM;
- the UE is actually an attacker who tries to run a denial-of-service attack against the network.

In both cases, the best course of action is to deny access to the UE in question.

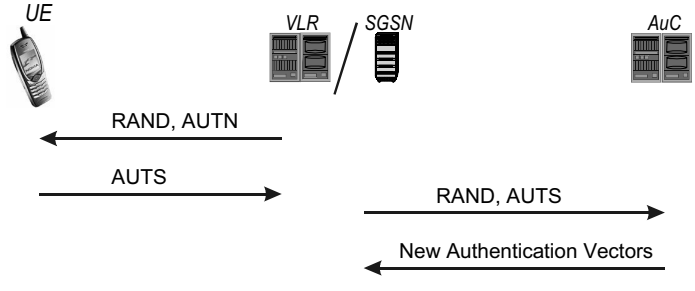

Figure 2.6 Resynchronization procedure
UE = User Equipment; VLR = Visitor Location Register; SGSN = Serving GPRS Support Node; AuC = Authentication Centre; RAND = random number; AUTN = authentication token; AUTS = authentication token in re-synchronization; GPRS = General Packet Radio Service

2.1.1.6 Illustrative flow chart of authentication

In Figure 2.7 we show a flow chart outlining the mutual authentication procedure, including the potential resynchronization phase. There also exists a procedure for reporting authentication failures from the VLR/SGSN to the Home Location Register (HLR) (see [9]). This procedure is not included in the flow chart.

2.1.2 Temporary identities

The permanent identity of the user in UMTS is IMSI (as is also the case in GSM). However, identification of the user in UTRAN (UMTS Terrestrial Radio Access Network) is in almost all cases effected by means of temporary identities: TMSI in the CS domain or P-TMSI in the PS domain. Confidentiality of user identity is thus protected (almost always) against passive eavesdroppers. Initial registration is, of course, the exception because a temporary identity cannot be used since the network does not yet know the permanent identity of the user. After that it is possible to use temporary identities.

The mechanism works as follows. Assume the user has already been identified in the SN by IMSI. Then the SN (VLR or SGSN) allocates a temporary identity (TMSI or P-TMSI) for the user and maintains an association between the permanent identity and the temporary identity. The latter only has local value and each VLR/SGSN simply takes care that it does not allocate the same TMSI/P-TMSI to two different users simultaneously. The allocated temporary identity is transferred to the user once encryption is turned on. This identity is then used in both uplink and downlink signalling until the network allocates a new TMSI (or P-TMSI). Paging, location update, attach and detach are examples of signalling that utilizes (P-)TMSI.

Allocation of a new temporary identity is acknowledged by the terminal, and then the old temporary identity is removed from the VLR (or SGSN). If allocation acknowledgement is not received by the VLR/SGSN it keeps both the old and new TMSIs and accept either of them in uplink signalling. In downlink signalling, IMSI must be used because the network does not know which temporary identity is currently stored in the terminal. In this case, VLR/SGSN tells the terminal to delete any stored TMSI/P-TMSI and a new reallocation follows.

However, one problem remains: how does the SN obtain the IMSI in the first place? Since the temporary identity only has local meaning, the identity of the local area has to be appended to it in order to obtain a unique identity for the user. This is resolved by appending the Location Area Identity (LAI) to the TMSI and the Routing Area Identity (RAI) to the P-TMSI.

If the UE enters a new area, then the association between IMSI and (P-)TMSI can be fetched from the old location or routing area if the new area knows its address

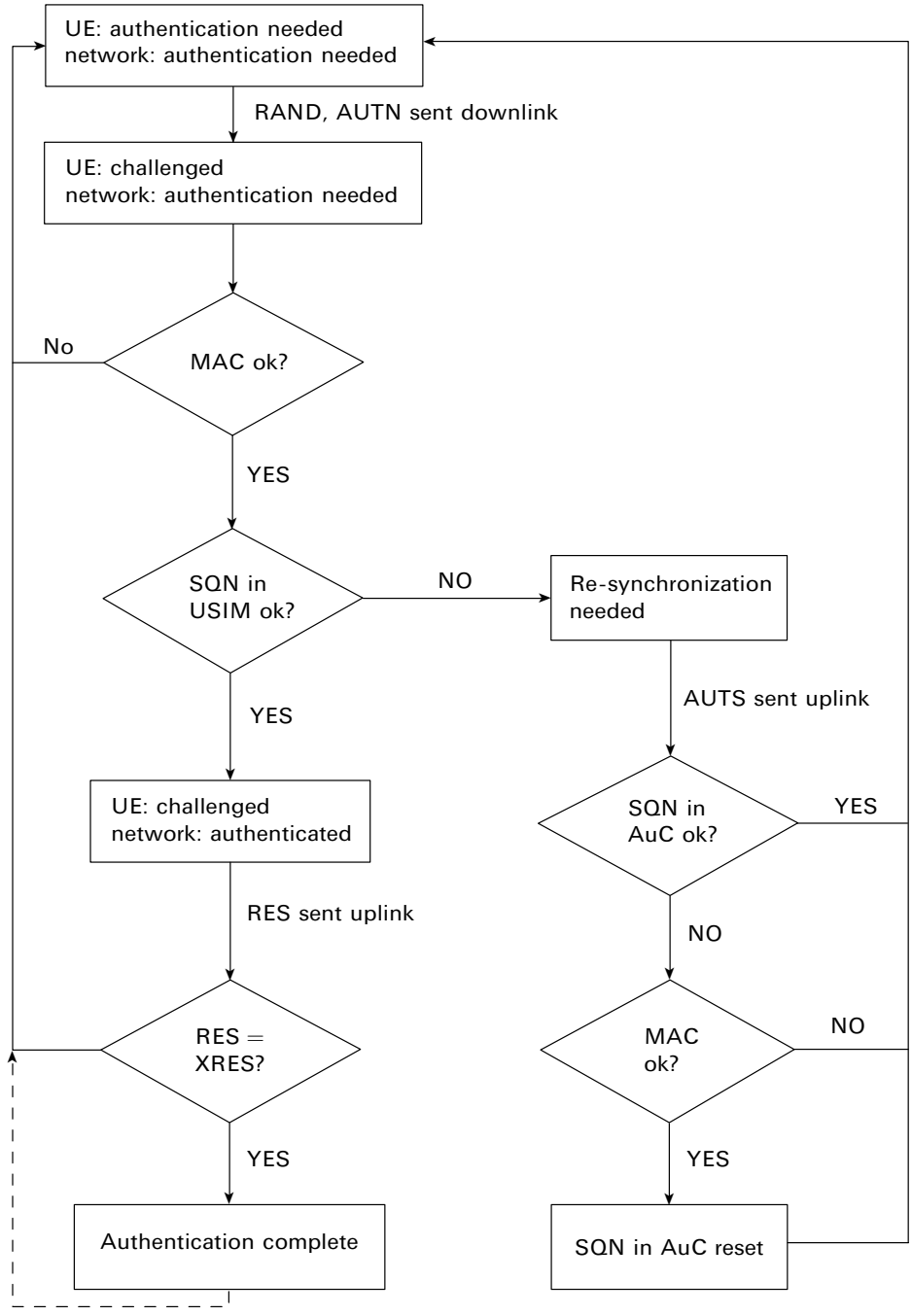

Figure 2.7 Authentication flow chart

UE = User Equipment; RAND = random number; AUTN = authentication token; MAC = Message Authentication Code; SQN = sequence number; USIM = Universal Subscriber Identity Module; AUTS = authentication token in re-synchronization; AuC = Authentication Centre; RES = user response; XRES = expected response

(based on LAI or RAI). At the same time, unused AVs can also be transferred from the old VLR/SGSN to the new VLR/SGSN (if there are any). If the address is not known or a connection to the old area cannot be established, then the IMSI must be requested from the UE.

There are certain places, such as airports, where lots of IMSIs may be transmitted over the radio interface as people switch on their mobile phones after the flight. This means that the people arriving can in principle be identified should an eavesdropper know their IMSIs. On the other hand, the ability to track people is also easier in such places (e.g., by observing who gets off which plane!).

Although the user identity confidentiality mechanism in UMTS does not give 100% protection, it offers a pretty good level of protection. Note that protection against an active attacker is not very good since the attacker may pretend to be a new SN and the user is likely to reveal his or her permanent identity. The mutual authentication mechanism does not help here since the user has to be identified before he or she can be authenticated.

Further details about handling temporary identities can be found in [29] and [2].

2.1.3 UTRAN encryption

Once the user and the network have authenticated each other they may begin secure communication. As described earlier, a CK is shared between the CN and the terminal after a successful authentication event. Before encryption can begin, the communicating parties also have to agree on the encryption algorithm. In a UMTS, implemented according to 3GPP Release 1999, only one algorithm is defined. At the time of writing, the specification process has begun with the remit of designing another encryption algorithm for fallback purposes.

It is in general a good security principle to take precautions against the potential situation where the cryptographic algorithm used in the system suddenly fails. Although there is typically a time gap between any first theoretical attack and widespread practical attacks, this time is not necessarily long enough to allow introduction of another algorithm. If two algorithms could be used at the same time, then if one of them fails the security of the system would not be jeopardized.

Encryption and decryption take place in the terminal and in the RNC on the network side, which means that the CK has to be transferred from the core network (CN) to the RAN. This is done in a specific Radio Access Network Application Protocol (RANAP) message, called the *security mode command*. After the RNC has obtained the CK, it can switch encryption on by sending a Radio Resource Control (RRC) security mode command to the terminal.

The UMTS encryption mechanism is based on a *stream cipher* concept as described in Figure 2.8. This means that plaintext data are added bit by bit to random-

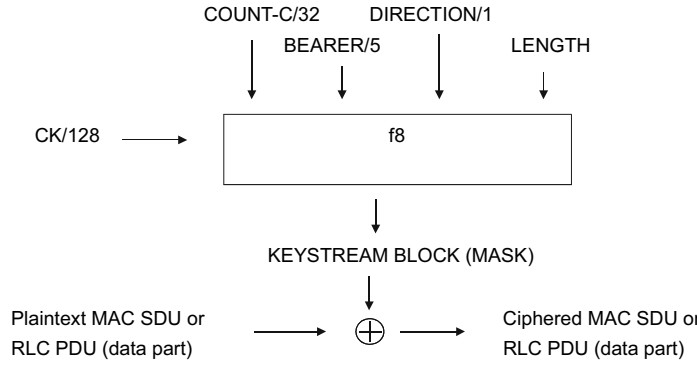

Figure 2.8 Stream cipher concept
CK = Cipher Key; MAC SDU = Medium Access Control Signalling Data Unit; RLC PDU = Radio Link Control Protocol Data Unit

looking mask data that are generated by the CK and a few other parameters. This type of encryption has the advantage that the mask data can be generated even before the plaintext is known, resulting in final encryption being a very fast bit operation. Decryption on the receiving side is done in exactly the same way, since adding the mask bits twice has the same result as adding zeros.

Because mask data do not depend on plaintext, there has to be another input parameter that changes every time a new mask is generated. Otherwise, the same mask would protect two different plaintexts, say P_1 and P_2, resulting in the following unwanted phenomenon: if we add P_1 to P_2 bit by bit and do the same to their encrypted counterparts, then the resultant bit string is exactly the same in both cases. This is a consequence of the fact that two identical masks cancel each other during bit-by-bit addition. Therefore, any attacker who eavesdrops the corresponding encrypted messages on the radio interface would know the bit-by-bit sum of P_1 and P_2. So, if two bit strings of meaningful data are added to each other bit by bit, their content could be discovered from the resultant bit string, which means encryption of the two messages P_1 and P_2 would be broken. The example below illustrates how effective this kind of break is.

2.1.3.1 Example: breaking encryption when mask is reused

Plaintext always has some structure. It is not just random data, it contains some *redundancy*. In our example we assume the plaintexts in question are in English. When coded into ASCII bit strings this assumption implies huge redundancy for these bit strings: most ASCII codes never appear and some appear very frequently. For illustrative purposes, however, let us assume a simplified coding for this example: we use only capital letters from A to Z plus space in between words

Table 2.2 Simplified coding for the English alphabet

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
0	1	2	3	4	5	6	7	8	9	10	11	12	13
O	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	**Space**	
14	15	16	17	18	19	20	21	22	23	24	25	26	

(i.e., no punctuation). Letter A is coded as 0. Similarly, B is coded as 1, C as 2, etc. Finally, Z is coded as 25 and space is coded as 26. The full coding is given in Table 2.2.

Encryption is effected as follows. A list of random integers from the interval between 0 and 26 is created and serves as a mask. Each plaintext letter, or more precisely, the number corresponding to it, is added to an entry in the random integer list in a modular fashion (i.e., the numbers 27 and 0 are considered to be equal). Also, for each number that exceeds 27, multiples of 27 are subtracted until the result is between 0 and 26 (e.g., $14 + 21 = 35$, but when 27 is subtracted, the result is 8). For example, assume the plaintext is CAT while the mask is (3, 17, 12). Then the encoded plaintext is (2, 0, 19) and the ciphertext is (5, 17, 4).

This encryption provides perfect security as long as the mask is truly random, is not known to the attacker and is *used only once*. Indeed, any three-letter plaintext could be transformed to the same ciphertext (5, 17, 4) with a suitable mask. MOM encodes to (12, 14, 12) and the mask producing our ciphertext would be (20, 3, 19). Similarly, XYZ is potential plaintext if the mask happens to be (9, 20, 6).

Let us now assume that the same mask has been used to encrypt two different (but extremely) short English texts. Let us try and discover the contents of both those texts. The ciphertexts are the following:

1. (21, 12, 22, 25, 21, 15, 6)
2. (6, 15, 9, 20, 13, 0, 1)

Let us start our analysis with the first two letters:

1. 21 12
2. 06 15

There exist various statistics about the frequencies of letters in average English text as well as statistics about the frequencies pairs of letters (*digrams*) (e.g., [52]). For instance, the 15 most common digrams cover almost 30% of all cases. Thus, we could try to jump-start the analysis by testing the most frequent digrams for the first two letters of the unknown plaintexts.

The most frequent digram in English is TH, encoded as 19 07. If plaintext 1 began with TH, then the mask would be 02 05 and plaintext 2 would begin with 04 10, decoded as EK. This is certainly possible in principle but not very promising, as EK is not among the top 100 commonest digrams and very few English words begin with it.

Next let us try TH for the beginning of plaintext 2: in this case the mask would be 14 08 and plaintext 1 would start with 07 04 (i.e., HE). This is more promising, as HE is one of the most frequent digrams.

The most probable continuation for plaintext 2 is THE + space, encoded 19 07 04 26. This yields a mask 14 08 05 21 and plaintext 1 would be encoded 07 04 17 04, corresponding to HERE. We still seem to be on the right track.

Plaintext 1 continues with a space, hence the next element in the mask is 22 and plaintext 2 continues with 18 (i.e., plaintext 2 is THE S??). The best tactics to adopt now seem to be testing common three-letter words that begin with S. Plaintext 2 = THE SEA would imply plaintext 1 = HERE TF, no good. THE SKY would imply HERE RD, no better. Finally, plaintext 2 = THE SUN implies plaintext 1 = HERE IS, and a very probable solution is found: ''Here is ... the sun''.

Of course, with an automated procedure we could do much more, but this analysis at least gives an idea about how the analysis of longer texts could be easily done.

2.1.3.2 Encryption parameters

UTRAN encryption occurs in either the Medium Access Control (MAC) layer or in the Radio Link Control (RLC) layer. In both cases, there is a counter that changes for each Protocol Data Unit (PDU). In the MAC this is the Connection Frame Number (CFN) and in the RLC it is a specific RLC sequence number (RLC-SN). If these counters were used as input for mask generation the problem explained in the previous paragraph would still occur since these counters wrap around very quickly. This is why a longer counter called a Hyper Frame Number (HFN) is introduced. It is incremented whenever the short counter (CFN in the MAC case and RLC-SN in the RLC case) wraps around. The combination of HFN and the shorter counter is called COUNT-C and is used as ever-changing input to mask generation inside the encryption mechanism.

In principle, the longer counter HFN could also eventually wrap around. Fortunately, it is reset to zero whenever a new key is generated during the AKA procedure. Authentication events are in practice frequent enough to rule out the possibility of HFN wrap-around.

The radio bearer identity BEARER is also needed as an input to the encryption algorithm since the counters for different radio bearers are maintained independently of each other. If the input BEARER was not in use, then this would again lead to a

situation where the same set of input parameters are fed into the algorithm and the same mask would be produced more than once. Consequently, the problem outlined in the example above would occur and the messages (this time in different radio bearers) encrypted with the same mask would be exposed to the attacker.

The DIRECTION parameter indicates whether we encrypt uplink or downlink traffic. The LENGTH parameter indicates the length of data to be encrypted. Note that the value of LENGTH only affects the *number* of bits in the mask bit stream, it does not have any effect on the bits themselves in the generated stream.

The core of the encryption mechanism is the mask generation algorithm, which is denoted as function f8. The specification is publicly available as 3GPP TS 35.201 [19] and is based on a novel *block cipher* called KASUMI (for which there is another 3GPP specification TS 35.202 [20]). This block cipher transforms 64-bit input to 64-bit output. The transformation is controlled by the 128-bit CK. If CK is not known, then there are no efficient algorithms to compute the output from the input or vice versa. In principle, transformation can be done if:

- all possible keys are tried until the correct one is found; or
- an enormous table of all 2^{64} input–output pairs is assembled.

However, both approaches are impossible in practice. These algorithms are presented in detail in Chapters 6 and 7.

It is possible for authentication not to be carried out at the beginning of the connection. In this case the previous CK is used for encryption. The key is stored in the USIM between the connections. The START parameter, which consists of the most significant part of the greatest HFN used so far, is also stored in the USIM. For the next connection, the stored value is incremented by 2 and used as the starting value for the most significant part of HFN. There is also a constant parameter in USIM, called THRESHOLD, that may be used to restrict the maximal lifetime of the keys CK and IK. Whenever START reaches THRESHOLD, generation of new keys is forced by the UE (i.e., the UE informs the network it has no valid keys).

2.1.3.3 UTRAN protocol structure

As the encryption mechanism is built into radio network protocols, we will discuss these protocols in this chapter. The protocols in the RAN in UMTS are divided into three layers:

1. physical layer;
2. link layer;
3. network layer.

This division follows the classical OSI (Open Systems Interconnection) model. Furthermore, layer 2 is divided into several sublayers:

- MAC;
- RLC;
- Packet Data Convergence Protocol (PDCP);
- Broadcast/Multicast Control (BMC).

The physical layer and the MAC support both user (U-) plane and control (C-) plane traffic in an essentially similar manner. Both the PDCP and the BMC only exist in the U-plane, while the RLC and layer 3 are divided into U-plane and C-plane.

Layer 3 is also divided into several sublayers. However, only the lowest sublayer, the RRC, terminates in the UTRAN in the RNC. Higher sublayers terminate in the CN. The RRC protocol exists only in the C-plane. There are two kinds of control messages transported over the radio interface: radio-specific messages generated by RRC and NAS (Non Access Stratum) control messages generated by higher layers. The NAS control traffic includes the Mobility Management (MM) and Call Control (CC) protocols. The RRC sublayer also provides interlayer communication with all lower layers, thus taking care of their configuration.

The services provided by each of the UTRAN layers to higher layers are summarized in the following.

2.1.3.3.1 Physical layer

Physical layer services convert *physical* radio channels to *transport* channels. These can be characterized as describing *how* the data are transferred rather than *what* data are transferred. Layer 1 services include error detection and correction, frequency and time synchronization, multiplexing of transport channels, interleaving, modulation, power control, measurements and execution of soft handovers, among others. The transport channels are divided into two main categories:

- common channels—if only one particular UE (User Equipment) needs to be addressed, inband signalling is used;
- dedicated channels (DCH)—the whole channel is reserved for one particular user.

Common channels include the Random Access Channel (RACH) for transmitting short uplink messages (e.g., for initial access), the Forward Access Channel (FACH)

for short downlink messages, the Paging Channel (PCH) and the Broadcast Channel (BCH), among others. In GSM, encryption is also done in the physical layer.

As the physical layer terminates at the BS, an important target for improved security in UMTS (compared with GSM) was to move the termination point of encryption further back into the network. For this reason, encryption is not done in the physical layer in UMTS.

2.1.3.3.2 MAC

The MAC layer converts transport channels into *logical channels*, which are characterized by *what* kinds of data are transferred. The main division of logical channels is the following:

- traffic channels—for U-plane information;
- control channels—for C-plane information.

Logical channels include the broadcast control channel, paging control channel, common control channel (CCCH), dedicated control channel (DCCH), common traffic channels and dedicated traffic channels. These logical channels are mapped into transport channels (e.g., the broadcast control channel can be mapped into either BCH or FACH and the dedicated traffic channel can be mapped into RACH, FACH, DCH, etc.).

The MAC layer contains, among others, the following functions:

- mapping logical channels into transport channels;
- choosing an appropriate transport format for each transport channel;
- identification of an addressed UE in common channels (this is the inband signalling referred to above);
- multiplexing of upper layer PDUs;
- traffic volume measurement.

The MAC layer also performs encryption in transparent RLC mode (e.g., in case of CS speech traffic). In this case the part that is encrypted is the MAC SDU (Signalling Data Unit) but the MAC header is not. The counter CFN consists of the least significant part of the encryption counter COUNT-C.

It is possible that several MAC PDUs are transmitted during the same Transmission Time Interval (TTI). In this case ciphering is not initialized in the middle of the TTI. Instead, the input parameter COUNT-C for the whole TTI is obtained from

the CFN of the first radio frame in the TTI. Then a long mask bit stream is generated and used to encrypt all the radio frames in the TTI.

2.1.3.3.3 RLC

The RLC layer provides the following services to the upper layers:

- transparent data transfer—upper layer PDUs are transmitted without any additional protocol information except possibly segmentation/reassembly of them;
- unacknowledged data transfer—upper layer PDUs are transmitted without guarantees of delivery, but with detection of transmission errors;
- acknowledged data transfer—upper layer PDUs are transmitted with guaranteed delivery, potential retransmissions are used for error-free delivery and double transmissions are also detected;
- maintenance of Quality of Service (QoS) as defined by upper layers;
- notification of irrecoverable errors to upper layers.

The most important RLC functions are: segmentation and reassembly of upper layer PDUs; concatenation of the first segment of an RLC SDU with the last segment of the previous RLC SDU into the same RLC PDU, adding padding bits in case no concatenation is possible; data transfer (transparent, unacknowledged or acknowledged); error correction; in-sequence delivery of upper layer PDUs; duplicate detection; RLC SQN check (in unacknowledged mode to provide the possibility of detecting errors when RLC PDUs are reassembled into RLC SDUs); protocol error detection and recovery.

The RLC layer also provides encryption in unacknowledged and acknowledged RLC modes, when ciphering is applied to the whole RLC PDU except the PDU header. The header consists of a SQN (7 bits) and an extension bit (making one octet) in the UM (Unacknowledged Mode) case, and of a SQN (12 bits) and 4 other bits (making two octets) in the AM (Acknowledged Mode) case. In the former case the extension bit of the header indicates whether a *length indicator* follows or the data. In the AM case the 4 bits that are included in the header in addition to the SQN indicate:

- whether the PDU contains control information or data;
- whether a status report (status PDU) is requested from the receiver;
- whether the length indicator follows or the data.

The structure of lower-layer data units is illustrated Figures 2.9–2.12.

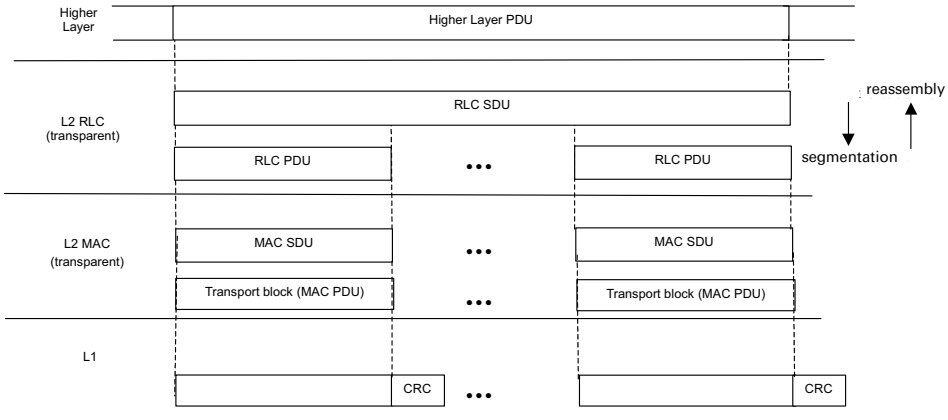

Figure 2.9 Both the MAC and RLC are in transparent mode
MAC = Medium Access Control; RLC = Radio Link Control; PDU = Protocol Data Unit; SDU = Signalling Data Unit; CRC = Cyclic Redundancy Check

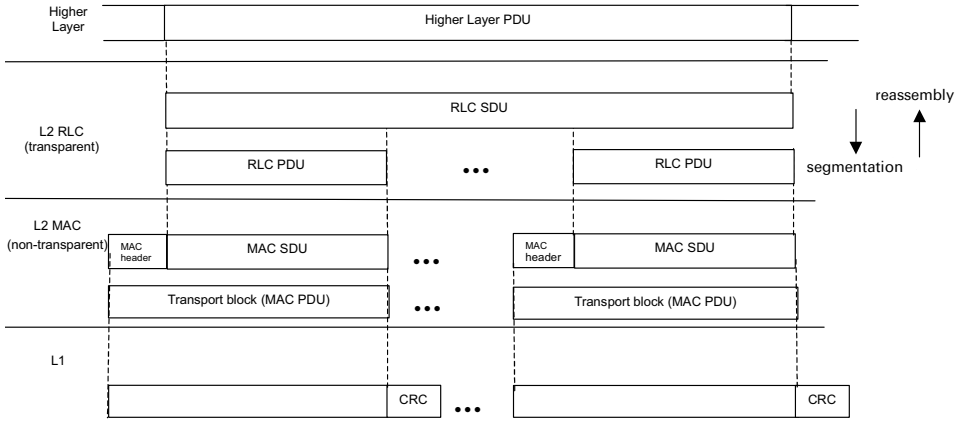

Figure 2.10 Transparent RLC mode and non-transparent MAC mode
RLC = Radio Link Control; MAC = Medium Access Control; PDU = Protocol Data Unit; SDU = Signalling Data Unit; CRC = Cyclic Redundancy Check

2.1.3.3.4 PDCP

The PDCP provides header compression/decompression of IP (Internet Protocol) traffic (e.g. for TCP (Transmission Control Protocol) or IP headers), among other things.

2.1.3.3.5 BMC

The BMC provides transmission and scheduling of BMC messages, and storage and delivery of cell broadcast messages.

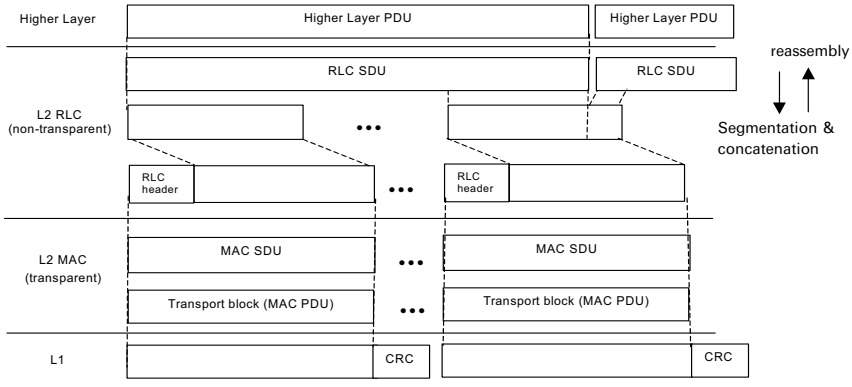

Figure 2.11 Non-transparent RLC mode and transparent MAC mode
RLC = Radio Link Control; MAC = Medium Access Control; PDU = Protocol Data Unit; SDU = Signalling Data Unit; CRC = Cyclic Redundancy Check

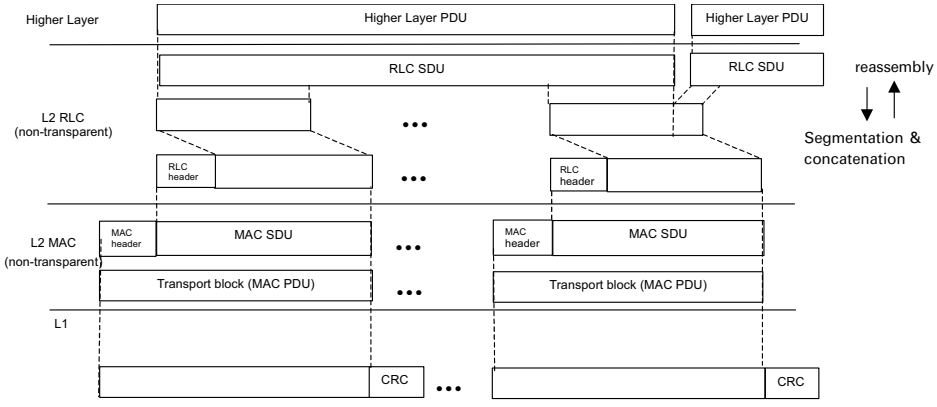

Figure 2.12 Both RLC and MAC are in non-transparent mode
RLC = Radio Link Control; MAC = Medium Access Control; PDU = Protocol Data Unit; SDU = Signalling Data Unit; CRC = Cyclic Redundancy Check

2.1.3.3.6 RRC

The RRC provides such functions as:

- broadcast of both NAS and Access Stratum (AS) information—for NAS information (e.g., general system-level information), the RRC provides scheduling, segmentation and repetition (AS information is typically cell-specific information about the radio environment);
- establishment, re-establishment, maintenance and release of RRC connections between the UE and RNC;

- establishment, reconfiguration and release of (U-plane) radio bearers requested by upper layers;
- RRC connection mobility functions, such as handovers, preparations for handovers to GSM, cell reselection;
- paging and notification requested by upper layers;
- control of requested QoS (appropriate radio resources have to be provided);
- control of UE measurements and related reporting.

The RRC also provides encryption control (i.e., it decides whether encryption is on or off between the UE and RNC) as well as executing *integrity protection* of both RRC-level signalling and higher layer signalling in the form of message authentication codes (MAC-I, the "I" stands for integrity of signalling data).

2.1.3.4 UE modes and identification

Inside UTRAN the UE can be in two different modes: *idle* or *connected*. After power has been switched on, the UE is in idle mode. When an RRC connection is established between the UE and RNC, the UE enters the connected mode and when the RRC connection is released the UE returns to idle mode.

In idle mode, the UE can only be identified by CN-level identities (i.e., by IMSI, TMSI or P-TMSI). In connected mode, it is also possible to use a UTRAN-level identity called Radio Network Temporary Identity (RNTI).

A necessary requirement for authentication is that the UE be identified first; hence, used identities play an important role in the overall security architecture of the system. Authentication is always done in NAS-level signalling and therefore is tied to IMSI, TMSI or P-TMSI. Integrity protection can be defined as *authentication of individual messages* and this type of authentication may be based on RNTI as well. As a consequence, the network must maintain the connection between NAS-level identities and RNTIs.

2.1.4 Integrity protection of RRC signalling

The purpose of integrity protection is to authenticate individual control messages. This is important, since separate authentication procedures only give assurance of the identities of the communicating parties at the time of the authentication. This leaves the door open for an attacker called "the man in the middle" to act as a simple relay and deliver all messages in their correct form until the authentication procedure is completely executed. After that, the man in the middle may begin to manipulate

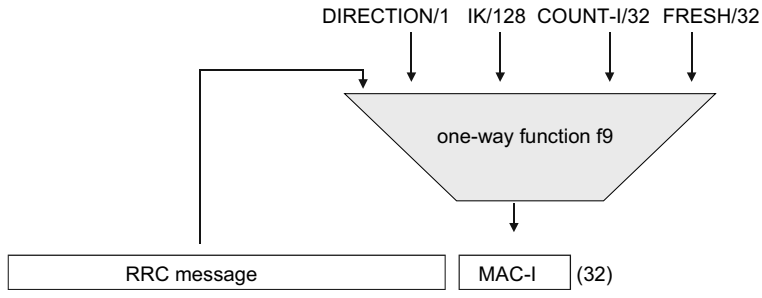

Figure 2.13 Message authentication code
IK = Integrity Key; RRC = Radio Resource Control; MAC-I = Message Authentication Code

messages freely. However, if messages are protected individually, deliberate manipulation of messages can be observed and false messages can be discarded.

Integrity protection is implemented at the RRC layer (i.e., between the terminal and RNC), just like the case for encryption. The IK is generated during the AKA procedure, again in the same way as the CK is generated. The IK is transferred to the RNC together with the CK in *security mode command*.

The integrity protection mechanism is based on the concept of a *message authentication code*, which is a one-way function controlled by the secret IK. The function is denoted by f9 and its output is MAC-I: a 32-bit, random-looking bit string. On the sending side, the MAC-I is computed and is appended to each RRC message. On the receiving side, the MAC-I is also computed and the result of the computation is checked to ensure it equals the bit string appended to the message. Any change in any of the input parameters affects the MAC-I in an unpredictable way.

The function f9 is depicted in Figure 2.13. Its inputs are IK, the RRC message itself, a counter COUNT-I, direction bit (uplink/downlink) and a random bit string called FRESH. The COUNT-I parameter resembles the corresponding counter for encryption. Its most significant part is an HFN that consists of 28 bits in this case, and the four least significant bits contain the RRC sequence number. Altogether, COUNT-I protects against replay of earlier control messages by guaranteeing that the set of values for input parameters is different for each run of the integrity protection function f9.

The algorithm for integrity protection is based on the same core function as encryption. Indeed, the KASUMI block cipher is used in a special mode to create a message authentication code function. A detailed description of the first 3GPP integrity protection algorithm is given in Section 6.8. At the time of writing, specification work has begun on defining another integrity algorithm for fallback purposes.

The FRESH parameter is chosen by the RNC and transmitted to the UE. It is needed to protect the network against a maliciously-chosen start value for COUNT-I. Indeed, the most significant part of HFN is stored in the USIM

between connections. An attacker could masquerade as the USIM and send a false value to the network, forcing the starting value of HFN to be too small. If the authentication procedure is not run and the old IK is brought into use, this would create a chance for the attacker to replay RRC signalling messages from earlier connections with recorded MAC-I values were the FRESH parameter not involved. By choosing FRESH randomly, the RNC is protected against such a replay attack (i.e., based on recording of earlier connections). As already explained, the ever-increasing counter COUNT-I protects against replay attacks that are based on recording during the *same* connection because FRESH stays constant over a single connection. From the terminal's point of view, it is still essential that the value COUNT-I never repeats itself even between different connections, because a false network could send an old FRESH value to the UE in order to try a replay attack in the downlink direction.

Note that radio bearer identity is not used as an input parameter for the integrity algorithm, although it is an input parameter for the encryption algorithm. Because there are also several parallel radio bearers for the control plane, this seems to leave room for possible replay of control messages that were recorded within the same RRC connection but on a different radio bearer. There is a historical reason for this state of affairs: at the time of freezing requirements for integrity protection algorithm design work, the specification for UTRAN contained only one signalling radio bearer.

Instead of changing the algorithm structure retrospectively, the following procedure was introduced in the integrity protection mechanism to remove the security hole. Radio bearer identity is always appended to the message when the message authentication code is calculated, although it is not transmitted with the message. So, not only does the radio bearer identity have an effect on the MAC-I value, we also have protection against replay attacks based on recordings from different radio bearers.

Clearly, there are RRC control messages whose integrity cannot be protected by the mechanism. Indeed, messages sent before the IK is in place cannot be protected. A typical example is the *RRC connection request* message sent from the UE. The following list contains all messages that are not integrity-protected:

- handover to UTRAN complete;
- paging type 1;
- push capacity request;
- physical shared channel allocation;
- RRC connection request;
- RRC connection set-up;

- RRC connection set-up complete;
- RRC connection reject;
- RRC connection release (CCCH (Common Control Channel) only);
- system information (broadcast information);
- system information change indication;
- transport format combination control (TM (Transparent Mode) DCCH only).

2.1.4.1 Periodic local authentication

The integrity protection mechanism in UTRAN is not applied for the U-plane for performance reasons. However, there is a specific (integrity-protected) control plane procedure that is used for *periodic local authentication*. As a result of this procedure, the amount of data sent during the RRC connection is checked. Hence, the *volume* of transmitted user data is integrity-protected and at the same time, the procedure provides local entity authentication.

Periodic local authentication is initiated by the RNC and triggered by some COUNT-C value that reaches a *critical value* (e.g., a certain bit in the HFN changes). Then the RNC sends a *counter check* message that contains the most significant part from each COUNT-C, corresponding to each active radio bearer. The UE compares the sent values with the most significant parts of its own COUNT-C values. All differences are reported back in a *counter check response* message. If the response message does not contain any values, then the procedure ends. If there are differences, the RNC may release the connection (in the event the differences cannot be accepted). The procedure is depicted in Figure 2.14.

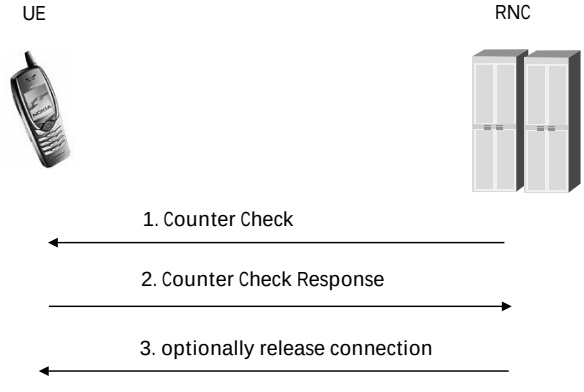

Figure 2.14 Periodic local authentication
UE = User Equipment; RNC = Radio Network Controller

Periodic local authentication gives protection against an attacker who tries to insert or delete data packets during uplink or downlink. The protection is especially important in case encryption is not in use. Note that in this case both the UE and the RNC need to maintain COUNT-C values despite the fact they are not used for encryption.

It is possible that the attacker could try and insert and delete the same number of packets in order to keep COUNT-C values synchronized. The legitimate user cannot stop this type of attack unless he or she notices a drop in service level.

2.1.4.2 Threats against UTRAN signalling

In this section we elaborate a bit more on the different types of threats that UTRAN signalling is exposed to. A simplified picture of the UTRAN link layers and network nodes involved is depicted in Figure 2.15.

The main signalling flows take place between the UE and the RNC. Signalling messages are exchanged at all three layers: RRC, RLC and MAC. The most important and sensitive are those on the RRC layer and their integrity is protected. Encryption provides protection for signalling in RLC and MAC.

In the rest of this section we give examples of the threats to signalling in each layer (the functions performed at each layer were listed in Section 2.1.3.3.

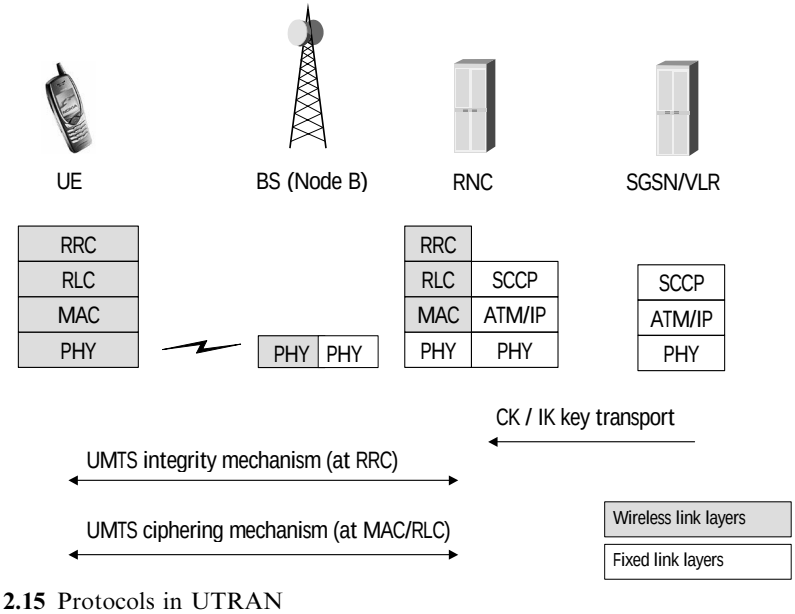

Figure 2.15 Protocols in UTRAN

UTRAN = UMTS Terrestrial Radio Access Network; UE = User Equipment; RRC = Radio Resource Control; RLC = Radio Link Control; MAC = Medium Access Control; PHY = physical layer; BS = Base Station; RNC = Radio Network Controller; SCCP = Signalling Connection Control Part; ATM = Asynchronous Transfer Mode; IP = Internet Protocol; CK = Cipher Key; IK = Integrity Key

- *Threats to MAC layer signalling*—
  - identification information sent by the UE over common transport channels can be tampered with;
  - access service class selection for RACH and CPCH (Common Packet Channel) transmission can be tampered with.
- *Threats to RLC functions*—
  - if RLC PDU (Protocol Data Units) headers are tampered with, duplicate detection is made impossible;
  - attackers can tamper with flow control messages and in this manner disturb the traffic flow and deteriorate the QoS of the victim UE;
  - attackers can tamper with SQNs, thus preventing the detection of corrupt RLC SDUs (Signalling Data Units).
- *Threats to RRC functions*—some of the threats to the RRC are addressed by the integrity protection mechanism, but as mentioned earlier in this section not all RRC messages can be protected—
  - broadcast of information provided by the CN cannot be integrity-protected;
  - broadcast of (typically cell-specific) information provided by the AS cannot be integrity-protected;
  - an attacker UE can try to hijack the connection by tampering with RRC connection re-establishment requests sent by the victim UE;
  - significant damage can be caused by tampering measurements made and sent by the victim UE;
  - paging information and target addresses can be tampered with;
  - measurement information sent by the victim UE can be tampered with;
  - idle mode measurements sent by the victim UE can be tampered with, thus forcing it to a non-suitable cell;
  - non-optimal performance can be caused by tampering with configuration messages sent by the RRC to the victim UE.

2.1.5 Set-up of UTRAN security mechanisms

The use of encryption and integrity protection is vital for the security of UTRAN. To guarantee that mechanisms cannot be bypassed, it is important to define exactly

how they are turned on after communication has been established. As mentioned earlier, the use of encryption is not mandatory in the system, but there must not be a way of avoiding the application of integrity protection.

In this section we describe how encryption and integrity protection are activated at the time of connection set-up.

2.1.5.1 Negotiation of the algorithms

Assume the UE wants to establish a connection with the network. First, a *classmark* that indicates the capabilities of the UE is sent to the network. These capabilities always include support for different encryption and integrity algorithms. Because this information is transferred at the very beginning of the connection, though the purpose of the information is to establish security later, there cannot be any protection for the transmitted classmark at this point. This problem is addressed by rechecking classmark information at a later stage of the security set-up procedure. Based on the received classmark, the network decides which algorithms to use:

- if there are no integrity protection algorithms in common, then the connection is shut down immediately;
- if there are no encryption algorithms in common, then the network may establish the connection without encryption.

All UEs compliant with 3GPP Release 1999 standards must support the integrity algorithm UIA1, hence the first case above should never occur.

Because the UE can establish connections in CS and PS domains independently of each other, in principle it could be possible for the network to choose different algorithms for different domains. However, this is not permitted. The reason for this is that security algorithms are implemented in the RNC, which is a common element of both CN domains. Therefore, using different algorithms for different domains would be an unnecessary burden for the RNC (and for the UE as well).

2.1.5.2 Existing parameters in USIM

When a new connection is established, some parameters are inherited from the previous connection (failing that, some links to the previous connection are needed).

The UE has stored the security keys that have been used for both domains (up to four keys in total). The UE has also stored the value of START for both CS and PS domains to the USIM. At the same time whether either of these values have reached

the maximal allowed value, called THRESHOLD, has been checked. The latter is configured to the USIM and provides a means of limiting security key lifetimes. If START has reached THRESHOLD for a CN domain, then the CK and IK for this domain are deleted from the USIM and START is set to be equal to THRESHOLD.

At the beginning of a new connection, START values and security keys are read from the USIM. The Key Set Identifier (KSI) is associated with a pair of security keys: the CK and IK that were generated during the same run of an AKA procedure. The KSI consists of three bits: the value "111" is reserved just in case there are no valid keys in the USIM. The value of KSI wraps around fairly often, every seventh time, but a period of this length is enough to remove the risk of ambiguity in practice. The values of START and KSI are transmitted to the network as part of the first messages as soon as the connection is made.

2.1.5.3 Security mode set-up procedure

We now describe those steps that are followed when integrity protection (and possibly encryption) are turned on. Integrity protection is not turned on:

1. if the connection is only for periodic location registration (without any change in registration information);
2. if the connection is only for indicating deactivation from the UE;
3. if authentication fails and, therefore, connection is immediately shut down;
4. if the connection is for an *emergency call* and there is neither a USIM nor a SIM (Subscriber Identity Module) in the UE.

Figure 2.16 describes the messages and security-relevant information elements that are transferred between the UE, RNC and SGSN/VLR during the set-up procedure.

Note that the procedure explicitly defines which message is the first to be integrity-protected at both uplink and downlink. On the other hand, this is not the case for encryption. At uplink the first encrypted message is the first message sent after the *security mode complete* message has been sent. At downlink the first encrypted message is the one sent after the *RANAP security mode complete* message has been sent to the CN. Because there may be messages in different layers waiting to be sent at the same time, it is not easy to decide which message should be the very first to be encrypted. For this purpose, a specific *ciphering activation time* parameter is exchanged between the UE and the RNC.

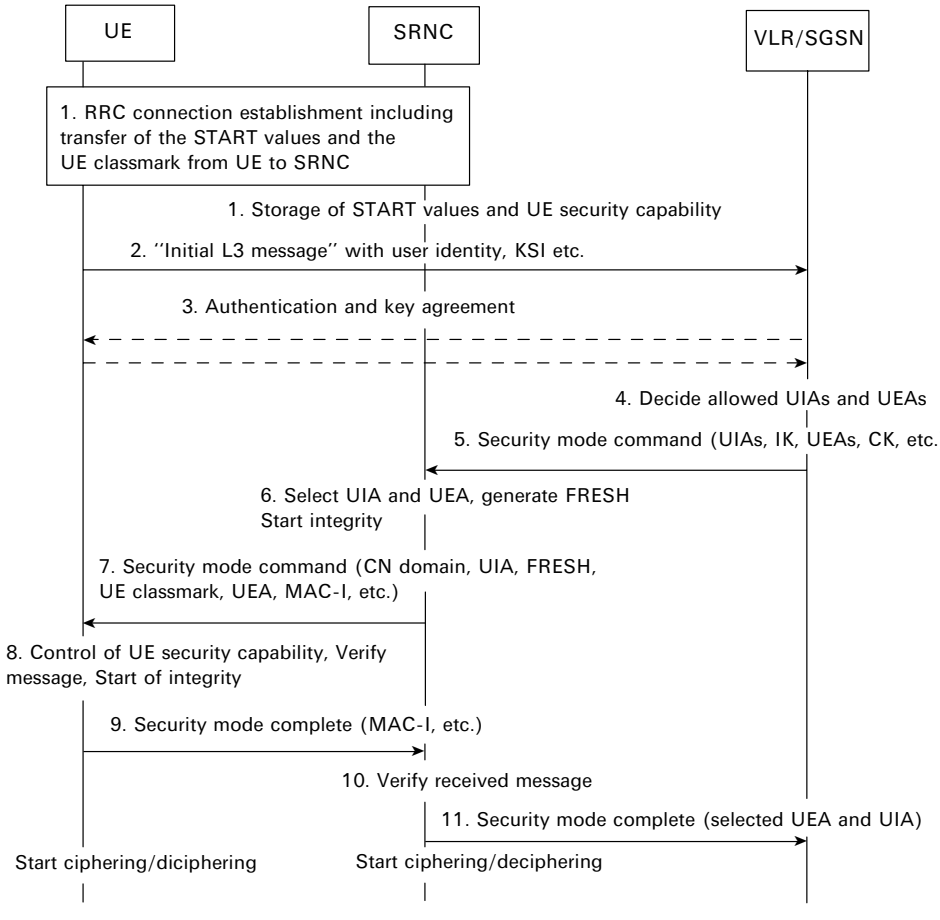

Figure 2.16 Security set-up
SRNC = Serving RNC; RNC = Radio Network Controller; VLR/SGSN = Visitor Location Register/Serving GPRS Support Node; GPRS = General Packet Radio Service; RRC = Radio Resource Control; UE = User Equipment; KSI = Key Set Identifier; UIA = UMTS Integrity Algorithm; UEA = UMTS Encryption Algorithm; IK = Integrity Key; CK = Cipher Key; CN = Core Network; MAC-I = Message Authentication Code

2.1.5.4 Security parameters for a new connection

If the AKA procedure is not carried out during connection establishment, then ''old'' security keys are used for the new connection. The COUNT-C and COUNT-I counter parameters are also initialized with the START value and the HFN is first initialized using START for the 20 most significant bits. The remaining HFN bits are set to 0. Note that different layers use HFNs of different lengths. The remaining COUNT bits are obtained from the layer-specific counters that are used for other purposes, in addition to security.

If the AKA procedure is carried out, then newly generated keys are used and START values are set to 0 at commencement of protection.

During an ongoing connection, START values are maintained in the ME: they consist of the greatest number obtained when comparing the 20 most significant bits of each COUNT value in each radio bearer that is in use for the CN domain in question.

In the U-plane, both the CS domain and PS domain use their own keys for protection algorithms, but the same *signalling* radio bearers are used for both domains. This implies that these signalling bearers have to use shared keys as well. Following general security principles, the keys generated most recently are used, regardless of whether they are for the CS domain or for PS domain. As a consequence, it is possible that protection keys may need to be changed for ongoing signalling connection.

2.1.6 Summary of access security in the CS and PS domains

We conclude Section 2.1 by presenting a schematic overview of the most important access security mechanisms and their relationships with each other. For the sake of clarity, many parameters are not shown in Figure 2.17 (e.g., HFN and FRESH are important parameters that are transmitted between different elements, but they are omitted from the figure).

2.2 Interworking with GSM

The UMTS CN is a straight evolution from that of GSM. The radio interfaces are completely different in both systems, but the early terminals still support both, allowing roaming from one system to another and, furthermore, handovers between the systems. As the security features in the two systems are different, it is not an easy task to define how security is managed during interoperation.

A smooth transition is needed from a pure GSM network to a mixed network that has wide area GSM coverage, enhanced by WCDMA islands. To enable this, the decision was taken that access to UTRAN would be possible with old SIM cards. Indeed, a user can use a 3G terminal without the need to change his or her smart card. The downside is a lower level of security: when a SIM card is used to access UTRAN, no authentication of the network is possible, because the card only provides 64 bits of key material (in the form of K_c) per authentication, while in the UTRAN side two 128-bit keys are needed. For this purpose, the 64-bit key K_c is *expanded* into 256 bits by using specific *conversion functions*. This procedure makes it possible to apply encryption and integrity protection in UTRAN when SIM cards

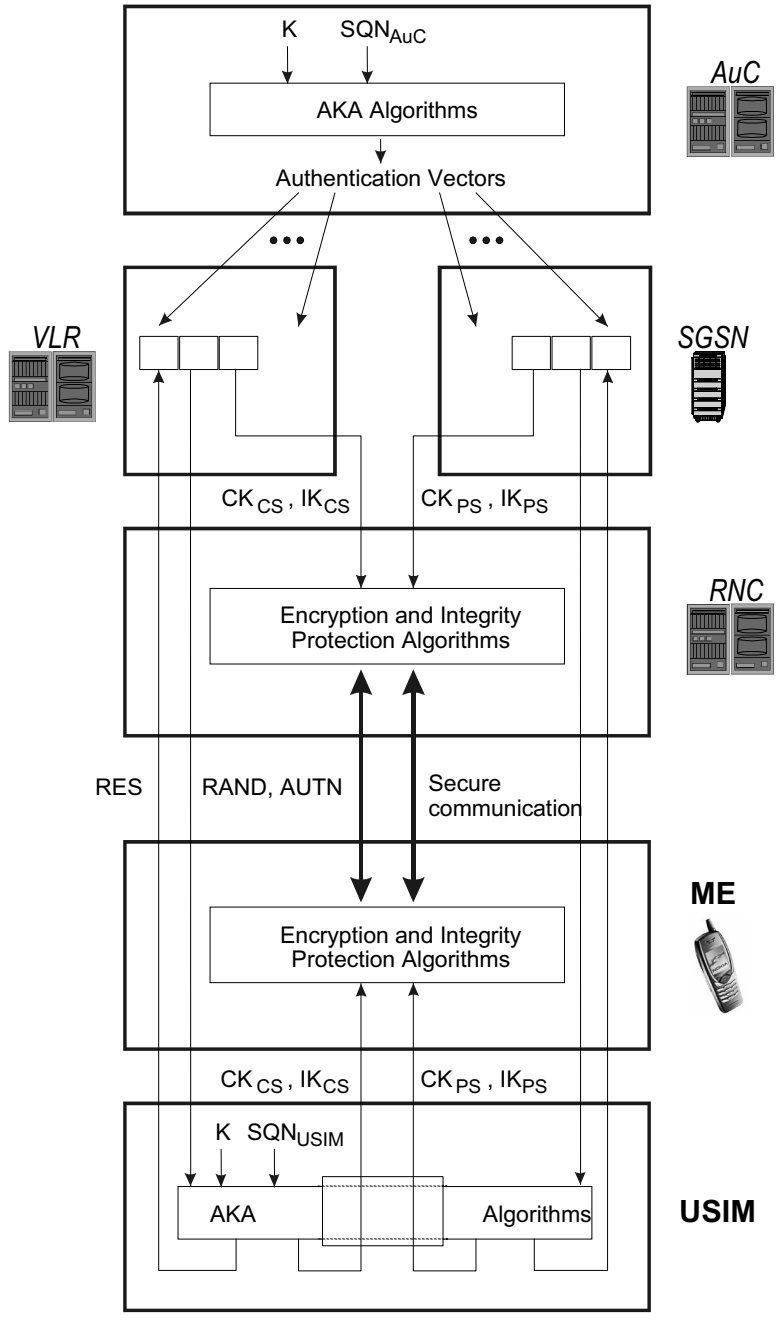

Figure 2.17 UMTS access security summary
SQN = sequence number; AuC = Authentication Centre; AKA = Authentication and Key Agreement; VLR = Visitor Location Register; SGSN = Serving GPRS Support Node; GPRS = General Packet Radio Service; CK = Cipher Key; CS = Circuit Switched; IK = Integrity Key; PS = Packet Switched; RNC = Radio Network Controller; RES = user response; RAND = random number; AUTN = authentication token; ME = Mobile Equipment; USIM = Universal Subscriber Identity Module

are used. However, the resulting security level can only be comparable with that of GSM because conversion functions only make keys longer *nominally*.

Another instance of interworking occurs when a 3G subscriber with a proper USIM needs to gain access outside WCDMA coverage. We then need to *compress* the longer keys provided by the USIM to 64 bits in order to use GSM encryption. Conversion functions are described in Section 8.8.

2.2.1 Interworking scenarios

In 3GPP technical report TR 31.900 [8], all possible interworking scenarios in a mixed 2G/3G environment are systematically studied. There are five basic entities in the system: the security module, terminal, radio network, serving CN and the home network. Each of these entities could be classified as either 2G or 3G. Some of these entities are already classified as mixed cases, but from the security point of view it is useful to define a clear-cut division between 2G and 3G for each entity:

- The security module can either be a SIM card (2G case) or a UICC (3G case). It is important to note that a UICC may contain a SIM *application* in addition to a USIM *application*.
- The ME (Mobile Equipment) is classified as 2G if it supports exclusively the GSM RAN and interworks with either a SIM card or a SIM application in a UICC. Otherwise the ME is 3G: in which case it supports either UTRAN only or both GSM radio access and UMTS radio access. The 3G ME interworks with either a USIM application in a UICC or with a SIM.
- The division for RANs is clear: the GSM Base Station Subsystem (BSS) is used for 2G and UTRAN for 3G.
- The SN VLR/SGSN is classified as 2G if it supports exclusively GSM authentication and can be attached exclusively to a GSM BSS. Otherwise, the VLR/SGSN is 3G (i.e., it supports both the UMTS AKA and GSM AKA, and can be attached to UTRAN and/or a GSM BSS). Furthermore, a 3G SN supports conversion functions.
- The HLR/AuC is 2G if it supports exclusively authentication triplet generation for 2G subscriptions. A 3G HLR/AuC supports authentication quintet generation for 3G subscriptions and conversion functions to support GSM authentication. It may also support pure triplet generation for 2G subscriptions.

Altogether, we have $2^5 = 32$ different combinations of 2G/3G entities. If we also count the SIM application in the UICC as a third possible case for the security module, we have $3 \times 16 = 48$ cases (all these theoretical combinations are listed

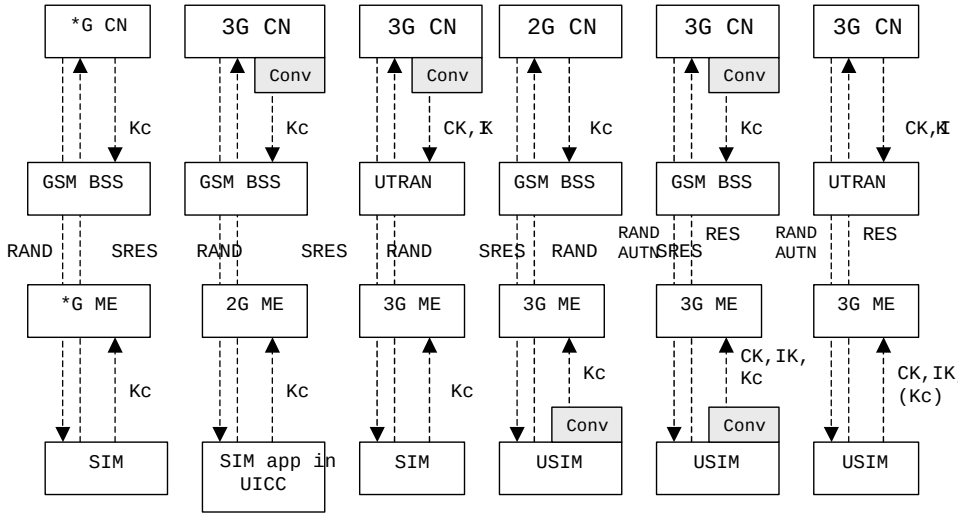

Figure 2.18 Interworking scenarios
CN = Core Network; CK = Cipher Key; IK = Integrity Key; GSM BSS = Global System for Mobile Base Station Subsystem; UTRAN = UMTS Terrestrial Radio Access Network; RAND = random number; SRES = user response in GSM; AUTN = authentication token; RES = user response; ME = Mobile Equipment; SIM = Subscriber Identity Module; UICC = Universal Integrated Circuit Card; USIM = Universal SIM

and analysed in [8]). In this book, we only highlight those scenarios that are essentially different from each other. To do this, we combine the CN entities in Figure 2.18 and say that the CN is 3G if either the SN or home network (or both) are 3G; otherwise, we say the CN is 2G. Six essentially different cases are depicted.

2.2.2 Cases with SIM

We have three essentially different cases where SIM is used as an access module.

2.2.2.1 SIM and GSM BSS

If SIM is used to access a GSM BSS, then we have a pure GSM case from the security point of view. It does not matter whether the ME is 3G or 2G and the same is true for the CN. As far as security features are concerned, we have 2G authentication and 2G encryption.

2.2.2.2 SIM application and GSM BSS

A slight variant of the previous case is when a UICC is used in 2G ME, when the RAN must be a GSM BSS. However, when SIM application is used in the UICC,

exactly the same security features are carried out as in the previous case. In the CN, conversion functions must be available to produce authentication triplets. As far as security features are concerned, we have 2G authentication and 2G encryption.

2.2.2.3 SIM and UTRAN

In this case both the CN and the ME must be 3G, as they both support UTRAN. The GSM encryption key K_c is expanded to CK and IK by conversion functions both in the CN and in the ME. As far as security features are concerned, we have 2G authentication and both encryption and integrity protection are 3G but accessed by a 2G key.

2.2.3 Cases with USIM

We have again three essentially different cases when USIM is used as the security module. In all cases the ME must be 3G, since it must support USIM. For a similar reason, the home network must be 3G.

2.2.3.1 USIM and GSM BSS and 2G SN

Here the home network must produce authentication triplets with conversion functions because the SN can only support triplets. On the terminal side, USIM itself applies a conversion function to derive the GSM encryption key K_c. As far as security features are concerned, we have 2G authentication and 2G encryption.

2.2.3.2 USIM and GSM BSS and 3G SN

Once authentication vectors can be used, even if the RAN is only 2G, it is possible to run the UMTS AKA, as this protocol is transparent to the radio network. However, CK and IK cannot be used. Thus, a conversion function has to be used both in the USIM and in the CN to generate the GSM encryption key K_c. Note that CK and IK are transferred to the ME to support potential future handovers to UTRAN. On the security side, we now have 3G authentication but 2G encryption.

2.2.3.3 Pure 3G case

In this case all elements are 3G and the full set of UMTS security features are in use. Note that the converted GSM key K_c may be derived for potential future handovers to the GSM BSS. It would of course be technically possible to run GSM authentication in this case as well. Indeed, USIM has no way of knowing whether the ME is

connected to UTRAN or GSM BSS. Therefore, the ME has to abort GSM authentication attempts in case it is connected to UTRAN and contains a USIM. It is only in this case that we have all the 3G security features: 3G authentication, 3G encryption and 3G integrity protection.

2.2.4 Handovers from one system to another

The concept of a handover is different for CS services than PS services. In PS it is much easier to send packets via different cells, but for a CS bit stream the transition from one cell to another has to be planned more carefully. This difference is also visible with inter-RAT (Radio Access Technology) handovers.

2.2.4.1 CS handovers from UTRAN to GSM BSS

The encryption algorithm must be changed during handover from UTRAN to the GSM BSS. The WCDMA algorithm UEA (UMTS Encryption Algorithm) is replaced by the GSM A5 algorithm and the UTRAN CK is replaced by a converted K_c. Information about supported/allowed GSM algorithms together with the key has to be transferred within the system infrastructure before the handover can take place. Of course, integrity protection is stopped at handover to GSM BSS.

2.2.4.2 CS handovers from GSM BSS to UTRAN

If the handover is done from GSM BSS to UTRAN, then the encryption algorithm is changed from A5 to UEA. Before the handover, GSM BSS requests UE to send information about its UTRAN security capabilities together with the associated parameters (e.g., CK, IK, START). This information is transferred within the system infrastructure to the target RNC before encryption and integrity protection can start on the UTRAN side.

2.2.4.3 Intersystem change for PS services

There are a couple of notable differences between intersystem handovers for CS services and corresponding intersystem changes to PS services. First, GPRS (General Packet Radio Service) encryption terminates in the CN and, therefore, transfer of keys is somewhat simpler. Second, there is a difference when the CN changes in addition to the radio network. If the UE moves to the area of a new MSC

(Mobile Switching Centre)/VLR, the old MSC/VLR still remains as the *anchor point* for the call. However, if the UE moves to the area of a new SGSN, then this new SGSN also becomes the anchor point for the connection.

2.3 Additional Security Features in Release 1999

There also exist other security features in the Release 1999 specification set: some are directly inherited from GSM as such and some are added for the first time in Release 1999. Let us now take a brief look at the individual features. Detailed information can be found in the relevant 3GPP specifications (see Table 1.2 for specifications under responsibility of the 3GPP security group SA3). Some relevant specifications (e.g., MExE (Mobile Execution Environment) or LCS (location services)) are not in this list, because security issues only partially cover them.

2.3.1 Ciphering indicator

There is a specific *ciphering indicator* in ME that is used to show the user whether encryption is applied or not, thus providing some visibility of the security mechanisms to the user. Note that although the use of ciphering is highly recommended it is still optional for the UMTS network. Details of the indicator are left to be implementation-specific and the best way to inform the user is very much dependent on the characteristics of the terminal itself (e.g., different display types may utilize different types of indicators).

In general, it is important that the security level is not dependent on whether the user is doing active checks. Nevertheless, for some specific actions, users may appreciate visibility of active security features.

2.3.2 Identification of the UE

In the GSM system, the ME can be identified by its International Mobile Equipment Identity (IMEI). This identity is not directly associated with the user because a SIM card may be moved from one terminal to another. There are, however, important features in the network that can only be based on the value of IMEI (e.g., it is possible to make *emergency calls* with a terminal without a SIM card). The only identification method in this case is to require the terminal to provide its IMEI, very useful also for tracing stolen phones.

This feature is carried over to the UMTS system as well. There are no mechanisms in either the GSM or UMTS that actually authenticate the provided IMEI. So,

protection methods for IMEI have to be based solely on the terminal side: it must be made difficult for the terminal to be modified in such a way that it provides a wrong value for IMEI when requested by the network.

2.3.3 Security for Location Services (LCS)

The mobile network has to be able to trace users while they are on the move. Otherwise, it would not be possible to serve them. In addition to the needs of the network itself, there are clearly many services that can benefit from knowing the position of users (e.g., a hungry user may want to know which restaurants are closest to his or her current position).

Location information is clearly sensitive. People are not comfortable with the idea that they could be traced at any time. Security mechanisms have been defined to protect against leakage of location information to unauthorized parties. The *privacy profile* concept plays a central role here: the user must be in charge of who know about her or his whereabouts.

2.3.4 User-to-USIM authentication

This feature also carries over from the GSM system to the UMTS system and is based on a Personal Identification Number (PIN) known only to the user and the USIM. The user has to be able to give the PIN, which is 4–8 digits long, to the USIM before further access to the latter is granted. It must be admitted though that mobile phones are frequently stolen while they are in fully operational mode and therefore this feature does not act as a defence against theft, because authentication has already happened before the phone is stolen.

2.3.5 Security in the USIM application toolkit

Similarly to GSM, it is possible to build applications that are executed in the USIM by using a feature called the (U)SIM application toolkit, which grants the home operator the possibility, among others, to send messages directly to the USIM. The USIM application toolkit also specifies what kind of protection may be provided for this message transfer. Many details of protection mechanisms are implementation-specific.

2.3.6 Mobile Execution Environment (MExE)

The 3GPP has specified a framework for running applications in the ME (see [1]). Several different technologies are included in the specification (e.g., WAP (Wireless

Application Protocol) and Java). A great deal of the specification effort has been devoted to make the environment secure. In particular, security issues with downloaded applications have been addressed in [1]. Protection mechanisms are partially based on public key cryptography.

2.3.7 Lawful interception

In most countries, legislation and regulations set the requirement that authorities must have a way of accessing sensitive information (e.g., law enforcement has to be able to listen to the phone calls of suspected criminals or to find out where the suspects are (or were) at a certain moment). Such information is also used as evidence in court cases.

In the GSM, the lawful interception functionality was later added to an already existing complete system. Clearly, it is more effective to have standardized mechanisms in that kind of situation and this is why, in the UMTS, that the lawful interception features and the interfaces needed for them have been standardized as an integral part of the system.

When new elements and services are added to the 3GPP system, any lawful interception aspects are taken into account from the beginning. In this way it is possible to provide effective standardized solutions for this special purpose as well.