3GPP TSG SA WG 3 Meeting #12
Stockholm, Sweden, 11-14 April 2000

Document S3-000269
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

33.102	**CR**		Current Version:	3.4.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: SA#7 *list expected approval meeting # here ↑*	for approval	X	strategic		*(for SMG use only)*
	for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME		UTRAN / Radio		Core Network	X

Source:	Siemens Atea			**Date:**	11 April 2000
Subject:	Removal of network domain security				
Work item:	Security				

Category: *(only one category shall be marked with an X)*			**Release:**	
	F	Correction		Phase 2	
	A	Corresponds to a correction in an earlier release		Release 96	
	B	Addition of feature		Release 97	
	C	Functional modification of feature	X	Release 98	
	D	Editorial modification		Release 99	X
				Release 00	X

Reason for change:	Decision by SA#6
Clauses affected:	3.3, 5.2, 6.3.1, 6.7.3.1, 7, Annex E

Other specs affected:			
	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:	

help.doc

<--------- double-click here for help and instructions on how to create a CR.

3.3 Abbreviations

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

AK	Anonymity Key
AKA	Authentication and key agreement
AMF	Authentication management field
AUTN	Authentication Token
AV	Authentication Vector
CK	Cipher Key
CKSN	Cipher key sequence number
CS	Circuit Switched
EMSI	Encrypted Mobile Subscriber Identity
EMSIN	Encrypted MSIN
~~$D_{SK(X)}$(data)~~	~~Decryption of "data" with Secret Key of X used for signing~~
~~$E_{KSXY(i)}$(data)~~	~~Encryption of "data" with Symmetric Session Key #i for sending data from X to Y~~
~~$E_{PK(X)}$(data)~~	~~Encryption of "data" with Public Key of X used for encryption~~
GI	Group Identifier
GK	Group Key
~~Hash(data)~~	~~The result of applying a collision-resistant one-way hash-function to "data"~~
HE	Home Environment
HLR	Home Location Register
IK	Integrity Key
IMSI	International Mobile Subscriber Identity
~~IV~~	~~Initialisation Vector~~
~~KAC_X~~	~~Key Administration Centre of Network X~~
~~$KS_{XY}(i)$~~	~~Symmetric Session Key #i for sending data from X to Y~~
KSI	Key Set Identifier
KSS	Key Stream Segment
LAI	Location Area Identity
~~MAP~~	~~Mobile Application Part~~
MAC	Message Authentication Code
MAC-A	The message authentication code included in AUTN, computed using f1
MS	Mobile Station
MSC	Mobile Services Switching Centre
MSIN	Mobile Station Identity Number
MT	Mobile Termination
~~NE_X~~	~~Network Element of Network X~~
PS	Packet Switched
P-TMSI	Packet-TMSI
Q	Quintet, UMTS authentication vector
RAI	Routing Area Identifier
RAND	Random challenge
~~RND_X~~	~~Unpredictable Random Value generated by X~~
SQN	Sequence number
SQN_{UIC}	Sequence number user for enhanced user identity confidentiality
SQN_{HE}	Sequence number counter maintained in the HLR/AuC
SQN_{MS}	Sequence number counter maintained in the USIM
SGSN	Serving GPRS Support Node
SIM	(GSM) Subscriber Identity Module
SN	Serving Network
T	Triplet, GSM authentication vector
TE	Terminal Equipment
TEMSI	Temporary Encrypted Mobile Subscriber Identity used for paging instead of IMSI
~~Text1~~	~~Optional Data Field~~
~~Text2~~	~~Optional Data Field~~
~~Text3~~	~~Public Key algorithm identifier and Public Key Version Number (eventually included in Public Key Certificate)~~
TMSI	Temporary Mobile Subscriber Identity
~~TTP~~	~~Trusted Third Party~~
UE	User equipment

UEA	UMTS Encryption Algorithm
UIA	UMTS Integrity Algorithm
UICC	UMTS IC Card
UIDN	User Identity Decryption Node
USIM	User Services Identity Module
VLR	Visitor Location Register
~~X~~	~~Network Identifier~~
XEMSI	Extended Encrypted Mobile Subscriber Identity
XRES	Expected Response
~~Y~~	~~Network Identifier~~

5.2 Network domain security

5.2.1 ~~Entity authentication~~Void

~~The following features with respect to authentication of network elements are provided:~~

- ~~**authentication mechanism agreement:** the property that two network entities can securely negotiate the mechanism for authentication that they shall use subsequently;~~
- ~~**network element authentication:** the property that a network element corroborates the identity of another network element it wants to communicate with;~~

~~This feature ensures that no malicious operational or maintenance commands can be injected into a network domain by an intruder. It provides network elements, in particular network elements belonging to different network operators, with the possibility to corroborate each other's identities before exchanging data.~~

~~This goal may be achieved either by an explicit or implicit entity authentication mechanism, to be performed each time data are exchanged between two network entities. Implicit authentication is realised by exchanging encrypted messages only, so that only an entity in possession of a certain shared key can make use of the data. The shared keys may be distributed among the network elements of a single operator in a manner outlined in Annex D.~~

~~Explicit authentication mechanisms can be achieved by asymmetrically based protocols (e.g. by using digital signatures) or by symmetric (e.g. challenge-response) protocols. Again, for explicit symmetric authentication, the necessary keys may be distributed as proposed in Annex E.~~

5.2.2 ~~Data confidentiality~~Void

~~The following security features are provided with respect to confidentiality of data exchanged between network elements:~~

- ~~**cipher algorithm agreement:** the property that two network elements can securely negotiate the algorithm that they shall use subsequently;~~
- ~~**cipher key agreement:** the property that two network elements agree on a cipher key that they may use subsequently;~~
- ~~**confidentiality of exchanged data:** the property that data exchanged between two network elements cannot be eavesdropped;~~

~~In case authentication data can be eavesdropped in the network domain, serious fraud problems will arise. Therefore, these features are needed to ensure the confidentiality of sensitive data, e.g. authentication or other subscriber data inside the network domain. The first two features may be realised in course of an authentication mechanism performed by the network elements; the agreed cipher key is then used for securing signalling and user data by means of the agreed cipher algorithm.~~

5.2.3 ~~Data integrity~~Void

~~The following security features are provided with respect to integrity of data exchanged between two network elements:~~

- ~~**integrity algorithm agreement:** the property that two network elements can securely negotiate the integrity algorithm that they shall use subsequently;~~
- ~~**integrity key agreement:** the property that two network elements agree on an integrity key that they may use subsequently;~~
- ~~**data integrity and data origin authentication of signalling data:** the property that the receiving network element is able to verify that signalling data has not been modified in an unauthorised way since it was sent by the sending element and that the data origin of the signalling data received is indeed the one claimed;~~

~~The feature data integrity of signalling data ensures that operation and maintenance commands or user data exchanged between two network elements cannot be modified by an intruder without being detected, while the third feature ensures that no malicious operational or maintenance commands can be injected into a network domain by an intruder~~

~~The first two features may be realised in course of an authentication mechanism performed by the network entities involved; the agreed integrity key is then used for securing integrity of the exchanged data by means of the agreed integrity algorithm.~~

5.2.4 Fraud information gathering system

NOTE: Some feature will be provided which will allow fraud information to be exchanged between 3GMS providers according to time constraints that yet have to be defined.

6.3.1 General

The mechanism described here achieves mutual authentication by the user and the network showing knowledge of a secret key K which is shared between and available only to the USIM and the AuC in the user's HE. In addition the USIM and the HE keep track of counters SEQ_{MS} and SEQ_{HE} respectively to support network authentication.

The method was chosen in such a way as to achieve maximum compatibility with the current GSM security architecture and facilitate migration from GSM to UMTS. The method is composed of a challenge/response protocol identical to the GSM subscriber authentication and key establishment protocol combined with a sequence number-based one-pass protocol for network authentication derived from the ISO standard ISO/IEC 9798-4 (section 5.1.1).

An overview of the mechanism is shown in Figure 5.

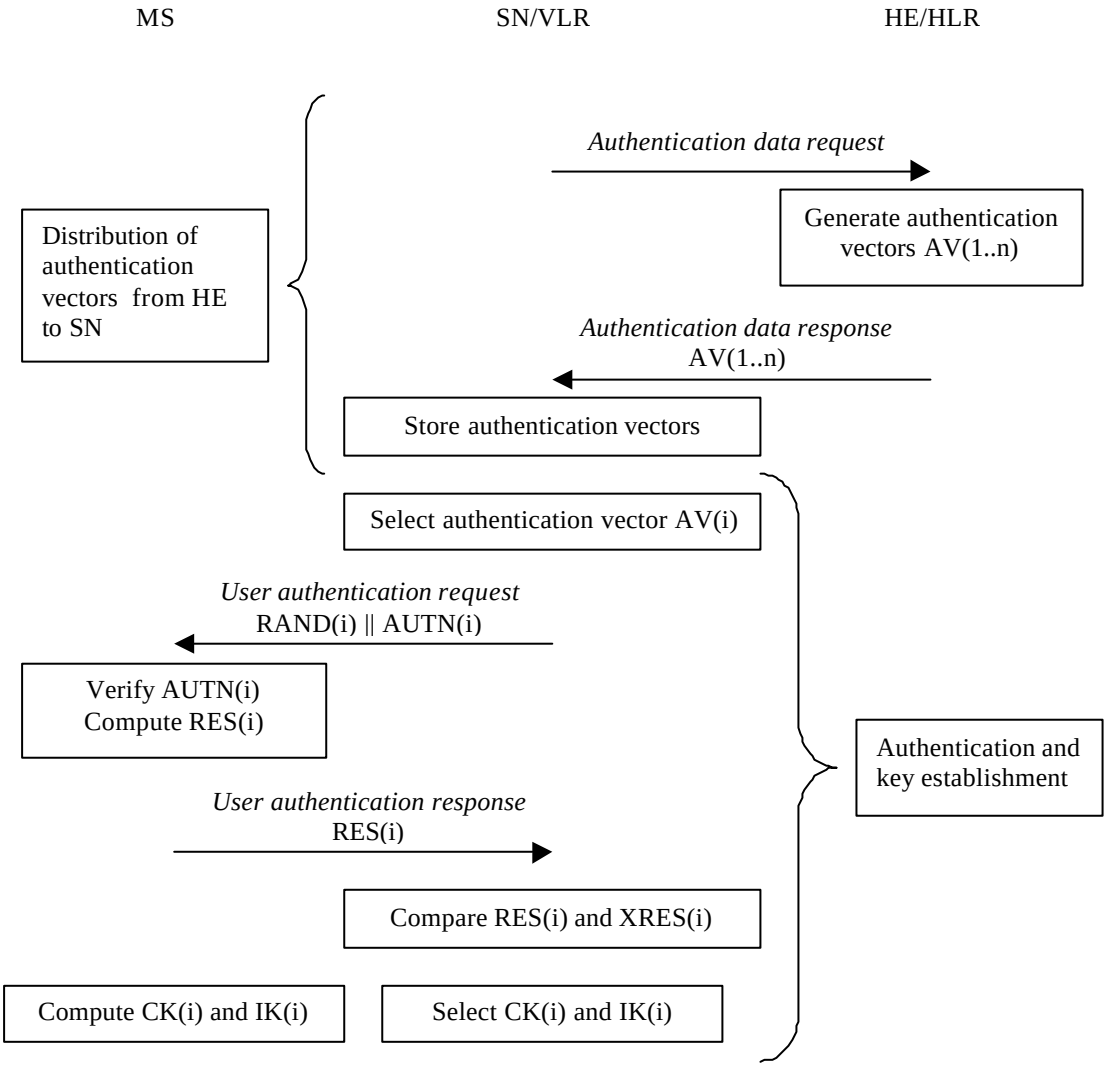


Figure 5: Authentication and key agreement

Upon receipt of a request from the VLR/SGSN, the HE/AuC sends an ordered array of *n* authentication vectors (the equivalent of a GSM "triplet") to the VLR/SGSN. Each authentication vector consists of the following components: a random number RAND, an expected response XRES, a cipher key CK, an integrity key IK and an authentication token AUTN. Each authentication vector is good for one authentication and key agreement between the VLR/SGSN and the USIM.

When the VLR/SGSN initiates an authentication and key agreement, it selects the next authentication vector from the array and sends the parameters RAND and AUTN to the user. The USIM checks whether AUTN can be accepted and, if so, produces a response RES which is sent back to the VLR/SGSN. The USIM also computes CK and IK. The VLR/SGSN compares the received RES with XRES. If they match the VLR/SGSN considers the authentication and key agreement exchange to be successfully completed. The established keys CK and IK will then be transferred by the

USIM and the VLR/SGSN to the entities which perform ciphering and integrity functions.

VLR/SGSNs can offer secure service even when HE/AuC links are unavailable by allowing them to use previously derived cipher and integrity keys for a user so that a secure connection can still be set up without the need for an authentication and key agreement. Authentication is in that case based on a shared integrity key, by means of data integrity protection of signalling messages (see 6.4).

The authenticating parties shall be the AuC of the user's HE (HE/AuC) and the USIM in the user's mobile station. The mechanism consists of the following procedures:

A procedure to distribute authentication information from the HE/AuC to the VLR/SGSN. This procedure is described in 6.3.2. The VLR/SGSN is assumed to be trusted by the user's HE to handle authentication information securely. It is also assumed that the intra-system links between the VLR/SGSN to the HE/AuC are adequately secure. ~~Mechanisms to secure these links are described in clause 7.~~ It is further assumed that the user trusts the HE.

A procedure to mutually authenticate and establish new cipher and integrity keys between the VLR/SGSN and the MS. This procedure is described in 6.3.3.

A procedure to distribute authentication data from a previously visited VLR to the newly visited VLR. This procedure is described in 6.3.4. It is also assumed that the links between VLR/SGSNs are adequately secure. ~~Mechanisms to secure these links are described in clause 7.~~

6.7.3.1 General case

We assume that signalling links within the network are confidentially protected on a link-by-link basis. In particular, we assume that the UE to RNC signalling links are protected using access link security domain keys (see clause 6). ~~We also assume that VLR to RNC signalling links and core network signalling links are protected using network security domain keys (see clause 7)~~. Note that if network-wide encryption can be provided across serving network boundaries (e.g. because inter-network TFO is available) then the signalling links requiring protection will cross network boundaries. In this situation it is important to note that the two serving networks may not be roaming partners yet they still must be able to confidentially protect inter-network signalling by establishing appropriate keys.

The key management scheme for network-wide encryption involves establishing an end-to-end session key between the end points of the traffic channel. It should not be possible to obtain this key by eavesdropping on any transmission links within the network. However, it may be possible to obtain the end-to-end key by compromising certain nodes within the network (e.g. nodes where link encryption terminates).

To satisfy lawful interception requirements it must be possible to decrypt end-to-end encrypted traffic within the core network to provide access to plaintext user traffic. Thus, the end-to-end encryption key (and decryption facilities) must be available in the core network for lawful interception reasons.

Issues for further study:

- Specification of key management scheme for the general case;
- The ability to terminate network-wide encryption key management at network gateways for inter-network user traffic channels.

7 ~~Network domain security mechanisms~~Void

~~This subclause describes mechanisms for establishing secure signalling links between network nodes, in particular between SN/VLRs and HE/AuCs. Such procedures may be incorporated into the roaming agreement establishment process.~~

~~7.1 Overview of Mechanism~~

~~The proposed mechanism consists of three layers.~~

~~7.1.1 Layer I~~

~~Layer I is a secret key transport mechanism based on an asymmetric crypto-system and is aimed at agreeing on a symmetric session key for each direction of communication between two networks X and Y.~~

~~NOTE 1: For secure transmission of sensitive data between elements of one and the same network operator only Layer II and Layer III will be involved. In this case Layer I can be dropped. There will also be only one symmetric key in this case, to be used for communication between network elements of one network operator in both directions.~~

~~The party wishing to send sensitive data initiates the mechanism and chooses the symmetric session key it wishes to use for sending the data to the other party. The other party shall choose a symmetric session key of its own, used for sending data in the other direction. This second key shall be transported immediately after the first key has been successfully transported. The session symmetric keys are protected by asymmetric techniques. They are exchanged between certain elements called the *Key Administration Centres* (KACs) of the network operators X and Y. The format of the Layer I transmissions is based on ISO/IEC 11770-3: *Key Management — Mechanisms using Asymmetric Techniques* [10]. Public Keys may be exchanged between a pair of network operators when setting up their roaming agreement (manual roaming) or they may be distributed by a TTP e.g. in case of automatic roaming.~~

~~NOTE 2: In the case of manual roaming no general PKI is required.~~

~~NOTE 3: For the transmission of the messages, no special assumptions regarding the transport protocol are made, a possible example would be IP.~~

~~7.1.2 Layer II~~

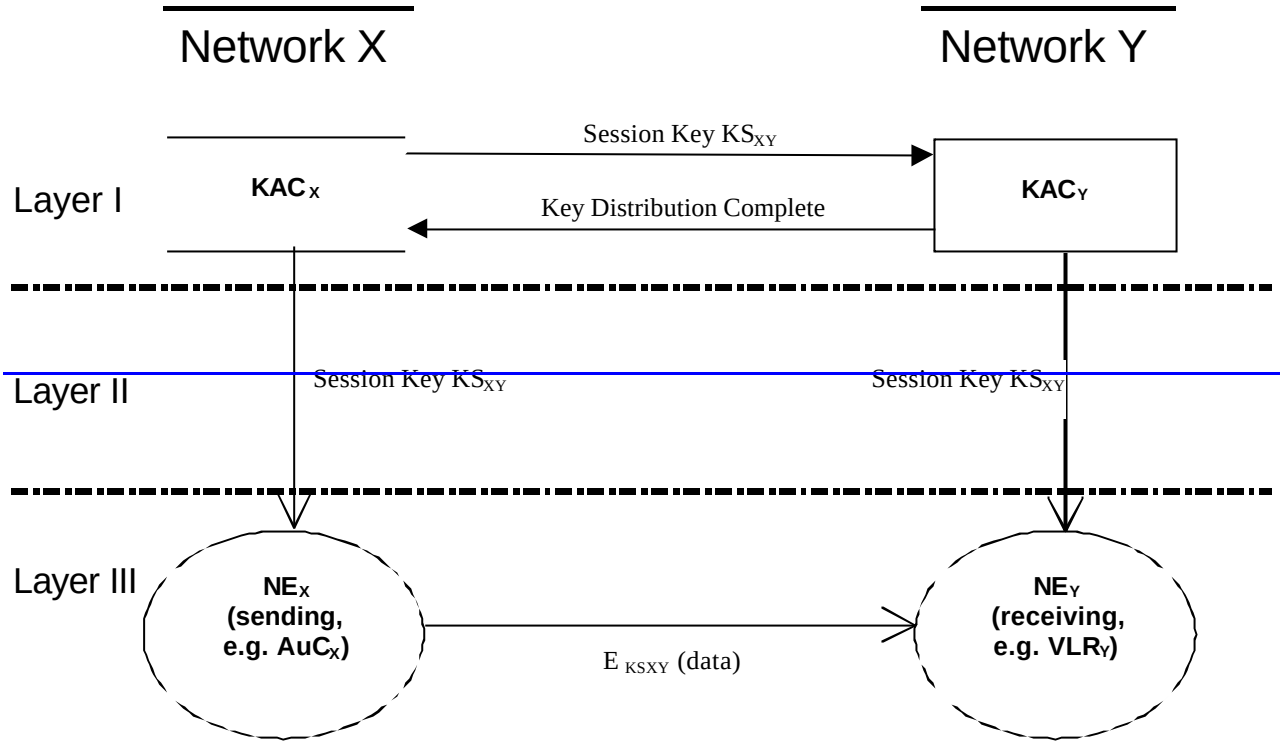
~~In Layer II the agreed symmetric keys for sending and receiving data are distributed by the KACs in each network to the relevant network elements. For example, an AuC will normally send sensitive authentication data to VLRs belonging to other networks and will therefore get a session key from its KAC. Layer II is carried out entirely inside one operator's network. It is clear that the distribution of the symmetric keys to the network elements must be carried out in a secure way, as not to compromise the whole system. Therefore, in Annex E a mechanism for distributing the keys, which very similar to that of Layer I, is proposed for Layer II.~~

~~7.1.3 Layer III~~

~~Layer III uses the distributed symmetric keys for securely exchanging sensitive data between the network elements of one operator (internal use) or different operators (external use) by means of a symmetric encryption algorithm. A block cipher (e.g. BEANO, which has been developed by ETSI SAGE [11]) shall be used for this purpose, as defined in 3G TS 33.105. The encrypted (resp. authenticity/integrity protected) messages will be transported via the MAP protocol.~~

~~7.1.4 General Overview~~

~~Figure 16 provides an overview of the whole mechanism. Note that the messages are not fully specified in this figure. Rather, only the "essential" parts of the messages are given. More details on the format of the messages in the single layers will be provided in subsequent chapters.~~



~~Figure 20: Overview of Proposed Mechanism~~

~~E_{KSXY}(data) denotes encryption of data by a symmetric algorithm using the session key from network X to network Y. (If the data are sent inside one operator's network, X = Y).~~

~~7.2 Layer I Message Format~~

~~Layer I describes the communication between two newly defined network entities of different networks, the so-called Key Administration Centres (KACs).~~

~~NOTE: We do not make any assumptions about the protocols to be used for this communications, although IP might be the most likely candidate.~~

~~7.2.1 Properties and Tasks of Key Administration Centres~~

~~There is only one KAC per network operator. KACs perform the following tasks:~~

- ~~Generation and storage of its own asymmetric key pairs (different key pairs used for signing/verifying and encrypting/decrypting, cf. 7.2.2)~~
- ~~Storage of public keys of KACs of other network operators~~
- ~~Generation and storage of symmetric session keys for sending sensitive information to network entities of other networks~~
- ~~Reception and storage of symmetric session keys for receiving sensitive information from network entities of other networks~~
- ~~Secure distribution of symmetric session keys to network entities in the same network~~

~~Due to these sensitive tasks, a KAC has to be physically secured.~~

~~7.2.2 Transport of Session Keys~~

~~The transport of session keys in Layer I is based on asymmetric cryptographic techniques (cf. [10]).~~

[Note: Public key certificates shall be included in Text3 if required.]

In order to establish a symmetric session key with version no. i to be used for sending data from X to Y, the KAC_X sends a message containing the following data to the KAC_Y:

$E_{PK(Y)}\{X||Y||i||KS_{XY}(i)||RND_X||Text1||D_{SK(X)}(Hash(X||Y||i||KS_{XY}(i)||RND_X||Text1))||Text2\}||Text3$

The reasons for this message format are as follows:

- Encrypting the message with the public key used for encrypting of the receiving network Y provides message confidentiality, while decrypting the message body with the private key used for signing of the sending network X provides message integrity and authenticity.

- X includes RND_X to make sure that the message contents contains some random data before signing.

NOTE: The hash function used shall be collision-resistant and have the one-way property.

The symmetric session keys $KS_{XY}(i)$ should be periodically updated by this process, thereby moving on to $KS_{XY}(i+1)$. For each new session key KS_{XY} i is incremented by one.

After having successfully decrypted the key transport message and having verified the digital signature of the sending network, including the hash value, and having checked the received i the receiving network starts Layer II activities.

If anything goes wrong, e.g. computing the hash value of $X||Y||i||KS_{XY}(i)||RND_X||Text1$ does not yield the expected result, a RESEND message should be sent by Y to X in the form

RESEND||Y||X

Y shall reject messages with i smaller or equal than the currently used i.

After having successfully distributed the symmetric session key received by network X to its own network entities, network Y sends to X a Key Distribution Complete Message. This is an indication to KAC_X to start with the distribution of the key to its own entities, which can then start to use the key immediately. The message takes the form

$KEY\_DIST\_COMPLETE||Y||X||i||RND_Y||D_{SK(Y)}(Hash(KEY\_DIST\_COMPLETE||Y||X||i||RND_Y))$

where i indicates the distributed key and RND_Y is a random number generated by Y. The digital signature is appended for integrity and authenticity purposes. Y includes RND_Y to make sure that the message contents determined by X will be modified before signing.

Since most of the signalling messages to be secured are bidirectional in character, immediately after successful completion the procedure described here shall be repeated, now with Y choosing a key $KS_{YX}(i)$ to be used in the reverse direction, and X being the receiving party. Thereby keys for both directions are established.

7.3 Layer II Message Format

It shall be stressed here once again that the distribution of the symmetric session keys, which has to be performed in Layer II, must be done securely. For a detailed proposal which is based on the asymmetric key transport mechanism of Layer I, see Annex E.

In order to ensure that no network element starts enciphering with a key that not all potentially corresponding network elements have received yet, the following approach is suggested:

The distribution of the session keys KS_{XY} in network X having initiated the Layer I message exchange should not begin before the Key Distribution Complete Message from the receiving network Y has been received by KAC_X in Layer I. As soon as a network element of X has received a session key KS_{XY}, it may start enciphering with this key.

A similar statement holds if the transported session keys are used internally only: In this case, all network elements of X should get the symmetric session keys KS_{XY} for internal use as decryption keys (marked with flag RECEIVED) first; if all network elements of X have acknowledged that they have recovered these keys, the KAC_X sends the same key KS_{XY} again as encryption keys (marked with flag SEND). Again, as soon as a network element of X has received an encryption key (marked with flag SEND), it may start enciphering with this key.

7.4 Layer III Message Format

7.4.1 General Structure of Layer III Messages

Layer III messages are transported via the MAP protocol, that means, they form the payload of a MAP message after the original MAP message header. For Layer III Messages, three levels of protection (or protection modes) are defined providing the following security features:

Protection Mode 0: No Protection

Protection Mode 1: Integrity, Authenticity

Protection Mode 2: Confidentiality, Integrity, Authenticity

NOTE: GTP based transmission data will also contain sensitive data. This data will require an equal level of security (e.g. authentication parameters, subscriber profile information, etc.). The specifications will extended to address GTP based transmissions using industry standard techniques (such as IPSEC) where appropriate. The possibility of extending these mechanisms to secure CAP/INAP signalling is also being investigated.

Layer III messages consists of a Security Header and the Layer III Message Body that is protected by the symmetric encryption algorithm, using the symmetric session keys that were distributed in layer II. Layer III Messages have the following structure:

Security Header	Layer III Message Body

In all three protection modes, the security header is transmitted in cleartext. It shall comprise the following information:

- protection mode;
- other security parameters (if required, e.g. IV, Version No. of Key Used, Encryption Algorithm Identifier, Mode of Operation of Encryption Algorithm, cf. section 7.4.3).

Both parts of the Layer III messages, security header and message body, will become part of the "new" MAP message body. Therefore, the complete "new" MAP messages take the following form in this proposal:

MAP Message Header	MAP Message Body

	Layer III Message

MAP Message Header	Security Header	Layer III Message Body

Like the security header, the MAP message header is transmitted in cleartext. In protection mode 2 providing confidentiality, the Layer III Message Body is essentially the encrypted "old" MAP message body. For integrity and authenticity, an encrypted hash calculated on the MAP message header, security header and the "old" MAP message body in cleartext is included in the Layer III Message Body in protection modes 1 and 2. In protection mode 0 no protection is offered, therefore the Layer III Message Body is identical to the "old" MAP message body in cleartext in this case.

Summing up, the Protected MAP Message (i.e. the Layer III Message) is a sequence of data elements consisting of the MAP Message Header, the Security Header and the Layer III Message Body. In the following subchapters, the contents

of the Layer III Message Body for the different protection modes and the security header will be specified in greater detail.

7.4.2 Format of Layer III Message Body

7.4.2.1 Protection Mode 0

Protection Mode 0 offers no protection at all. Therefore, the Layer III message body in protection mode 0 is identical to the original MAP message body in cleartext.

7.4.2.2 Protection Mode 1

The message body of Layer III messages in protection mode 1 takes the following form:

$$Cleartext \| TVP \| E_{KSXY(i)}(Hash(MAP\ Header \| Security\ Header \| Cleartext \| TVP))$$

where "Cleartext" is the message body of the original MAP message in cleartext. Therefore, in Protection Mode 1 the Layer III Message Body is a sequence of the following data elements and data types:

- Cleartext (OCTET STRING)
- Time Variant Parameter (UTCTime)
- Integrity Check (OCTET STRING)

Authentication of origin is achieved by encrypting the hash value of the cleartext, since only a network element knowing $KS_{XY}(i)$ can encrypt in this way. Message integrity and validation is achieved by hashing and encrypting the cleartext.

[Note: The case X=Y, i.e. only one key for sending and receiving, corresponds to internal use inside network X.]

Note that protection mode 1 is compatible to the present MAP protocol, since everything appended to the cleartext may be ignored by a receiver incapable of decrypting.

7.4.2.3 Protection Mode 2

The Layer III Message Body in protection mode 2 takes the following form:

$$E_{KSXY(i)}(Cleartext \| TVP \| Hash(MAP\ Header \| Security\ Header \| Cleartext \| TVP))$$

where "Cleartext" is the original MAP message in cleartext. Therefore, in protection mode 2 the Layer III message body is just an OCTET STRING which can only be interpreted after having decrypted it. After decryption, the data structure is similar to that in Protection Mode 1.

Message confidentiality is achieved by encrypting with the session key. This also provides for authentication of origin, since only a network element knowing KSXY(i) can encrypt in this way. Message integrity and validation is achieved by hashing the cleartext. TVP is a random number that avoids traceability.

[Note1: There is need for replay protection of Layer III messages; this is for further study.
By making use of a TVP as timestamp (perhaps derived from an overall present master time) this could be achieved.]

[Note2: In protection mode 2, the original MAP message body will be encrypted in order to achieve confidentiality. For integrity and authenticity, an encrypted hash calculated on the MAP message header and body in cleartext (i.e. the original MAP message) is appended to the messages in protection mode 1 and 2. All protection modes need a security header to be added.
When implementing these changes, care has to be taken that the maximum length of a MAP message (approx. 250 byte) is not exceeded by the protected MAP messages of Layer III, otherwise substantial changes to the underlying SS7 protocol levels (TCAP and SCCP) would have to be made.]

7.4.3 Structure of Security Header

The security header is a sequence of the following data elements and data types:

- Protection Mode (INTEGER)
- Key Identifier (INTEGER)
- Algorithm Identifier (AlgorithmIdentifier)
- Mode of Operation (INTEGER)
- Initialisation Vector (OCTET STRING OPTIONAL)

NOTE: Whether the Initialisation Vector is needed depends on the mode of operation of the encryption algorithm.

7.5 Mapping of MAP Messages and Modes of Protection

The network operator should be able to assign the mode of protection to each MAP message in order to adapt the level of protection according to its own security policy. Guidance may be obtained from the SS7 Signalling Protocols Threat Analysis [12].

7.6 Distribution of security parameters to UTRAN

Confidentiality and integrity between the user and the network is handled by the UE/USIM and the RNC.

The security parameters for the confidentiality and integrity algorithms must be distributed from the core network to the RNC over the Iu interface in a secure manner. The actual mechanism for securing these parameters has not yet been identified.

Annex E (informative): Void
A Proposal for Layer II Message Format

E.1 Introduction

In Layer II symmetric session keys (to encrypt/decrypt data before sending/after receiving) are distributed by the KACs in each network to the relevant network elements. For example, an AuC_X will normally send sensitive authentication data to VLR_X and will therefore get a session KS_{XX} key from its KAC_X. Layer II is carried out entirely inside one operator's network.

However, in order to achieve a more consistent overall scheme, in this annex it is suggested to use for Layer II the same mechanism for distributing the keys as in Layer I. This requires the KACs of the different networks to generate and distribute asymmetric key pairs for the network elements of that network. These key pairs will then be used to transfer the symmetric session keys in the same way as in Layer I.

The public and private key pairs needed for the network entities should be distributed to the entities in a secure way, which is in principle an operation & maintenance task. One way to do this is to distribute the key pairs, along with the necessary crypto-software, to the network entities in the form of chipcards, which can also carry out the necessary computations. Therefore, all that has to be added to the present network entities are chipcard readers with a standardised interface. Thus, on adoption of this proposal, in addition to their present tasks, the network entities would have to:

- Store the symmetric session keys to encrypt/decrypt data before sending/after receiving to/from network entities of other networks (external) and of their own network (internal);
- Encrypt/decrypt MAP messages according to their Mode of protection (cf. 7.4). The necessary computations may be carried out by a chipcard.

In addition to their tasks listed in 7.2.1 of the main document, the KACs would have to:

- Generate and store asymmetric key pairs for network entities in the same network;
- Distribute asymmetric key pairs to network entities in the same network.

E.2 Proposed Layer II Message Format

The Layer II messages themselves take the same form as in 7.2 of the main document, where the 'receiving network Y' has to be replaced by 'receiving network entity NE_X' (or X by NE_X). Further, the Key Distribution Complete message is not needed in Layer II. However, the distribution of the session keys KS_{XX} in network X having initiated the Layer I message exchange should not begin before the Key Distribution Complete Message from the receiving network Y has been received by the KAC_X in Layer I. As soon as a network element of X has received a session key KS_{XX}, it may start enciphering with this key. A similar statement holds if the transported keys are used internally only: In this case, all network elements of X should get the symmetric session key KS_{XX} to be used internal for encryption (marked as decryption key with flag RECEIVE) first; if all network elements have acknowledged that they have recovered these keys, the KAC_X sends the same key again (marked as encryption key with flag SEND). Again, as soon as a network element has received the session key KS_{XX} (with flag SEND), it may start enciphering with this key.

[Note: As for layer I, no assumptions about the transport protocol are made, although IP might be a good candidate.]

E.2.1 Sending a session key for decryption

In order to transport a symmetric session key (marked with flag RECEIVE) with version no. i to be used to decrypt received data from network elements of network X in NE_Y, the KAC of Y sends a message containing the following data to NE_Y:

$E_{PK(NEY)}$ {X||NE_X||RECEIVE||i||KS_{XX}(i)||RND_X||Text1||$D_{SK(X)}$(*Hash*(X||NE_X||RECEIVE||i||KS_{XX}(i)||RND_X||Text1))||Text2||Text3

After having successfully decrypted the key transport message and having verified the digital signature of the sending network including the hash value, the receiving network entity sends an key installed message to its Key Administration Centre KAC_Y. The message takes the form

KEY_INSTALLED||X||NE_Y||RND_X||i

This message can only be sent by the receiving network entity, because only this entity can know about RND_X. If anything goes wrong, e.g. computing the Hash of X||NE_X||RECEIVE||i||KS_{XX}(i)||RND_X||Text1 does not yield the expected result, a RESEND message should be sent by NE_Y to KAC_Y in the form

RESEND||NE_Y

E.2.2 Sending a session key for encryption

In order to transport a symmetric SEND key with version no. i to be used for sending data from NE_X to network elements of network Y, KAC_Y sends a message containing the following data to NE_X:

$E_{PK(NEX)}$ {NE_X||Y||SEND||i||KS_{XX}(i)||RND_X||Text1||$D_{SK(Y)}$(*Hash*(NE_X||Y||SEND||i||KS_{XX}(i)||RND_X||Text1))||Text2||Text3