

**23-26 May, 2000**

**Yokohama, Japan**

---

**3GPP TSG CN WG4  
Rotenburg, Germany  
22 – 26 May 2000**

**Tdoc N4-000366**

**From: TSG CN WG4<sup>1</sup>**  
**To: TSG SA WG3, TSG SA WG5**  
**Subject: LS on Key Exchange and Key Distribution**

Besides their work on introduction of the secure transport of MAP messages (Layer III) in 29.002, TSG CN WG4 had a look at the possibility of introducing the Key Exchange and Key Distribution mechanisms for MAP entities via MAP.

Basis for the consideration is attached proposed CR to 3G TS 29.002 on Introduction of Key Exchange and Key Distribution (Layer I and Layer II).

It was identified in TSG CN WG4 that with additional effort it would be possible to stabilise this Change Request until TSG CN #8.

Before this effort is spent, TSG CN WG4 urgently asks TSG SA WG3 (together with TSG SA WG5) for a general decision on the mechanism to be used for Key Exchange and Key Distribution.

Attached TDoc N4-000210, CR 29.002-131 on Key exchange and distribution for MAP security

---

<sup>1</sup> Contact:  
Steffen Habermann, T-Mobil  
Steffen.Habermann@t-mobil.de

## CHANGE REQUEST

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

**29.002 CR 131**

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: **CN#08**  
*list expected approval meeting # here ↑*

for approval   
 for information

strategic   
 non-strategic  *(for SMG use only)*

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

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

**Source:** Siemens **Date:** 10.05.00

**Subject:** Key exchange and distribution for MAP security

**Work item:** MAP Security

<b>Category:</b>	F Correction <input type="checkbox"/>	<b>Release:</b>	Phase 2 <input type="checkbox"/>
<i>(only one category shall be marked with an X)</i>	A Corresponds to a correction in an earlier release <input type="checkbox"/>		Release 96 <input type="checkbox"/>
	B Addition of feature <input checked="" type="checkbox"/>		Release 97 <input type="checkbox"/>
	C Functional modification of feature <input type="checkbox"/>		Release 98 <input type="checkbox"/>
	D Editorial modification <input type="checkbox"/>		Release 99 <input type="checkbox"/>
		Release 00 <input checked="" type="checkbox"/>	

**Reason for change:** Introduction of Layer I and Layer II messages for distribution of MAP security keys.

**Clauses affected:** 7.6.1.4, 13B (new), 16.2.2, 17.1.5, 17.1.6, 17.2.2, 17.3.2, 17.3.3, 17.5, 17.6.6, 17.6.9 (new), 17.7.7, 17.7.14 (new), 24B (new).

<b>Other specs affected:</b>	Other 3G core specifications <input type="checkbox"/>	→ List of CRs:
	Other GSM core specifications <input type="checkbox"/>	→ List of CRs:
	MS test specifications <input type="checkbox"/>	→ List of CRs:
	BSS test specifications <input type="checkbox"/>	→ List of CRs:
	O&M specifications <input type="checkbox"/>	→ List of CRs:

**Other comments:** This CR does not cover the Layer III message protection



help.doc

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

#### 1 7.6.1.4 User error

2 This parameter can take values as follows:

3 NOTE: The values are grouped in order to improve readability; the grouping has no other significance.

4 a) Generic error:

5 - system failure, i.e. a task cannot be performed because of a problem in another entity. The type of entity or  
6 network resource may be indicated by use of the network resource parameter;

7 - data missing, i.e. an optional parameter required by the context is missing;

8 - unexpected data value, i.e. the data type is formally correct but its value or presence is unexpected in the  
9 current context;

10 - resource limitation;

11 - initiating release, i.e. the receiving entity has started the release procedure;

12 - facility not supported, i.e. the requested facility is not supported by the PLMN;

13 - incompatible terminal, i.e. the requested facility is not supported by the terminal.

14 b) Identification or numbering problem:

15 - unknown subscriber, i.e. no such subscription exists;

16 - number changed, i.e. the subscription does not exist for that number any more;

17 - unknown MSC;

18 - unidentified subscriber, i.e. if the subscriber is not contained in the database and it has not or cannot be  
19 established whether or not a subscription exists;

20 - unallocated roaming number;

21 - unknown equipment;

22 - unknown location area.

23 c) Subscription problem:

24 - roaming not allowed, i.e. a location updating attempt is made in an area not covered by the subscription;

25 - illegal subscriber, i.e. illegality of the access has been established by use of authentication procedure;

26 - bearer service not provisioned;

27 - teleservice not provisioned;

28 - illegal equipment, i.e. the IMEI check procedure has shown that the IMEI is blacklisted or not whitelisted.

29 d) Handover problem:

30 - no handover number available, i.e. the VLR cannot allocate a number for handover or cannot allocate the  
31 required amount of numbers for relocation;

32 - subsequent handover failure, i.e. handover to a third MSC failed for some reason.

33 e) Operation and maintenance problem:

34 - tracing buffer full, i.e. tracing cannot be performed because the tracing capacity is exceeded.

- 35 f) Call set-up problem:
- 36 - no roaming number available, i.e. a roaming number cannot be allocated because all available numbers are in  
37 use;
- absent subscriber, i.e. the subscriber has activated the detach service or the system detects the absence  
condition. This error may be qualified to indicate whether the subscriber was IMSI detached, in a restricted  
area or did not respond to paging;
- 38 - busy subscriber. This error may be qualified to indicate that the subscriber was busy due to CCBS and that  
39 CCBS is possible;
- 40 - no subscriber reply;
- 41 - forwarding violation, i.e. the call has already been forwarded the maximum number of times that is allowed;
- 42 - CUG reject, i.e. the call does not pass a CUG check; additional information may also be given in order to  
43 indicate rejection due to e.g. incoming call barred or non-CUG membership.
- 44 - call barred. Optionally, additional information may be included for indicating either that the call meets a  
45 barring condition set by the subscriber or that the call is barred for operator reasons. In case of barring of  
46 Mobile Terminating Short Message, the additional information may indicate a barring condition due to  
47 «unauthorised Message Originator».
- 48 - optimal routeing not allowed, i.e. the entity which sends the error does not support optimal routeing, or the  
49 HLR will not accept an optimal routeing interrogation from the GMSC, or the call cannot be optimally routed  
50 because it would contravene optimal routeing constraints.
- 51 - forwarding failed, i.e. the GMSC interrogated the HLR for forwarding information but the HLR returned an  
52 error.
- 53 g) Supplementary services problem:
- 54 - call barred;
- 55 - illegal SS operation;
- 56 - SS error status;
- 57 - SS not available;
- 58 - SS subscription violation;
- 59 - SS incompatibility;
- 60 - negative password check;
- 61 - password registration failure;
- 62 - Number of Password Attempts;
- 63 - USSD Busy;
- 64 - Unknown Alphabet.
- 65 - short term denial;
- 66 - long term denial.
- 67 For definition of these errors see GSM 04.80.

- 68 h) Short message problem:
- 69 - SM delivery failure with detailed reason as follows:
- 70 - memory capacity exceeded;
- 71 - MS protocol error;
- 72 - MS not equipped;
- 73 - unknown service centre (SC);
- 74 - SC congestion;
- 75 - invalid SME address;
- 76 - subscriber is not an SC subscriber;
- 77 - and possibly detailed diagnostic information, coded as specified in TS GSM 03.40, under SMS-SUBMIT-  
78 REPORT and SMS-DELIVERY-REPORT. If the SM entity which returns the SM Delivery Failure error  
79 includes detailed diagnostic information, it shall be forwarded in the  
80 MAP\_MO\_FORWARD\_SHORT\_MESSAGE and in the MAP\_MT\_FORWARD\_SHORT\_MESSAGE  
81 response.
- 82 - message waiting list full, i.e. no further SC address can be added to the message waiting list;
- 83 - Subscriber busy for MT SMS, i.e. the mobile terminated short message transfer cannot be completed because:
- 84 - another mobile terminated short message transfer is going on and the delivery node does not support  
85 message buffering; or
- 86 - another mobile terminated short message transfer is going on and it is not possible to buffer the message  
87 for later delivery; or
- 88 - the message was buffered but it is not possible to deliver the message before the expiry of the buffering  
89 time defined in GSM 03.40;
- Absent Subscriber SM, i.e. the mobile terminated short message transfer cannot be completed because the network cannot contact the subscriber. Diagnostic information regarding the reason for the subscriber's absence may be included with this error.

- 90 i) Location services problem:
- 91 - Unauthorized Requesting Network
  - 92 - Unauthorized LCS Client with detailed reason as follows
  - 93 - Unauthorized Privacy Class
  - 94 - Unauthorized Call Unrelated External Client
  - 95 - Unauthorized Call Related External Client
  - 96 - Privacy override not applicable
  - 97 - Position method failure with detailed reason as follows:
    - 98 - Congestion
    - 99 - Insufficient resources
    - 100 - Insufficient Measurement Data
    - 101 - Inconsistent Measurement Data
    - 102 - Location procedure not completed
    - 103 - Location procedure not supported by target MS
    - 104 - QoS not attainable
  - 105 - Unknown or unreachable LCS Client
- 106 k) MAP message protection problem:
- 107 - Security Association Not Supported, i.e. the encryption algorithm or the mode of operation is not supported,  
108 or the MAP dialogue is not supported in protected mode.
  - 109 - Key Distribution Failure, i.e. the security association has not successfully been distributed to all relevant  
110 MAP entities.
  - 111 - DecryptionFailure, i.e. when decrypting the encrypted part of the message a failure occurred, e.g. integrity  
112 failed.
- 113
- 114
- 115 \*\*\*\*\* NEXT MODIFICATION \*\*\*\*\*

---

## 116 13B Key Distribution Services

### 117 13B.1 MAP-UPDATE-SECURITY-ASSOCIATION Service

#### 118 13B.1.1 Definition

119 This service is used between KAC-A and KAC-B to initiate updating of a security association in PLMN B. The MAP-  
120 UPDATE-SECURITY-ASSOCIATION Service is a confirmed service using the primitives from table 13B.1/1.

#### 121 13B.1.2 Service Primitives

122 **Table 13B.1/1: MAP-UPDATE-SECURITY-ASSOCIATION**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
PLMN-Id	M	M(=)		
Layer I encrypted security association	M	M(=)		
User error			C	C(=)
Provider error				O

123

#### 124 13B.1.3 Parameter Use

##### 125 Invoke Id

126 See definition in subclause 7.6.1.

##### 127 PLMN-Id

128 MCC and MNC of PLMN-A

##### 129 Layer I encrypted security association

130 This parameter represents in encrypted mode the security association to be used for MAP message protection when  
131 MAP messages are to be sent from PLMN A to PLMN B.

132 A Security Association encompasses the following parameters:

##### 133 Encryption Algorithm Identifier

134 identifies the encryption Algorithm used for MAP message protection. May be absent if it is not changed.

##### 135 Key Version Number

136 Version number of the encryption key to be used for MAP message protection. Shall be present if and only if the  
137 parameter 'key' is present.

##### 138 Key

139 Encryption Key to be used for MAP message protection. May be absent if it is not changed.

##### 140 Protection Modes and Modes of Operation

141 Indicates whether a MAP dialogue needs protection, and if so, indicates for every component of the dialogue the  
142 protection mode and mode of operation of the encryption algorithm to be used

##### 143 Remove All Indicator

144 indicates that PLMN A is going to cease protecting of MAP messages.

145 User error

146 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

147 - SecurityAssociationNotSupported;

148 - KeyDistributionFailure;

149 - DecryptionFailure;

150 - System failure;

151

152 Provider error

153 For definition of provider errors see subclause 7.6.1.

## 154 13B.2 MAP-NOTE-NEW-SECURITY-ASSOCIATION Service

### 155 13B.2.1 Definition

156 This service is used between the KAC and any MAP entity to distribute a new security association within a PLMN. The  
157 MAP-NOTE-NEW-SECURITY-ASSOCIATION Service is a confirmed service using the primitives from  
158 table 13B.2/1.

### 159 13B.2.2 Service Primitives

160

**Table 13B.2/1: MAP-UPDATE-SECURITY-ASSOCIATION**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Direction	M	M(=)		
PLMN-Id	M	M(=)		
Layer II encrypted security association	M	M(=)		
User error			C	C(=)
Provider error				O

161

### 162 13B.2.3 Parameter Use

163 Invoke Id

164 See definition in subclause 7.6.1.

165 Direction

166 This parameter indicates whether the new security association is to be used for sending or for receiving of MAP  
167 messages.

168 PLMN-Id

169 MCC and MNC of the peer PLMN.

170 Layer II encrypted security association

171 This parameter represents in encrypted mode the security association to be used for MAP message protection when  
172 MAP messages are to be sent or to be received from the peer PLMN.

173 A Security Association encompasses the following parameters:

174 Encryption Algorithm Identifier

175 identifies the encryption Algorithm used for MAP message protection. May be absent if it is not changed.



176 Key Version Number

177 Version number of the encryption key to be used for MAP message protection. Shall be present if and only if the  
 178 parameter 'key' is present.

179 Key

180 Encryption Key to be used for MAP message protection. May be absent if it is not changed.

181 Protection Modes and Modes of Operation

182 Indicates whether a MAP dialogue needs protection, and if so, indicates for every component of the dialogue the  
 183 protection mode and mode of operation of the encryption algorithm to be used

184 Remove All Indicator

185 indicates that protection of MAP messages for the indicated direction is going to be ceased.

186 User error

187 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

188 - SecurityAssociationNotSupported;

189 - DecryptionFailure;

190 - System failure;

191

192 Provider error

193 For definition of provider errors see subclause 7.6.1.

## 194 13B.3 MAP-SECURITY-UPDATE-COMPLETE Service

### 195 13B.3.1 Definition

196 This service is used between the KAC-A and KAC-B as well as between KAC and any MAP entity to indicate that key  
 197 distribution is complete. The MAP-SECURITY-UPDATE-COMPLETE Service is a non-confirmed service using the  
 198 primitives from table 13B.3/1.

### 199 13B.3.2 Service Primitives

200 **Table 13B.3/1: MAP-SECURITY-UPDATE-COMPLETE**

Parameter name	Request	Indication
Invoke Id	M	M(=)
Success Indicator	C	C(=)

201

202

203 **13B.3.3 Parameter Use**

204 Invoke Id

205 See definition in subclause 7.6.1.

206 Success Indicator

207 This parameter indicates by its presence that the new security association has been successfully distributed and that the  
208 old security association shall no longer be used.

209 This parameter indicates by its absence that the new security association has not been successfully distributed and that  
210 the old security association (if any) remains valid.

211

212 \*\*\*\*\* NEXT MODIFICATION \*\*\*\*\*

213 **16.2.2 Use of other parameters of component handling primitives**

214 **16.2.2.1 Dialogue Id**

215 The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

216 **16.2.2.2 Class**

217 The value of this parameter is set by the MAP PM according to the type of the operation to be invoked.

218 **16.2.2.3 Linked Id**

219 When a service response is mapped to a class 4 operation, the value of this parameter is set by the MAP PM and  
220 corresponds to the value assigned by the user to the initial service request (i.e. the value of the invoke ID parameter of  
221 the request primitive). Otherwise if such a parameter is included in MAP request/indication primitives it is directly  
222 mapped to the linked ID parameter of the associated TC-INVOKE request/indication primitives.

223 **16.2.2.4 Operation**

224 When mapping a request primitive on to a Remote Operations PDU (invoke), the MAP PM shall set the operation code  
225 according to the mapping described in table 16.2/1.

226 When mapping a response primitive on to a Remote Operations service, the MAP PM shall set the operation code of the  
227 TC-RESULT-L/NL primitive (if required) to the same value as the one received at invocation time.

Table 16.2/1: Mapping of MAP specific services on to MAP operations

MAP-SERVICE	operation
MAP-ACTIVATE-SS	activateSS
MAP-ACTIVATE-TRACE-MODE	activateTraceMode
MAP-ALERT-SERVICE-CENTRE	alertServiceCentre
MAP-ANY-TIME-INTERROGATION	anyTimeInterrogaton
MAP_AUTHENTICATION_FAILURE_REPORT	authenticationFailureReport
MAP-ANY-TIME-MODIFICATION	anyTimeModification
MAP-ANY-TIME-SUBSCRIPTION-INTERROGATION	anyTimeSubscriptionInterrogaton
MAP-CANCEL-LOCATION	cancelLocation
MAP-CHECK-IMEI	checkIMEI
MAP-DEACTIVATE-SS	deactivateSS
MAP-DEACTIVATE-TRACE-MODE	deactivateTraceMode
MAP-DELETE-SUBSCRIBER-DATA	deleteSubscriberData
MAP-ERASE-CC-ENTRY	eraseCC-Entry
MAP-ERASE-SS	eraseSS
MAP-FAILURE-REPORT	failureReport
MAP-FORWARD-ACCESS-SIGNALLING	forwardAccessSignalling
MAP-FORWARD-CHECK-SS-INDICATION	forwardCheckSsIndication
MAP-FORWARD-GROUP-CALL-SIGNALLING	forwardGroupCallSignalling
MAP-MT-FORWARD-SHORT-MESSAGE	mt-forwardSM
MAP-MO-FORWARD-SHORT-MESSAGE	mo-forwardSM
MAP-GET-PASSWORD	getPassword
MAP-INFORM-SERVICE-CENTRE	informServiceCentre
MAP-INSERT-SUBSCRIBER-DATA	insertSubscriberData
MAP-INTERROGATE-SS	interrogateSs
MAP-IST-ALERT	istAlert
MAP-IST-COMMAND	istCommand
MAP-NOTE-MS-PRESENT-FOR-GPRS	noteMsPresentForGprs
MAP-NOTE-SUBSCRIBER-DATA-MODIFIED	noteSubscriberDataModified
MAP-PREPARE-GROUP-CALL	prepareGroupCall
MAP-PREPARE-HANDOVER	prepareHandover
MAP-PREPARE-SUBSEQUENT-HANDOVER	prepareSubsequentHandover
MAP-PROCESS-ACCESS-SIGNALLING	processAccessSignalling
MAP-PROCESS-GROUP-CALL-SIGNALLING	processGroupCallSignalling
MAP-PROCESS-UNSTRUCTURED-SS-REQUEST	processUnstructuredSS-Request
MAP-PROVIDE-ROAMING-NUMBER	provideRoamingNumber
MAP-PROVIDE-SIWFS-NUMBER	provideSIWFSNumber
MAP-PROVIDE-SUBSCRIBER-LOCATION	provideSubscriberLocation
MAP-PROVIDE-SUBSCRIBER-INFO	provideSubscriberInfo
MAP-PURGE-MS	purgeMS
MAP-READY-FOR-SM	readyForSM
MAP-REGISTER-CC-ENTRY	registerCC-Entry
MAP-REGISTER-PASSWORD	registerPassword
MAP-REGISTER-SS	registerSS
MAP-REMOTE-USER-FREE	remoteUserFree
MAP-REPORT-SM-DELIVERY-STATUS	reportSmDeliveryStatus
MAP-RESET	reset
MAP-RESTORE-DATA	restoreData
MAP-SEND_GROUP-CALL_END_SIGNAL	sendGroupCallEndSignal
MAP-SEND-END-SIGNAL	sendEndSignal
MAP-SEND-AUTHENTICATION-INFO	sendAuthenticationInfo
MAP-SEND-IMSI	sendIMSI
MAP-SEND-IDENTIFICATION	sendIdentification
MAP-SEND-ROUTING-INFO-FOR-SM	sendRoutingInfoForSM
MAP-SEND-ROUTING-INFO-FOR-GPRS	sendRoutingInfoForGprs
MAP-SEND-ROUTING-INFO-FOR-LCS	sendRoutingInfoForLCS
MAP-SEND-ROUTING-INFORMATION	sendRoutingInfo
MAP-SET-REPORTING-STATE	setReportingState
MAP-SIWFS-SIGNALLING-MODIFY	SIWFSsignallingModify
MAP-STATUS-REPORT	statusReport
MAP-SUBSCRIBER-LOCATION-REPORT	subscriberLocationReport
MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION	ss-Invocation-Notification

MAP-UNSTRUCTURED-SS-NOTIFY	unstructuredSS-Notify
MAP-UNSTRUCTURED-SS-REQUEST	unstructuredSS-Request
MAP-UPDATE-GPRS-LOCATION	updateGprsLocation
MAP-UPDATE-LOCATION	updateLocation
MAP-NOTE-MM-EVENT	NoteMM-Event
MAP-UPDATE-SECURITY-ASSOCIATION	updateSecurityAssociation
MAP-SECURITY-UPDATE-COMPLETE	securityUpdateComplete
MAP-NOTE-NEW-SECURITY-ASSOCIATION	noteNewSecurityAssociation

229

230 \*\*\*\*\* NEXT MODIFICATION \*\*\*\*\*

231

## 232 17.1.5 Structure of the Abstract Syntax of MAP

233 For each MAP parameter which has to be transferred by a MAP Protocol Data Unit (MAP message), there is a PDU  
234 field (an ASN.1 NamedType) whose ASN.1 identifier has the same name as the corresponding parameter, except for the  
235 differences required by the ASN.1 notation (blanks between words are removed or replaced by hyphen, the first letter of  
236 the first word is lower-case and the first letter of the following words are capitalized, e.g. "no reply condition time" is  
237 mapped to "noReplyConditionTime"). Additionally some words may be abbreviated as follows:

- 238 bs basic service
- 239 ch call handling
- 240 cug closed user group
- 241 ho handover
- 242 ic incoming call
- 243 id identity
- 244 info information
- 245 mm mobility management
- 246 lcs location services
- 247 ms mobile service
- 248 oc outgoing call
- 249 om operation & maintenance
- 250 pw Password
- 251 sm short message service
- 252 ss supplementary service
- 253 st secure transport

254 The MAP protocol is composed of several ASN.1 modules dealing with either operations, errors, data types, and, if  
255 applicable, split into those dealing with mobile services, call handling services, supplementary services and short  
256 message services. For operations and errors no values are assigned, but only the operation and error types in order to  
257 allow use of the defined types also by other protocols (e.g. TS GSM 04.80). The values (operation codes and error  
258 codes) are defined in a separate module. The ASN.1 source lines are preceded by line-numbers at the left margin in  
259 order to enable the usage of the cross-reference in annex A.

260 The module containing the definition of the operation packages for MAP is:

- 261 1. MAP-OperationPackages.

262 The module containing the definition of the application contexts for MAP is:

- 263 2. MAP-ApplicationContexts.

264 The module containing the data types for the Abstract Syntax to be used for TCAPMessages.DialoguePortion for MAP  
265 is:

266 3. MAP-DialogueInformation.

267 The module containing the operation codes and error codes for MAP is:

268 4. MAP-Protocol.

269 The modules containing all operation type definitions for MAP are:

270 5. MAP-MobileServiceOperations;

271 6. MAP-OperationAndMaintenanceOperations;

272 7. MAP-CallHandlingOperations;

273 8. MAP-SupplementaryServiceOperations;

274 9. MAP-ShortMessageServiceOperations;

275 10. MAP-Group-Call-Operations.

276 11. MAP-LocationServiceOperations

277 12. MAP-SecureTransportOperations

278 The module containing all error type definitions for MAP is:

279 13. MAP-Errors.

280 Modules containing all data type definitions for MAP are:

281 14. MAP-MS-DataTypes;

282 15. MAP-OM-DataTypes;

283 16. MAP-CH-DataTypes;

284 17. MAP-SS-DataTypes;

285 18. MAP-SS-Code;

286 19. MAP-SM-DataTypes;

287 20. MAP-ER-DataTypes;

288 21. MAP-CommonDataTypes;

289 22. MAP-TS-Code;

290 23. MAP-BS-Code;

291 24. MAP-ExtensionDataTypes;

292 25. MAP-GR-DataTypes;

293 26. MAP-LCS-DataTypes;

294 27. MAP-ST-DataTypes.

295 References are made also to modules defined outside of the present document. They are defined in the technical  
 296 specification Mobile Services Domain and technical specification Transaction Capability respectively:

297 MobileDomainDefinitions;

298 TCAPMessages;

299 DialoguePDUs.

### 300 17.1.6 Application Contexts

301 The following informative table lists the latest versions of the Application Contexts used in this specification, with the  
 302 operations used by them and, where applicable, whether or not the operation description is exactly the same as for  
 303 previous versions. Information in 17.6 & 17.7 relates only to the ACs in this table.

AC Name	AC Version	Operations Used	Comments*
locationCancellationContext	v3	cancelLocation	
equipmentMngtContext	v2	checkIMEI	
imsiRetrievalContext	v2	sendIMSI	
infoRetrievalContext	v3	sendAuthenticationInfo	
interVlrInfoRetrievalContext	v3	sendIdentification	
handoverControlContext	v3	prepareHandover forwardAccessSignalling sendEndSignal processAccessSignalling prepareSubsequentHandover	the syntax of this operation has been extended in comparison with release 98 version
mwdMngtContext	v3	readyForSM	
msPurgingContext	v3	purgeMS	
shortMsgAlertContext	v2	alertServiceCentre	
resetContext	v2	reset	
networkUnstructuredSsContext	v2	processUnstructuredSS-Request unstructuredSS-Request unstructuredSS-Notify	
tracingContext	v3	activateTraceMode deactivateTraceMode	
networkFunctionalSsContext	v2	registerSS eraseSS activateSS deactivateSS registerPassword interrogateSS getPassword	
shortMsgMO-RelayContext	v3	mo-forwardSM	
shortMsgMT-RelayContext	v3	mt-forwardSM	
shortMsgGatewayContext	v3	sendRoutingInfoForSM reportSM-DeliveryStatus InformServiceCentre	the syntax of this operation has been extended in comparison with release 96 version
networkLocUpContext	v3	updateLocation forwardCheckSs-Indication restoreData insertSubscriberData activateTraceMode	the syntax is the same in v1 & v2
gprsLocationUpdateContext	v3	updateGprsLocation insertSubscriberData activateTraceMode	
subscriberDataMngtContext	v3	insertSubscriberData deleteSubscriberData	
roamingNumberEnquiryContext	v3	provideRoamingNumber	
locationInfoRetrievalContext	v3	sendRoutingInfo	

gprsNotifyContext	v3	noteMsPresentForGprs	
gprsLocationInfoRetrievalContext	v3	sendRoutingInfoForGprs	
failureReportContext	v3	failureReport	
callControlTransferContext	v4	resumeCallHandling	
subscriberInfoEnquiryContext	v3	provideSubscriberInfo	
anyTimeEnquiryContext	v3	anyTimeInterrogation	
anyTimeInfoHandlingContext	v3	anyTimeSubscriptionInterrogation anyTimeModification	
ss-InvocationNotificationContext	v3	ss-InvocationNotification	
sIWFSAllocationContext	v3	provideSIWFSNumber sIWFSsignallingModify	
groupCallControlContext	v3	prepareGroupCall processGroupCallSignalling forwardGroupCallSignalling sendGroupCallEndSignal	
reportingContext	v3	setReportingState statusReport remoteUserFree	
callCompletionContext	v3	registerCC-Entry eraseCC-Entry	
istAlertingContext	v3	istAlert	
ImmediateTerminationContext	v3	istCommand	
locationSvcEnquiryContext	v3	provideSubscriberLocation subscriberLocationReport	
locationSvcGatewayContext	v3	sendRoutingInfoForLCS	
mm-EventReportingContext	v3	noteMM-Event	
subscriberDataModificationNotificationContext	v3	noteSubscriberDataModified	
authenticationFailureReportContext	v3	authenticationFailureReport	
layer1SecurityContext	v3	updateSecurityAssociation securityUpdateComplete	
layer2SecurityContext	v3	noteNewSecurityAssociation securityUpdateComplete	

304

305 \*\*\*\*\*NEXT MODIFICATION \*\*\*\*\*

## 306 17.2.2 Packages specifications

307 .....

### 308 17.2.2.56 Layer 1 Key Distribution

309 This operation package includes the operations required for layer 1 key distribution between two KACs.

310

```

311 Layer1SecurityPackage-v3 ::= OPERATION-PACKAGE
312   -- Supplier is KAC if Consumer is KAC
313   CONSUMER INVOKES {
314     UpdateSecurityAssociation,
315     SecurityUpdateComplete
316   }

```

316

317

318 This package is v3 only.

319 17.2.2.57 Layer 2 Key Distribution

320 This operation package includes the operations required for layer 2 key distribution between KAC and any MAP-entity.

321

```

322 Layer2SecurityPackage-v3 ::= OPERATION-PACKAGE
    -- Supplier is any MAP entity if Consumer is KAC
323 CONSUMER INVOKES {
324     NoteNewSecurityAssociation,
325     SecurityUpdateComplete
326 }

```

327

328

329 This package is v3 only.

\*\*\*\*\*NEXT MODIFICATION\*\*\*\*\*

330 17.3.2 Application context definitions

331 .....

332 17.3.2.51 Layer 1 Key Distribution

333 This application context is used between two KACs for distribution of symmetric keys.

```

334 layer1SecurityContext-v3 APPLICATION-CONTEXT
335     -- Responder is KAC if Initiator is KAC
336     INITIATOR CONSUMER OF {
337         Layer1SecurityPackage-v3 }
338 ::= {map-ac failureReport(41) version3(3)}
339

```

340 This application-context is v3 only.

341 17.3.2.52 Layer 2 Key Distribution

342 This application context is used between two KACs for distribution of symmetric keys.

```

343 layer2SecurityContext-v3 APPLICATION-CONTEXT
344     -- Responder is any MAP entity if Initiator is KAC
345     INITIATOR CONSUMER OF {
346         Layer2SecurityPackage-v3 }
347 ::= {map-ac failureReport(44) version3(3)}
348

```

349 This application-context is v3 only.

17.3.3 ASN.1 Module for application-context-names

350 . . . . .

```

352 layer1SecurityContext-v3 OBJECT IDENTIFIER ::=
353     {map-ac layer1SecurityContext(41) version3(3)}
354

```

```

355 layer2SecurityContext-v3 OBJECT IDENTIFIER ::=
356     {map-ac layer2SecurityContext(44) version3(3)}
357

```

358 . . . . .

359

360 \*\*\*\*\*NEXT MODIFICATION\*\*\*\*\*



## 17.5 MAP operation and error codes

```
1 MAP-Protocol {
2     ccitt identified-organization (4) etsi (0) mobileDomain (0)
3     gsm-Network (1) modules (3) map-Protocol (4) version6 (6)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 IMPORTS
12     UpdateLocation,
13     CancelLocation,
14     PurgeMS,
15     SendIdentification,
16     UpdateGprsLocation,
17     PrepareHandover,
18     SendEndSignal,
19     ProcessAccessSignalling,
20     ForwardAccessSignalling,
21     PrepareSubsequentHandover,
22     SendAuthenticationInfo,
23     AuthenticationFailureReport,
24     CheckIMEI,
25     InsertSubscriberData,
26     DeleteSubscriberData,
27     Reset,
28     ForwardCheckSS-Indication,
29     RestoreData,
30     ProvideSubscriberInfo,
31     AnyTimeInterrogation,
32     AnyTimeSubscriptionInterrogation,
33     AnyTimeModification,
34     SendRoutingInfoForGprs,
35     FailureReport,
36     NoteMsPresentForGprs,
37     NoteMM-Event,
38     NoteSubscriberDataModified
39
40
41 FROM MAP-MobileServiceOperations {
42     ccitt identified-organization (4) etsi (0) mobileDomain (0)
43     gsm-Network (1) modules (3) map-MobileServiceOperations (5)
44     version6 (6)}
45
46     ActivateTraceMode,
47     DeactivateTraceMode,
48     SendIMSI
49 FROM MAP-OperationAndMaintenanceOperations {
50     ccitt identified-organization (4) etsi (0) mobileDomain (0)
51     gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
52     version6 (6)}
53
54     SendRoutingInfo,
55     ProvideRoamingNumber,
56     ResumeCallHandling,
57     ProvideSIWFSSNumber,
58     SIWFSSignallingModify,
59     SetReportingState,
60     StatusReport,
61     RemoteUserFree,
62     IST-Alert,
63     IST-Command
64 FROM MAP-CallHandlingOperations {
65     ccitt identified-organization (4) etsi (0) mobileDomain (0)
66     gsm-Network (1) modules (3) map-CallHandlingOperations (7)
67     version6 (6)}
68
69     RegisterSS,
70     EraseSS,
71     ActivateSS,
72     DeactivateSS,
73     InterrogateSS,
74     ProcessUnstructuredSS-Request,
75     UnstructuredSS-Request,
```

```

76   UnstructuredSS-Notify,
77   RegisterPassword,
78   GetPassword,
79   SS-InvocationNotification,
80   RegisterCC-Entry,
81   EraseCC-Entry
82 FROM MAP-SupplementaryServiceOperations {
83   ccitt identified-organization (4) etsi (0) mobileDomain (0)
84   gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
85   version6 (6)}
86
87   SendRoutingInfoForSM,
88   MO-ForwardSM,
89   MT-ForwardSM,
90   ReportSM-DeliveryStatus,
91   AlertServiceCentre,
92   InformServiceCentre,
93   ReadyForSM
94 FROM MAP-ShortMessageServiceOperations {
95   ccitt identified-organization (4) etsi (0) mobileDomain (0)
96   gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
97   version6 (6)}
98
99   PrepareGroupCall,
100  ProcessGroupCallSignalling,
101  ForwardGroupCallSignalling,
102  SendGroupCallEndSignal
103 FROM MAP-Group-Call-Operations {
104   ccitt identified-organization (4) etsi (0) mobileDomain (0)
105   gsm-Network (1) modules (3) map-Group-Call-Operations (22)
106   version6 (6)}
107
108   ProvideSubscriberLocation,
109   SendRoutingInfoForLCS,
110   SubscriberLocationReport
111 FROM MAP-LocationServiceOperations {
112   ccitt identified-organization (4) etsi (0) mobileDomain (0)
113   gsm-Network (1) modules (3) map-LocationServiceOperations (24)
114   version6 (6)}
115
116   UpdateSecurityAssociation,
117   NoteNewSecurityAssociation,
118   SecurityUpdateComplete
119 FROM MAP-SecureTransportOperations {
120   ccitt identified-organization (4) etsi (0) mobileDomain (0)
121   gsm-Network (1) modules (3) map-SecureTransportOperations (26)
122   version6 (6)}
123
124
125   SystemFailure,
126   DataMissing,
127   UnexpectedDataValue,
128   FacilityNotSupported,
129   UnknownSubscriber,
130   NumberChanged,
131   UnknownMSC,
132   UnidentifiedSubscriber,
133   UnknownEquipment,
134   RoamingNotAllowed,
135   IllegalSubscriber,
136   IllegalEquipment,
137   BearerServiceNotProvisioned,
138   TeleserviceNotProvisioned,
139   NoHandoverNumberAvailable,
140   SubsequentHandoverFailure,
141   TracingBufferFull,
142   OR-NotAllowed,
143   NoRoamingNumberAvailable,
144   AbsentSubscriber,
145   BusySubscriber,
146   NoSubscriberReply,
147   CallBarred,
148   ForwardingViolation,
149   ForwardingFailed,
150   CUG-Reject,
151   ATI-NotAllowed,
152   IllegalSS-Operation,
153   SS-ErrorStatus,
154   SS-NotAvailable,

```

```

155     SS-SubscriptionViolation,
156     SS-Incompatibility,
157     UnknownAlphabet,
158     USSD-Busy,
159     PW-RegistrationFailure,
160     NegativePW-Check,
161     NumberOfPW-AttemptsViolation,
162     SubscriberBusyForMT-SMS,
163     SM-DeliveryFailure,
164     MessageWaitingListFull,
165     AbsentSubscriberSM,
166     ResourceLimitation,
167     NoGroupCallNumberAvailable,
168     ShortTermDenial,
169     LongTermDenial,
170     IncompatibleTerminal,
171     UnauthorizedRequestingNetwork,
172     UnauthorizedLCSCClient,
173     PositionMethodFailure,
174     UnknownOrUnreachableLCSCClient,
175     ATSI-NotAllowed,
176     ATM-NotAllowed,
177     InformationNotAvailable,
178     MM-EventNotSupported,
179     SecurityAssociationNotSupported,
180     KeyDistributionFailure,
181     DecryptionFailure
182
183 FROM MAP-Errors {
184     ccitt identified-organization (4) etsi (0) mobileDomain (0)
185     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
186 ;
187
188
189 -- location registration operation codes
190
191 updateLocation UpdateLocation ::= localValue 2
192 cancelLocation CancelLocation ::= localValue 3
193 purgeMS PurgeMS ::= localValue 67
194 sendIdentification SendIdentification ::= localValue 55
195
196
197 -- handover operation codes
198
199 prepareHandover PrepareHandover ::= localValue 68
200 sendEndSignal SendEndSignal ::= localValue 29
201 processAccessSignalling ProcessAccessSignalling ::= localValue 33
202 forwardAccessSignalling ForwardAccessSignalling ::= localValue 34
203 prepareSubsequentHandover PrepareSubsequentHandover ::=
204     localValue 69
205
206
207 -- authentication operation codes
208
209 sendAuthenticationInfo SendAuthenticationInfo ::= localValue 56
210 authenticationFailureReport AuthenticationFailureReport ::= localValue 15
211
212
213 -- IMEI MANAGEMENT operation codes
214
215 checkIMEI CheckIMEI ::= localValue 43
216
217
218 -- subscriber management operation codes
219
220 insertSubscriberData InsertSubscriberData ::= localValue 7
221 deleteSubscriberData DeleteSubscriberData ::= localValue 8
222
223
224 -- fault recovery operation codes
225
226 reset Reset ::= localValue 37
227 forwardCheckSS-Indication ForwardCheckSS-Indication ::=
228     localValue 38
229 restoreData RestoreData ::= localValue 57
230
231

```

```

232 -- operation and maintenance operation codes
233
234 activateTraceMode ActivateTraceMode ::= localValue 50
235 deactivateTraceMode DeactivateTraceMode ::= localValue 51
236 sendIMSI SendIMSI ::= localValue 58
237
238
239 -- call handling operation codes
240
241 sendRoutingInfo SendRoutingInfo ::= localValue 22
242 provideRoamingNumber ProvideRoamingNumber ::= localValue 4
243 resumeCallHandling ResumeCallHandling ::= localValue 6
244 provideSIWFSNumber ProvideSIWFSNumber ::= localValue 31
245 siWFSSignallingModify SIWFSSignallingModify ::= localValue 32
246 setReportingState SetReportingState ::= localValue 73
247 statusReport StatusReport ::= localValue 74
248 remoteUserFree RemoteUserFree ::= localValue 75
249 istAlert IST-Alert ::= localValue 87
250 istCommand IST-Command ::= localValue 88
251
252
253 -- supplementary service handling operation codes
254
255 registerSS RegisterSS ::= localValue 10
256 eraseSS EraseSS ::= localValue 11
257 activateSS ActivateSS ::= localValue 12
258 deactivateSS DeactivateSS ::= localValue 13
259 interrogateSS InterrogateSS ::= localValue 14
260 processUnstructuredSS-Request ProcessUnstructuredSS-Request ::=
261     localValue 59
262 unstructuredSS-Request UnstructuredSS-Request ::= localValue 60
263 unstructuredSS-Notify UnstructuredSS-Notify ::= localValue 61
264 registerPassword RegisterPassword ::= localValue 17
265 getPassword GetPassword ::= localValue 18
266 registerCC-Entry RegisterCC-Entry ::= localValue 76
267 eraseCC-Entry EraseCC-Entry ::= localValue 77
268
269
270 -- short message service operation codes
271
272 sendRoutingInfoForSM SendRoutingInfoForSM ::= localValue 45
273 mo-forwardSM MO-ForwardSM ::= localValue 46
274 mt-forwardSM MT-ForwardSM ::= localValue 44
275 reportSM-DeliveryStatus ReportSM-DeliveryStatus ::= localValue 47
276 informServiceCentre InformServiceCentre ::= localValue 63
277 alertServiceCentre AlertServiceCentre ::= localValue 64
278 readyForSM ReadyForSM ::= localValue 66
279
280 -- provide subscriber info operation codes
281
282 provideSubscriberInfo ProvideSubscriberInfo ::= localValue 70
283
284 -- any time interrogation operation codes
285
286 anyTimeInterrogation AnyTimeInterrogation ::= localValue 71
287
288 -- any time information handling operation codes
289
290 anyTimeSubscriptionInterrogation AnyTimeSubscriptionInterrogation ::= localValue 62
291 anyTimeModification AnyTimeModification ::= localValue 65
292
293 -- subscriber data modification notification operation codes
294
295 noteSubscriberDataModified NoteSubscriberDataModified ::= localValue 5
296
297 -- supplementary service invocation notification operation codes
298
299 ss-InvocationNotification SS-InvocationNotification ::= localValue 72
300
301
302 --Group Call operation codes
303
304 prepareGroupCall PrepareGroupCall ::= localValue 39
305 sendGroupCallEndSignal SendGroupCallEndSignal ::= localValue 40
306 processGroupCallSignalling ProcessGroupCallSignalling ::= localValue 41
307 forwardGroupCallSignalling ForwardGroupCallSignalling ::= localValue 42
308

```

```

309
310 -- gprs location updating operation codes
311
312 updateGprsLocation UpdateGprsLocation ::= localValue 23
313
314 -- gprs location information retrieval operation codes
315
316 sendRoutingInfoForGprs SendRoutingInfoForGprs ::= localValue 24
317
318 -- failure reporting operation codes
319
320 failureReport FailureReport ::= localValue 25
321
322 -- GPRS notification operation codes
323
324 noteMsPresentForGprs NoteMsPresentForGprs ::= localValue 26
325
326 -- Location service operation codes
327
328 provideSubscriberLocation ProvideSubscriberLocation ::= localValue 83
329 sendRoutingInfoForLCS SendRoutingInfoForLCS ::= localValue 85
330 subscriberLocationReport SubscriberLocationReport ::= localValue 86
331
332
333 -- Mobility Management operation codes
334
335 noteMM-Event NoteMM-Event ::= localValue 89
336
337
338 -- Key Distribution operation codes
339
340 updateSecurityAssociation UpdateSecurityAssociation ::= localValue 53
341 noteNewSecurityAssociation NoteNewSecurityAssociation ::= localValue 82
342 securityUpdateComplete SecurityUpdateComplete ::= localValue 84
343
344
345 -- generic error codes
346
347 systemFailure SystemFailure ::= localValue 34
348 dataMissing DataMissing ::= localValue 35
349 unexpectedDataValue UnexpectedDataValue ::= localValue 36
350 facilityNotSupported FacilityNotSupported ::= localValue 21
351 incompatibleTerminal IncompatibleTerminal ::= localValue 28
352 resourceLimitation ResourceLimitation ::= localValue 51
353
354
355 -- identification and numbering error codes
356
357 unknownSubscriber UnknownSubscriber ::= localValue 1
358 numberChanged NumberChanged ::= localValue 44
359 unknownMSC UnknownMSC ::= localValue 3
360 unidentifiedSubscriber UnidentifiedSubscriber ::= localValue 5
361 unknownEquipment UnknownEquipment ::= localValue 7
362
363
364 -- subscription error codes
365
366 roamingNotAllowed RoamingNotAllowed ::= localValue 8
367 illegalSubscriber IllegalSubscriber ::= localValue 9
368 illegalEquipment IllegalEquipment ::= localValue 12
369 bearerServiceNotProvisioned BearerServiceNotProvisioned ::=
370     localValue 10
371 teleserviceNotProvisioned TeleserviceNotProvisioned ::=
372     localValue 11
373
374
375 -- handover error codes
376
377 noHandoverNumberAvailable NoHandoverNumberAvailable ::=
378     localValue 25
379 subsequentHandoverFailure SubsequentHandoverFailure ::=
380     localValue 26
381
382
383 -- operation and maintenance error codes
384
385 tracingBufferFull TracingBufferFull ::= localValue 40

```

```

386
387
388 -- call handling error codes
389
390 noRoamingNumberAvailable NoRoamingNumberAvailable ::= localValue 39
391 absentSubscriber AbsentSubscriber ::= localValue 27
392 busySubscriber BusySubscriber ::= localValue 45
393 noSubscriberReply NoSubscriberReply ::= localValue 46
394 callBarred CallBarred ::= localValue 13
395 forwardingFailed ForwardingFailed ::= localValue 47
396 or-NotAllowed OR-NotAllowed ::= localValue 48
397 forwardingViolation ForwardingViolation ::= localValue 14
398 cug-Reject CUG-Reject ::= localValue 15
399
400
401 -- any time interrogation error codes
402
403 ati-NotAllowed ATI-NotAllowed ::= localValue 49
404
405 -- any time information handling error codes
406 atsi-NotAllowed ATSI-NotAllowed ::= localValue 60
407 atm-NotAllowed ATM-NotAllowed ::= localValue 61
408 informationNotAvailable InformationNotAvailable ::= localValue 62
409
410
411 -- Group Call error codes
412 noGroupCallNumberAvailable NoGroupCallNumberAvailable ::= localValue 50
413
414
415 -- supplementary service error codes
416
417 illegalSS-Operation IllegalSS-Operation ::= localValue 16
418 ss-ErrorStatus SS-ErrorStatus ::= localValue 17
419 ss-NotAvailable SS-NotAvailable ::= localValue 18
420 ss-SubscriptionViolation SS-SubscriptionViolation ::= localValue 19
421 ss-Incompatibility SS-Incompatibility ::= localValue 20
422 unknownAlphabet UnknownAlphabet ::= localValue 71
423 ussd-Busy USSD-Busy ::= localValue 72
424 pw-RegistrationFailure PW-RegistrationFailure ::= localValue 37
425 negativePW-Check NegativePW-Check ::= localValue 38
426 numberOfPW-AttemptsViolation NumberOfPW-AttemptsViolation ::=
427     localValue 43
428 shortTermDenial ShortTermDenial ::= localValue 29
429 longTermDenial LongTermDenial ::= localValue 30
430
431
432 -- short message service error codes
433
434 subscriberBusyForMT-SMS SubscriberBusyForMT-SMS ::= localValue 31
435 sm-DeliveryFailure SM-DeliveryFailure ::= localValue 32
436 messageWaitingListFull MessageWaitingListFull ::= localValue 33
437 absentSubscriberSM AbsentSubscriberSM ::= localValue 6
438
439 -- location service error codes
440
441 unauthorizedRequestingNetwork UnauthorizedRequestingNetwork ::= localValue 52
442 unauthorizedLCSCClient UnauthorizedLCSCClient ::= localValue 53
443 positionMethodFailure PositionMethodFailure ::= localValue 54
444 unknownOrUnreachableLCSCClient UnknownOrUnreachableLCSCClient ::= localValue 58
445
446
447 -- Mobility Management error codes
448 mm-EventNotSupported MM-EventNotSupported ::= localValue 59
449
450 -- Key Distribution error codes
451 securityAssociationNotSupported SecurityAssociationNotSupported ::= localValue 15
452 keyDistributionFailure KeyDistributionFailure ::= localValue 22
453 decryptionFailure DecryptionFailure ::= localValue 41
454
455 -- The following operation codes are reserved for operations
456 -- existing in previous versions of the protocol
457

```

```

458 -- Operation Name          AC used          Oper. Code
459 --
460 -- sendParameters          map-ac infoRetrieval (14) version1 (1)    localValue 9
461 -- processUnstructuredSS-Data map-ac networkFunctionalSs (18) version1 (1) localValue 19
462 -- performHandover        map-ac handoverControl (11) version1 (1)  localValue 28
463 -- performSubsequentHandover map-ac handoverControl (11) version1 (1)  localValue 30
464 -- noteInternalHandover   map-ac handoverControl (11) version1 (1)  localValue 35
465 -- noteSubscriberPresent   map-ac mwdMngt (24) version1 (1)         localValue 48
466 -- alertServiceCentreWithoutResult map-ac shortMsgAlert (23) version1 (1)  localValue 49
467 -- traceSubscriberActivity map-ac handoverControl (11) version1 (1)  localValue 52
468 -- beginSubscriberActivity map-ac networkFunctionalSs (18) version1 (1) localValue 54
469
470 -- The following error codes are reserved for errors
471 -- existing in previous versions of the protocol
472
473 -- Error Name              AC used          Error Code
474 --
475 -- unknownBaseStation     map-ac handoverControl (11) version1 (1)  localValue 2
476 -- invalidTargetBaseStation map-ac handoverControl (11) version1 (1)  localValue 23
477 -- noRadioResourceAvailable map-ac handoverControl (11) version1 (1)  localValue 24
478
479
480 END
481
482 *****NEXT MODIFICATION*****

```

## 17.6.6 Errors

```

1  MAP-Errors {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
4
5  DEFINITIONS
6
7  ::=
8
9  BEGIN
10
11  EXPORTS
12
13      -- generic errors
14      SystemFailure,
15      DataMissing,
16      UnexpectedDataValue,
17      FacilityNotSupported,
18      IncompatibleTerminal,
19      ResourceLimitation,
20
21      -- identification and numbering errors
22      UnknownSubscriber,
23      NumberChanged,
24      UnknownMSC,
25      UnidentifiedSubscriber,
26      UnknownEquipment,
27
28      -- subscription errors
29      RoamingNotAllowed,
30      IllegalSubscriber,
31      IllegalEquipment,
32      BearerServiceNotProvisioned,
33      TeleserviceNotProvisioned,
34
35      -- handover errors
36      NoHandoverNumberAvailable,
37      SubsequentHandoverFailure,
38
39      -- operation and maintenance errors
40      TracingBufferFull,
41
42      -- call handling errors
43      OR-NotAllowed,
44      NoRoamingNumberAvailable,
45      BusySubscriber,
46      NoSubscriberReply,
47      AbsentSubscriber,
48      CallBarred,
49      ForwardingViolation,

```

```

50 ForwardingFailed,
51 CUG-Reject,
52
53 -- any time interrogation errors
54 ATI-NotAllowed,
55
56 -- any time information handling errors
57 ATSI-NotAllowed,
58 ATM-NotAllowed,
59 InformationNotAvailable,
60
61 -- supplementary service errors
62 IllegalSS-Operation,
63 SS-ErrorStatus,
64 SS-NotAvailable,
65 SS-SubscriptionViolation,
66 SS-Incompatibility,
67 UnknownAlphabet,
68 USSD-Busy,
69 PW-RegistrationFailure,
70 NegativePW-Check,
71 NumberOfPW-AttemptsViolation,
72 ShortTermDenial,
73 LongTermDenial,
74
75 -- short message service errors
76 SubscriberBusyForMT-SMS,
77 SM-DeliveryFailure,
78 MessageWaitingListFull,
79 AbsentSubscriberSM,
80
81 -- Group Call errors
82 NoGroupCallNumberAvailable,
83
84 -- location service errors
85 UnauthorizedRequestingNetwork,
86 UnauthorizedLCSCClient,
87 PositionMethodFailure,
88 UnknownOrUnreachableLCSCClient,
89
90 -- Mobility Management errors
91 MM-EventNotSupported,
92
93 -- Key Distribution errors
94 SecurityAssociationNotSupported,
95 KeyDistributionFailure,
96 DecryptionFailure
97
98
99 ;
100
101 IMPORTS
102 ERROR
103 FROM TCAPMessages {
104     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
105
106     SS-Status
107 FROM MAP-SS-DataTypes {
108     ccitt identified-organization (4) etsi (0) mobileDomain (0)
109     gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
110
111     SS-IncompatibilityCause,
112     PW-RegistrationFailureCause,
113     SM-DeliveryFailureCause,
114     SystemFailureParam,
115     DataMissingParam,
116     UnexpectedDataParam,
117     FacilityNotSupParam,
118     UnknownSubscriberParam,
119     NumberChangedParam,
120     UnidentifiedSubParam,
121     RoamingNotAllowedParam,
122     IllegalSubscriberParam,
123     IllegalEquipmentParam,
124     BearerServNotProvParam,
125     TeleservNotProvParam,
126     TracingBufferFullParam,
127     NoRoamingNbParam,
128     OR-NotAllowedParam,

```



```

129 AbsentSubscriberParam,
130 BusySubscriberParam,
131 NoSubscriberReplyParam,
132 CallBarredParam,
133 ForwardingViolationParam,
134 ForwardingFailedParam,
135 CUG-RejectParam,
136 ATI-NotAllowedParam,
137 SubBusyForMT-SMS-Param,
138 MessageWaitListFullParam,
139 AbsentSubscriberSM-Param,
140 ResourceLimitationParam,
141 NoGroupCallNbParam,
142 IncompatibleTerminalParam,
143 ShortTermDenialParam,
144 LongTermDenialParam,
145 UnauthorizedRequestingNetwork-Param,
146 UnauthorizedLCSCClient-Param,
147 PositionMethodFailure-Param,
148 UnknownOrUnreachableLCSCClient-Param,
149 MM-EventNotSupported-Param,
150 ATSI-NotAllowedParam,
151 ATM-NotAllowedParam,
152 IllegalSS-OperationParam,
153 SS-NotAvailableParam,
154 SS-SubscriptionViolationParam,
155 InformationNotAvailableParam,
156 SecurityAssociationNotSupportedParam,
157 KeyDistributionFailureParam,
158 DecryptionFailureParam
159
160
161
162 FROM MAP-ER-DataTypes {
163     ccitt identified-organization (4) etsi (0) mobileDomain (0)
164     gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}
165 ;
166
167
168 -- generic errors
169
170 SystemFailure ::= ERROR
171     PARAMETER
172         systemFailureParam          SystemFailureParam
173         -- optional
174
175 DataMissing ::= ERROR
176     PARAMETER
177         dataMissingParam            DataMissingParam
178         -- optional
179         -- dataMissingParam must not be used in version <3
180
181 UnexpectedDataValue ::= ERROR
182     PARAMETER
183         unexpectedDataParam          UnexpectedDataParam
184         -- optional
185         -- unexpectedDataParam must not be used in version <3
186
187 FacilityNotSupported ::= ERROR
188     PARAMETER
189         facilityNotSupParam          FacilityNotSupParam
190         -- optional
191         -- facilityNotSupParam must not be used in version <3
192
193 IncompatibleTerminal ::= ERROR
194     PARAMETER
195         incompatibleTerminalParam    IncompatibleTerminalParam
196         -- optional
197
198 ResourceLimitation ::= ERROR
199     PARAMETER
200         resourceLimitationParam      ResourceLimitationParam
201         -- optional
202
203 -- identification and numbering errors
204

```

```

205 UnknownSubscriber ::= ERROR
206     PARAMETER
207         unknownSubscriberParam          UnknownSubscriberParam
208         -- optional
209         -- unknownSubscriberParam must not be used in version <3
210
211 NumberChanged ::= ERROR
212     PARAMETER
213         numberChangedParam              NumberChangedParam
214         -- optional
215
216 UnknownMSC ::= ERROR
217
218 UnidentifiedSubscriber ::= ERROR
219     PARAMETER
220         unidentifiedSubParam            UnidentifiedSubParam
221         -- optional
222         -- unidentifiedSubParam must not be used in version <3
223
224 UnknownEquipment ::= ERROR
225
226
227 -- subscription errors
228
229 RoamingNotAllowed ::= ERROR
230     PARAMETER
231         roamingNotAllowedParam          RoamingNotAllowedParam
232
233 IllegalSubscriber ::= ERROR
234     PARAMETER
235         illegalSubscriberParam          IllegalSubscriberParam
236         -- optional
237         -- illegalSubscriberParam must not be used in version <3
238
239 IllegalEquipment ::= ERROR
240     PARAMETER
241         illegalEquipmentParam           IllegalEquipmentParam
242         -- optional
243         -- illegalEquipmentParam must not be used in version <3
244
245 BearerServiceNotProvisioned ::= ERROR
246     PARAMETER
247         bearerServNotProvParam          BearerServNotProvParam
248         -- optional
249         -- bearerServNotProvParam must not be used in version <3
250
251 TeleserviceNotProvisioned ::= ERROR
252     PARAMETER
253         teleservNotProvParam            TeleservNotProvParam
254         -- optional
255         -- teleservNotProvParam must not be used in version <3
256
257
258 -- handover errors
259
260 NoHandoverNumberAvailable ::= ERROR
261
262 SubsequentHandoverFailure ::= ERROR
263
264
265 -- operation and maintenance errors
266
267 TracingBufferFull ::= ERROR
268     PARAMETER
269         tracingBufferFullParam          TracingBufferFullParam
270         -- optional
271
272
273 -- call handling errors
274
275 NoRoamingNumberAvailable ::= ERROR
276     PARAMETER
277         noRoamingNbParam                NoRoamingNbParam
278         -- optional
279

```

```

280 AbsentSubscriber ::= ERROR
281     PARAMETER
282         absentSubscriberParam          AbsentSubscriberParam
283         -- optional
284         -- absentSubscriberParam must not be used in version <3
285
286
287 BusySubscriber ::= ERROR
288     PARAMETER
289         busySubscriberParam            BusySubscriberParam
290         -- optional
291
292 NoSubscriberReply ::= ERROR
293     PARAMETER
294         noSubscriberReplyParam        NoSubscriberReplyParam
295         -- optional
296
297 CallBarred ::= ERROR
298     PARAMETER
299         callBarredParam                CallBarredParam
300         -- optional
301
302 ForwardingViolation ::= ERROR
303     PARAMETER
304         forwardingViolationParam      ForwardingViolationParam
305         -- optional
306
307 ForwardingFailed ::= ERROR
308     PARAMETER
309         forwardingFailedParam         ForwardingFailedParam
310         -- optional
311
312 CUG-Reject ::= ERROR
313     PARAMETER
314         cug-RejectParam               CUG-RejectParam
315         -- optional
316
317 OR-NotAllowed ::= ERROR
318     PARAMETER
319         or-NotAllowedParam            OR-NotAllowedParam
320         -- optional
321
322
323 -- any time interrogation errors
324 ATI-NotAllowed ::= ERROR
325     PARAMETER
326         ati-NotAllowedParam           ATI-NotAllowedParam
327         -- optional
328
329 -- any time information handling errors
330 ATSI-NotAllowed ::= ERROR
331     PARAMETER
332         atsi-NotAllowedParam         ATSI-NotAllowedParam
333         -- optional
334
335 ATM-NotAllowed ::= ERROR
336     PARAMETER
337         atm-NotAllowedParam           ATM-NotAllowedParam
338         -- optional
339
340 InformationNotAvailable ::= ERROR
341     PARAMETER
342         informationNotAvailableParam  InformationNotAvailableParam
343         -- optional
344
345
346 -- supplementary service errors
347
348 IllegalSS-Operation ::= ERROR
349     PARAMETER
350         illegalSS-OperationParam     IllegalSS-OperationParam
351         -- optional
352         -- illegalSS-OperationParam must not be used in version <3
353

```

```

354 SS-ErrorStatus ::= ERROR
355     PARAMETER
356         ss-Status                               SS-Status
357         -- optional
358
359 SS-NotAvailable ::= ERROR
360     PARAMETER
361         ss-NotAvailableParam                   SS-NotAvailableParam
362         -- optional
363         -- ss-NotAvailableParam must not be used in version <3
364
365 SS-SubscriptionViolation ::= ERROR
366     PARAMETER
367         ss-SubscriptionViolationParam         SS-SubscriptionViolationParam
368         -- optional
369         -- ss-NotAvailableParam must not be used in version <3
370
371 SS-Incompatibility ::= ERROR
372     PARAMETER
373         ss-IncompatibilityCause               SS-IncompatibilityCause
374         -- optional
375
376 UnknownAlphabet ::= ERROR
377
378 USSD-Busy ::= ERROR
379
380 PW-RegistrationFailure ::= ERROR
381     PARAMETER
382         pw-RegistrationFailureCause           PW-RegistrationFailureCause
383
384 NegativePW-Check ::= ERROR
385
386 NumberOfPW-AttemptsViolation ::= ERROR
387
388 ShortTermDenial ::= ERROR
389     PARAMETER
390         shortTermDenialParam                 ShortTermDenialParam
391         -- optional
392
393 LongTermDenial ::= ERROR
394     PARAMETER
395         longTermDenialParam                   LongTermDenialParam
396         -- optional
397
398
399 -- short message service errors
400
401 SubscriberBusyForMT-SMS ::= ERROR
402     PARAMETER
403         subBusyForMT-SMS-Param               SubBusyForMT-SMS-Param
404         -- optional
405
406 SM-DeliveryFailure ::= ERROR
407     PARAMETER
408         sm-DeliveryFailureCause              SM-DeliveryFailureCause
409
410 MessageWaitingListFull ::= ERROR
411     PARAMETER
412         messageWaitListFullParam             MessageWaitListFullParam
413         -- optional
414
415 AbsentSubscriberSM ::= ERROR
416     PARAMETER
417         absentSubscriberSM-Param             AbsentSubscriberSM-Param
418         -- optional
419
420 -- Group Call errors
421
422 NoGroupCallNumberAvailable ::= ERROR
423     PARAMETER
424         noGroupCallNbParam                    NoGroupCallNbParam
425         -- optional
426
427 -- location service errors
428

```

```

429 UnauthorizedRequestingNetwork ::= ERROR
430     PARAMETER
431         unauthorizedRequestingNetwork-Param  UnauthorizedRequestingNetwork-Param
432         -- optional
433
434 UnauthorizedLCSCClient ::= ERROR
435     PARAMETER
436         unauthorizedLCSCClient-Param      UnauthorizedLCSCClient-Param
437         -- optional
438
439 PositionMethodFailure ::= ERROR
440     PARAMETER
441         positionMethodFailure-Param      PositionMethodFailure-Param
442         -- optional
443
444 UnknownOrUnreachableLCSCClient ::= ERROR
445     PARAMETER
446         unknownOrUnreachableLCSCClient-Param  UnknownOrUnreachableLCSCClient-Param
447         -- optional
448
449 MM-EventNotSupported ::= ERROR
450     PARAMETER
451         mm-EventNotSupported-Param      MM-EventNotSupported-Param
452         -- optional
453
454 -- key distribution errors
455
456 SecurityAssociationNotSupported ::= ERROR
457     PARAMETER
458         securityAssociationNotSupportedParam SecurityAssociationNotSupportedParam
459
460 KeyDistributionFailure ::= ERROR
461     PARAMETER
462         keyDistributionFailureParam      KeyDistributionFailureParam
463         -- optional
464
465 DecryptionFailure ::= ERROR
466     PARAMETER
467         decryptionFailureParam          DecryptionFailureParam
468         -- optional
469
470
471 END
472
473 ***** NEXT MODIFICATION *****

```

## 17.6.9 Secure Transport service operations

```

1  MAP-SecureTransportOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-SecureTransportOperations (26)
4      version6 (6)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     UpdateSecurityAssociation,
14     NoteNewSecurityAssociation,
15     SecurityUpdateComplete
16 ;
17
18 IMPORTS
19     OPERATION
20 FROM TCAPMessages {
21     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
22
23 SecurityAssociationNotSupported,
24 KeyDistributionFailure,
25 DecryptionFailure,
26 SystemFailure
27 FROM MAP-Errors {
28     ccitt identified-organization (4) etsi (0) mobileDomain (0)
29     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

```

```

30
31 UpdateSecurityAssociationArg,
32 UpdateSecurityAssociationRes,
33 NoteNewSecurityAssociationArg,
34 NoteNewSecurityAssociationRes,
35 SecurityUpdateCompleteArg
36 FROM MAP-ST-DataTypes {
37     ccitt identified-organization (4) etsi (0) mobileDomain (0)
38     gsm-Network (1) modules (3) map-ST-DataTypes (27) version6 (6)}
39 ;
40
41
42
43 UpdateSecurityAssociation ::= OPERATION --Timer m
44     ARGUMENT
45         updateSecurityAssociationArg    UpdateSecurityAssociationArg
46     RESULT
47         updateSecurityAssociationRes    UpdateSecurityAssociationRes
48     ERRORS {
49         SecurityAssociationNotSupported,
50         KeyDistributionFailure,
51         SystemFailure,
52         DecryptionFailure}
53
54 NoteNewSecurityAssociation ::= OPERATION --Timer m
55     ARGUMENT
56         noteNewSecurityAssociationArg    NoteNewSecurityAssociationArg
57     RESULT
58         noteNewSecurityAssociationRes    NoteNewSecurityAssociationRes
59     ERRORS {
60         SecurityAssociationNotSupported,
61         SystemFailure,
62         DecryptionFailure}
63
64 SecurityUpdateComplete ::= OPERATION
65     ARGUMENT
66         securityUpdateCompleteArg        SecurityUpdateCompleteArg
67
68
69 END
70
71 *****NEXT MODIFICATION*****

```

## 17.7.7 Error data types

```

1  MAP-ER-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     RoamingNotAllowedParam,
15     CallBarredParam,
16     CUG-RejectParam,
17     SS-IncompatibilityCause,
18     PW-RegistrationFailureCause,
19     SM-DeliveryFailureCause,
20     SystemFailureParam,
21     DataMissingParam,
22     UnexpectedDataParam,
23     FacilityNotSupParam,
24     OR-NotAllowedParam,
25     UnknownSubscriberParam,
26     NumberChangedParam,
27     UnidentifiedSubParam,
28     IllegalSubscriberParam,
29     IllegalEquipmentParam,
30     BearerServNotProvParam,
31     TeleservNotProvParam,
32     TracingBufferFullParam,
33     NoRoamingNbParam,

```

```

34 AbsentSubscriberParam,
35 BusySubscriberParam,
36 NoSubscriberReplyParam,
37 ForwardingViolationParam,
38 ForwardingFailedParam,
39 ATI-NotAllowedParam,
40 SubBusyForMT-SMS-Param,
41 MessageWaitListFullParam,
42 AbsentSubscriberSM-Param,
43 AbsentSubscriberDiagnosticSM,
44 ResourceLimitationParam,
45 NoGroupCallNbParam,
46 IncompatibleTerminalParam,
47 ShortTermDenialParam,
48 LongTermDenialParam,
49 UnauthorizedRequestingNetwork-Param,
50 UnauthorizedLCSCClient-Param,
51 PositionMethodFailure-Param,
52 UnknownOrUnreachableLCSCClient-Param,
53 MM-EventNotSupported-Param,
54 ATSI-NotAllowedParam,
55 ATM-NotAllowedParam,
56 IllegalSS-OperationParam,
57 SS-NotAvailableParam,
58 SS-SubscriptionViolationParam,
59 InformationNotAvailableParam,
60 SecurityAssociationNotSupportedParam,
61 KeyDistributionFailureParam,
62 DecryptionFailureParam
63
64
65
66
67
68 ;
69
70 IMPORTS
71   SS-Status
72 FROM MAP-SS-DataTypes {
73   ccitt identified-organization (4) etsi (0) mobileDomain (0)
74   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
75
76   SignalInfo,
77   BasicServiceCode,
78   NetworkResource
79 FROM MAP-CommonDataTypes {
80   ccitt identified-organization (4) etsi (0) mobileDomain (0)
81   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
82
83   SS-Code
84 FROM MAP-SS-Code {
85   ccitt identified-organization (4) etsi (0) mobileDomain (0)
86   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
87
88   ExtensionContainer
89 FROM MAP-ExtensionDataTypes {
90   ccitt identified-organization (4) etsi (0) mobileDomain (0)
91   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
92 ;
93

```

<pre> 94 RoamingNotAllowedParam ::= SEQUENCE { 95   roamingNotAllowedCause          RoamingNotAllowedCause, 96   extensionContainer              ExtensionContainer 97   ...} </pre>	OPTIONAL,
--	-----------

<pre> 99 RoamingNotAllowedCause ::= ENUMERATED { 100   plmnRoamingNotAllowed (0), 101   operatorDeterminedBarring (3)} </pre>
---

<pre> 103 CallBarredParam ::= CHOICE { 104   callBarringCause          CallBarringCause, 105   -- call BarringCause must not be used in version 3 106   extensibleCallBarredParam ExtensibleCallBarredParam 107   -- extensibleCallBarredParam must not be used in version &lt;3 108   } </pre>
---

109

```

110 CallBarringCause ::= ENUMERATED {
111     barringServiceActive (0),
112     operatorBarring (1)}
113
114 ExtensibleCallBarredParam ::= SEQUENCE {
115     callBarringCause          CallBarringCause          OPTIONAL,
116     extensionContainer        ExtensionContainer        OPTIONAL,
117     ... ,
118     unauthorisedMessageOriginator [1] NULL            OPTIONAL }
119
120 CUG-RejectParam ::= SEQUENCE {
121     cug-RejectCause          CUG-RejectCause          OPTIONAL,
122     extensionContainer        ExtensionContainer        OPTIONAL,
123     ...}
124
125 CUG-RejectCause ::= ENUMERATED {
126     incomingCallsBarredWithinCUG (0),
127     subscriberNotMemberOfCUG (1),
128     requestedBasicServiceViolatesCUG-Constraints (5),
129     calledPartySS-InteractionViolation (7)}
130
131 SS-IncompatibilityCause ::= SEQUENCE {
132     ss-Code                   [1] SS-Code              OPTIONAL,
133     basicService              BasicServiceCode         OPTIONAL,
134     ss-Status                  [4] SS-Status           OPTIONAL,
135     ...}
136
137 PW-RegistrationFailureCause ::= ENUMERATED {
138     undetermined (0),
139     invalidFormat (1),
140     newPasswordsMismatch (2)}
141
142
143 SM-EnumeratedDeliveryFailureCause ::= ENUMERATED {
144     memoryCapacityExceeded (0),
145     equipmentProtocolError (1),
146     equipmentNotSM-Equipped (2),
147     unknownServiceCentre (3),
148     sc-Congestion (4),
149     invalidSME-Address (5),
150     subscriberNotSC-Subscriber (6)}
151
152 SM-DeliveryFailureCause ::= SEQUENCE {
153     sm-EnumeratedDeliveryFailureCause SM-EnumeratedDeliveryFailureCause,
154     diagnosticInfo                    SignalInfo              OPTIONAL,
155     extensionContainer                ExtensionContainer        OPTIONAL,
156     ...}
157
158 AbsentSubscriberSM-Param ::= SEQUENCE {
159     absentSubscriberDiagnosticSM      AbsentSubscriberDiagnosticSM  OPTIONAL,
160     -- AbsentSubscriberDiagnosticSM can be either for non-GPRS
161     -- or for GPRS
162     extensionContainer                ExtensionContainer        OPTIONAL,
163     ... ,
164     additionalAbsentSubscriberDiagnosticSM [0] AbsentSubscriberDiagnosticSM OPTIONAL }
165     -- if received, additionalAbsentSubscriberDiagnosticSM
166     -- is for GPRS and absentSubscriberDiagnosticSM is
167     -- for non-GPRS
168
169 AbsentSubscriberDiagnosticSM ::= INTEGER (0..255)
170     -- AbsentSubscriberDiagnosticSM values are defined in ETS 300 536 (GSM 03.40)
171
172 SystemFailureParam ::= CHOICE {
173     networkResource                  NetworkResource,
174     -- networkResource must not be used in version 3
175     extensibleSystemFailureParam      ExtensibleSystemFailureParam
176     -- extensibleSystemFailureParam must not be used in version <3
177 }
178
179 ExtensibleSystemFailureParam ::= SEQUENCE {
180     networkResource                  NetworkResource          OPTIONAL,
181     extensionContainer                ExtensionContainer        OPTIONAL,
182     ...}
183

```



184	<b>DataMissingParam</b> ::= SEQUENCE {		
185	extensionContainer	ExtensionContainer	OPTIONAL,
186	...}		
187			
188	<b>UnexpectedDataParam</b> ::= SEQUENCE {		
189	extensionContainer	ExtensionContainer	OPTIONAL,
190	...}		
191			
192	<b>FacilityNotSupParam</b> ::= SEQUENCE {		
193	extensionContainer	ExtensionContainer	OPTIONAL,
194	...}		
195			
196	<b>OR-NotAllowedParam</b> ::= SEQUENCE {		
197	extensionContainer	ExtensionContainer	OPTIONAL,
198	...}		
199			
200	<b>UnknownSubscriberParam</b> ::= SEQUENCE {		
201	extensionContainer	ExtensionContainer	OPTIONAL,
202	...,		
203	unknownSubscriberDiagnostic	UnknownSubscriberDiagnostic	OPTIONAL}
204			
205	<b>UnknownSubscriberDiagnostic</b> ::= ENUMERATED {		
206	imsiUnknown (0),		
207	gprsSubscriptionUnknown (1),		
208	...,		
209	npdbMismatch (2)}		
210	-- if unknown values are received in		
211	-- unknownSubscriberDiagnostic they shall be discarded		
212			
213			
214	<b>NumberChangedParam</b> ::= SEQUENCE {		
215	extensionContainer	ExtensionContainer	OPTIONAL,
216	...}		
217			
218	<b>UnidentifiedSubParam</b> ::= SEQUENCE {		
219	extensionContainer	ExtensionContainer	OPTIONAL,
220	...}		
221			
222	<b>IllegalSubscriberParam</b> ::= SEQUENCE {		
223	extensionContainer	ExtensionContainer	OPTIONAL,
224	...}		
225			
226	<b>IllegalEquipmentParam</b> ::= SEQUENCE {		
227	extensionContainer	ExtensionContainer	OPTIONAL,
228	...}		
229			
230	<b>BearerServNotProvParam</b> ::= SEQUENCE {		
231	extensionContainer	ExtensionContainer	OPTIONAL,
232	...}		
233			
234	<b>TeleservNotProvParam</b> ::= SEQUENCE {		
235	extensionContainer	ExtensionContainer	OPTIONAL,
236	...}		
237			
238	<b>TracingBufferFullParam</b> ::= SEQUENCE {		
239	extensionContainer	ExtensionContainer	OPTIONAL,
240	...}		
241			
242	<b>NoRoamingNbParam</b> ::= SEQUENCE {		
243	extensionContainer	ExtensionContainer	OPTIONAL,
244	...}		
245			
246	<b>AbsentSubscriberParam</b> ::= SEQUENCE {		
247	extensionContainer	ExtensionContainer	OPTIONAL,
248	...,		
249	absentSubscriberReason	[0] AbsentSubscriberReason	OPTIONAL}
250			

```

251 AbsentSubscriberReason ::= ENUMERATED {
252     imsiDetach (0),
253     restrictedArea (1),
254     noPageResponse (2),
255     ... ,
256     purgedMS (3)}
257 -- exception handling: at reception of other values than the ones listed the
258 -- AbsentSubscriberReason shall be ignored.
259 -- The AbsentSubscriberReason: purgedMS is defined for the Super-Charger feature
260 -- (see TS 23.116). If this value is received in a Provide Roaming Number response
261 -- it shall be mapped to the AbsentSubscriberReason: imsiDetach in the Send Routing
262 -- Information response
263
264 BusySubscriberParam ::= SEQUENCE {
265     extensionContainer          ExtensionContainer          OPTIONAL,
266     ...,
267     ccbs-Possible              [0] NULL                  OPTIONAL,
268     ccbs-Busy                  [1] NULL                  OPTIONAL}
269
270 NoSubscriberReplyParam ::= SEQUENCE {
271     extensionContainer          ExtensionContainer          OPTIONAL,
272     ...}
273
274 ForwardingViolationParam ::= SEQUENCE {
275     extensionContainer          ExtensionContainer          OPTIONAL,
276     ...}
277
278 ForwardingFailedParam ::= SEQUENCE {
279     extensionContainer          ExtensionContainer          OPTIONAL,
280     ...}
281
282 ATI-NotAllowedParam ::= SEQUENCE {
283     extensionContainer          ExtensionContainer          OPTIONAL,
284     ...}
285
286 ATSI-NotAllowedParam ::= SEQUENCE {
287     extensionContainer          ExtensionContainer          OPTIONAL,
288     ...}
289
290 ATM-NotAllowedParam ::= SEQUENCE {
291     extensionContainer          ExtensionContainer          OPTIONAL,
292     ...}
293
294 IllegalSS-OperationParam ::= SEQUENCE {
295     extensionContainer          ExtensionContainer          OPTIONAL,
296     ...}
297
298 SS-NotAvailableParam ::= SEQUENCE {
299     extensionContainer          ExtensionContainer          OPTIONAL,
300     ...}
301
302 SS-SubscriptionViolationParam ::= SEQUENCE {
303     extensionContainer          ExtensionContainer          OPTIONAL,
304     ...}
305
306 InformationNotAvailableParam ::= SEQUENCE {
307     extensionContainer          ExtensionContainer          OPTIONAL,
308     ...}
309
310 SubBusyForMT-SMS-Param ::= SEQUENCE {
311     extensionContainer          ExtensionContainer          OPTIONAL,
312     ...,
313     gprsConnectionSuspended    NULL                  OPTIONAL }
314 -- If GprsConnectionSuspended is not understood it shall
315 -- be discarded
316
317 MessageWaitListFullParam ::= SEQUENCE {
318     extensionContainer          ExtensionContainer          OPTIONAL,
319     ...}
320
321 ResourceLimitationParam ::= SEQUENCE {
322     extensionContainer          ExtensionContainer          OPTIONAL,
323     ...}
324

```

```

325 NoGroupCallNbParam ::= SEQUENCE {
326     extensionContainer          ExtensionContainer          OPTIONAL,
327     ...}
328
329 IncompatibleTerminalParam ::= SEQUENCE {
330     extensionContainer          ExtensionContainer          OPTIONAL,
331     ...}
332
333 ShortTermDenialParam ::= SEQUENCE {
334     ...}
335
336 LongTermDenialParam ::= SEQUENCE {
337     ...}
338
339 UnauthorizedRequestingNetwork-Param ::= SEQUENCE {
340     extensionContainer          ExtensionContainer          OPTIONAL,
341     ...}
342
343 UnauthorizedLCSCClient-Param ::= SEQUENCE {
344     unauthorizedLCSCClient-Diagnostic [0] UnauthorizedLCSCClient-Diagnostic OPTIONAL,
345     extensionContainer                [1] ExtensionContainer          OPTIONAL,
346     ...}
347
348 UnauthorizedLCSCClient-Diagnostic ::= ENUMERATED {
349     noAdditionalInformation (0),
350     clientNotInMSPPrivacyExceptionList (1),
351     callToClientNotSetup (2),
352     privacyOverrideNotApplicable (3),
353     disallowedByLocalRegulatoryRequirements (4),
354     ...}
355 -- exception handling:
356 -- any unrecognized value shall be ignored
357
358 PositionMethodFailure-Param ::= SEQUENCE {
359     positionMethodFailure-Diagnostic [0] PositionMethodFailure-Diagnostic OPTIONAL,
360     extensionContainer                [1] ExtensionContainer          OPTIONAL,
361     ...}
362
363 PositionMethodFailure-Diagnostic ::= ENUMERATED {
364     congestion (0),
365     insufficientResources (1),
366     insufficientMeasurementData (2),
367     inconsistentMeasurementData (3),
368     locationProcedureNotCompleted (4),
369     locationProcedureNotSupportedByTargetMS (5),
370     qosNotAttainable (6),
371     ...}
372 -- exception handling:
373 -- any unrecognized value shall be ignored
374
375 UnknownOrUnreachableLCSCClient-Param ::= SEQUENCE {
376     extensionContainer          ExtensionContainer          OPTIONAL,
377     ...}
378
379 MM-EventNotSupported-Param ::= SEQUENCE {
380     extensionContainer          ExtensionContainer          OPTIONAL,
381     ...}
382
383 SecurityAssociationNotSupportedParam ::= SEQUENCE {
384     unsupportedAC-OrOperation      [0] NULL                OPTIONAL,
385     algorithmNotSupported          [1] NULL                OPTIONAL,
386     modeOfOperationNotSupported   [2] NULL                OPTIONAL,
387     extensionContainer             ExtensionContainer          OPTIONAL,
388     ...}
389
390 KeyDistributionFailureParam ::= SEQUENCE {
391     extensionContainer            ExtensionContainer          OPTIONAL,
392     ...}
393
394 DecryptionFailureParam ::= SEQUENCE {
395     extensionContainer            ExtensionContainer          OPTIONAL,
396     ...}
397
398
399
400 END
401

```

## 17.7.14 Secure Transport data types

```

1  MAP-ST-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-ST-DataTypes (27) version6 (6)}
4
5  DEFINITIONS
6  IMPLICIT TAGS
7  ::=
8  BEGIN
9
10 EXPORTS
11     SecurityHeader,
12     ProtectedPayload,
13     SecurityEnvelopeArg,
14     SecurityEnvelopeRes,
15     UpdateSecurityAssociationArg,
16     UpdateSecurityAssociationRes,
17     NoteNewSecurityAssociationArg,
18     NoteNewSecurityAssociationRes,
19     SecurityUpdateCompleteArg
20 ;
21
22 IMPORTS
23     ISDN-AddressString
24 FROM MAP-CommonDataTypes {
25     ccitt identified-organization (4) etsi (0) mobileDomain (0)
26     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
27
28 ;
29
30
31 ProtectionMode ::= ENUMERATED {
32     noProtection (0),
33     integrityAuthenticity (1),
34     confidentialityIntegrityAuthenticity (2)}
35
36 EncryptionAlgorithmIdentifier ::= ENUMERATED {
37     beano (0),
38     des (1),
39     triple-DES (2),
40     idea (3),
41     ...}
42
43 ModeOfOperation ::= ENUMERATED {
44     ecb (0),
45     cbc (1),
46     cfb (2),
47     ofb (3),
48     ...}
49     -- modes of operation are defined in ISO/IEC 10116 (1991)
50
51 KeyVersionNumber ::= INTEGER (0..127)
52
53 OperationCode ::= CHOICE {
54     localValue INTEGER,
55     globalValue OBJECT IDENTIFIER}
56
57 UpdateSecurityAssociationArg ::= SEQUENCE {
58     plmn-Id ISDN-AddressString,
59     layer-I-encryptedSecurityAssociation Layer-I-EncryptedSecurityAssociation,
60     ...}
61
62 Layer-I-EncryptedSecurityAssociation ::= OCTET STRING (SIZE (1..1000))
63     -- Length of Layer-I-EncryptedSecurityAssociation is adjusted according to the
64     -- capabilities of lower protocol layers
65     -- The Layer-I-EncryptedSecurityAssociation carries the result of applying the
66     -- layer I encryption function specified in TS 33.102 to the encoding of the
67     -- SecurityAssociation.
68
69

```

```

70 SecurityAssociation ::= SEQUENCE {
71     encryptionAlgorithmIdentifier [0] EncryptionAlgorithmIdentifier OPTIONAL,
72     keyVersionNumber              [2] KeyVersionNumber           OPTIONAL,
73     key                            [3] Key                       OPTIONAL,
74     protectionModesForDialogueList [4] ProtectionModesForDialogueList OPTIONAL,
75     removeAll                      [5] NULL                     OPTIONAL,
76     ...}
77
78 Key ::= OCTET STRING (SIZE (8..16))
79
80 ProtectionModesForDialogueList ::= SEQUENCE SIZE (1..100) OF
81     ProtectionModesForDialogue
82
83 ProtectionModesForDialogue ::= SEQUENCE {
84     dialogueId DialogueId,
85     add-removeIndicator Add-RemoveIndicator,
86     ...}
87
88 DialogueId ::= SEQUENCE {
89     ac OBJECT IDENTIFIER,
90     firstOperation OperationCode,
91     ...}
92
93 Add-RemoveIndicator ::= CHOICE {
94     add ProtectionInfo,
95     remove NULL,
96     }
97
98 ProtectionInfo ::= SEQUENCE {
99     userInfoProtectionMode [1] Mode OPTIONAL,
100    componentProtectionModeList [2] ComponentProtectionModeList,
101    ...}
102
103 ComponentProtectionModeList ::= SEQUENCE SIZE (1..10) OF
104     ComponentProtectionModes
105
106 ComponentProtectionModes ::= SEQUENCE {
107     operation OperationCode,
108     argument Mode,
109     result [0] Mode OPTIONAL,
110     error [1] Mode OPTIONAL,
111     ...}
112
113 Mode ::= SEQUENCE {
114     protectionMode ProtectionMode,
115     modeOfOperation ModeOfOperation OPTIONAL,
116     ...}
117
118 UpdateSecurityAssociationRes ::= SEQUENCE {
119     ...
120     }
121
122
123 NoteNewSecurityAssociationArg ::= SEQUENCE {
124     direction Direction,
125     plmn-Id ISDN-AddressString,
126     layer-II-encryptedSecurityAssociation Layer-II-EncryptedSecurityAssociation,
127     ...}
128
129 Direction ::= ENUMERATED {
130     sending (0),
131     receiving (1)
132     }
133
134 Layer-II-EncryptedSecurityAssociation ::= OCTET STRING (SIZE (1..1000))
135     -- Length of Layer-II-EncryptedSecurityAssociation is adjusted according to the
136     -- capabilities of lower protocol layers
137     -- The Layer-II-EncryptedSecurityAssociation carries the result of applying the
138     -- layer II encryption function specified in TS 33.102 to the encoding of the
139     -- SecurityAssociation.
140
141
142

```

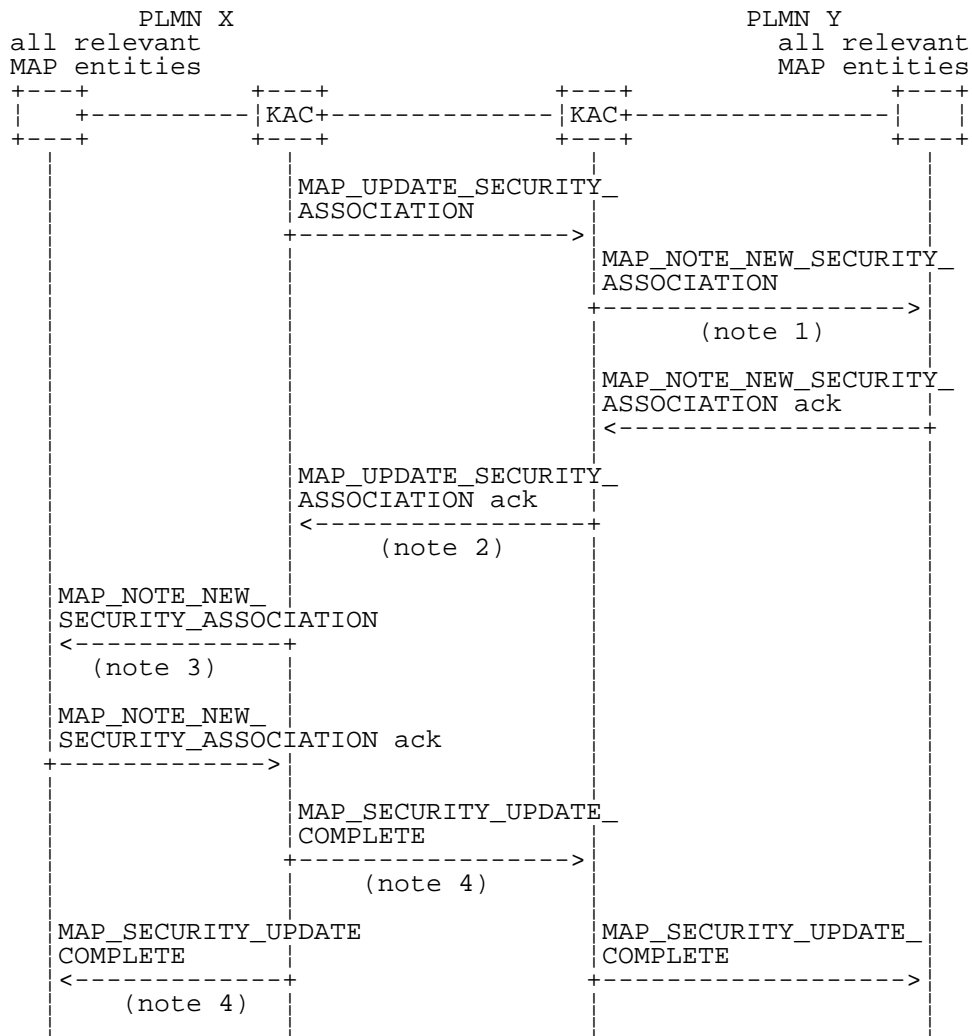
```
143 NoteNewSecurityAssociationRes ::= SEQUENCE {  
144     ...  
145 }  
146  
147 SecurityUpdateCompleteArg ::= SEQUENCE {  
148     successIndicator          NULL OPTIONAL,  
149     ...  
150 }  
151  
152 END  
153  
154
```

## 24B Key Distribution procedures

### 24B.1 General

The MAP Key Distribution procedures are used to establish a new security association between two PLMNs. After successful establishment of the new security association a previously established security association becomes invalid.

The message flows for successful security association establishment are shown in figure 24B.1/1:



Notes:

NOTE 1: To all relevant MAP entities in PLMN Y.

NOTE 2: MAP\_UPDATE\_SECURITY\_ASSOCIATION ack is sent as soon as all MAP\_NOTE\_NEW\_SECURITY\_ASSOCIATION ack messages have been received.

NOTE 3: To all relevant MAP entities in PLMN X.

NOTE 4: MAP\_SECURITY\_UPDATE\_COMPLETE is sent as soon as all MAP\_NOTE\_NEW\_SECURITY\_ASSOCIATION ack messages have been received.

**Figure 24B.1/1: Message flow for successful Security Association establishment**

## 24B.2 Processes in the MAP Entities

The text in this clause (if any) is a supplement to the definition in the SDL diagrams; it does not duplicate the information in the SDL diagrams.

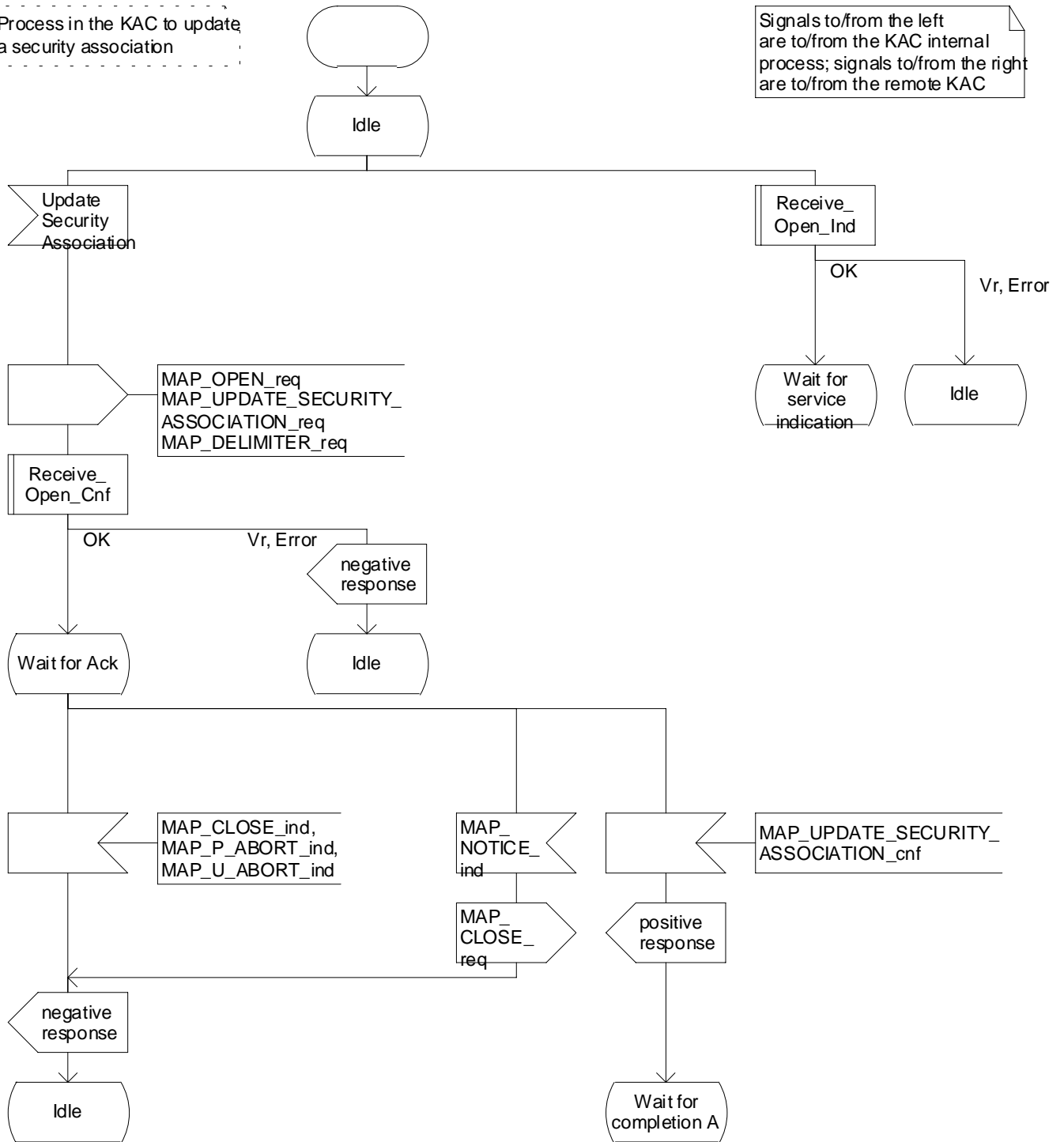
### 24B.2.1 Process in the KAC



# Process USA\_KAC

Process in the KAC to update a security association

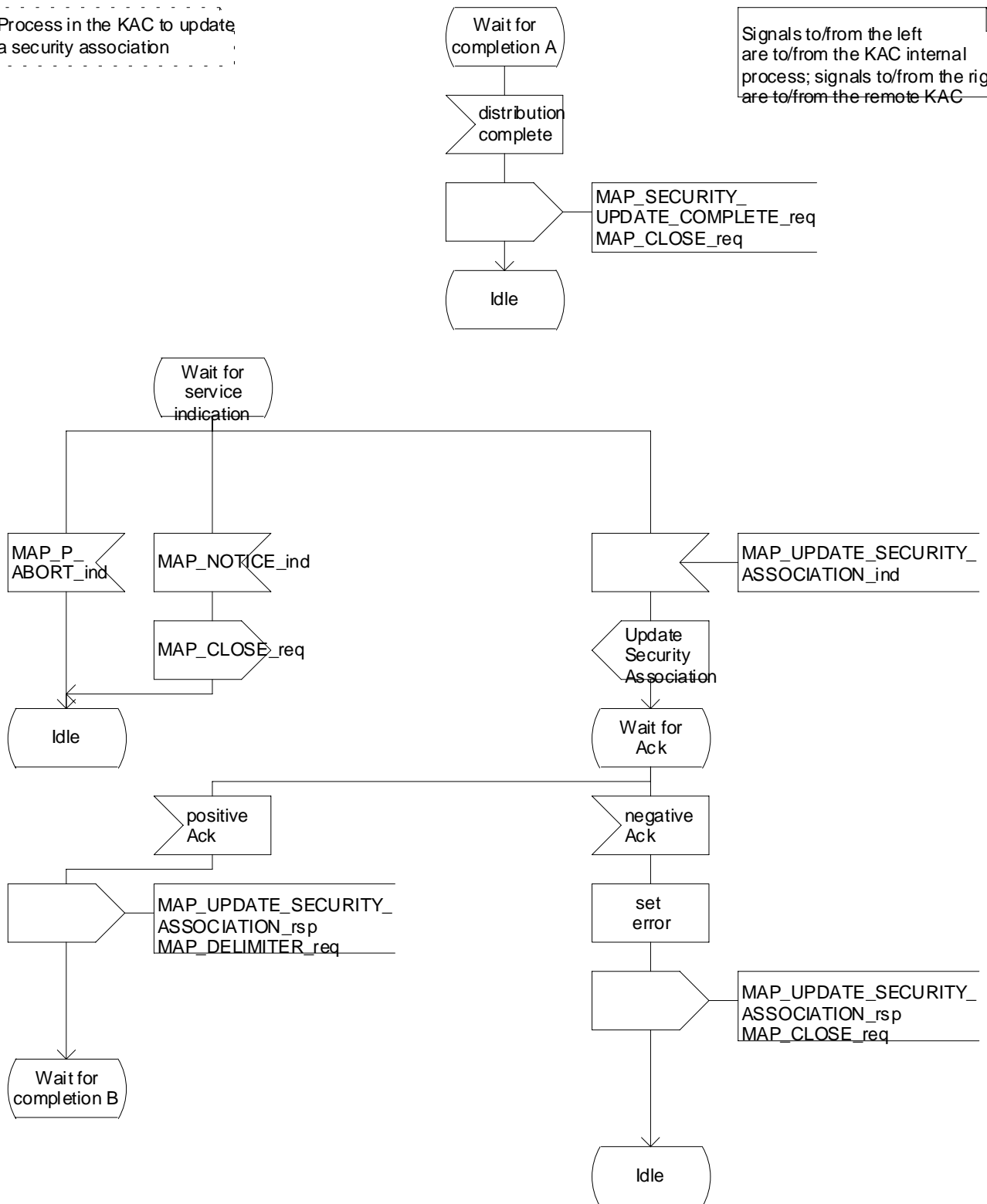
Signals to/from the left are to/from the KAC internal process; signals to/from the right are to/from the remote KAC



# Process USA\_KAC

Process in the KAC to update a security association

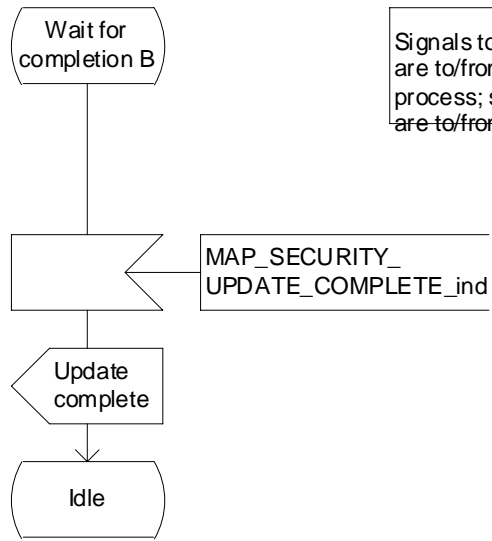
Signals to/from the left are to/from the KAC internal process; signals to/from the right are to/from the remote KAC



# Process USA\_KAC

3(

Process in the KAC to update a security association



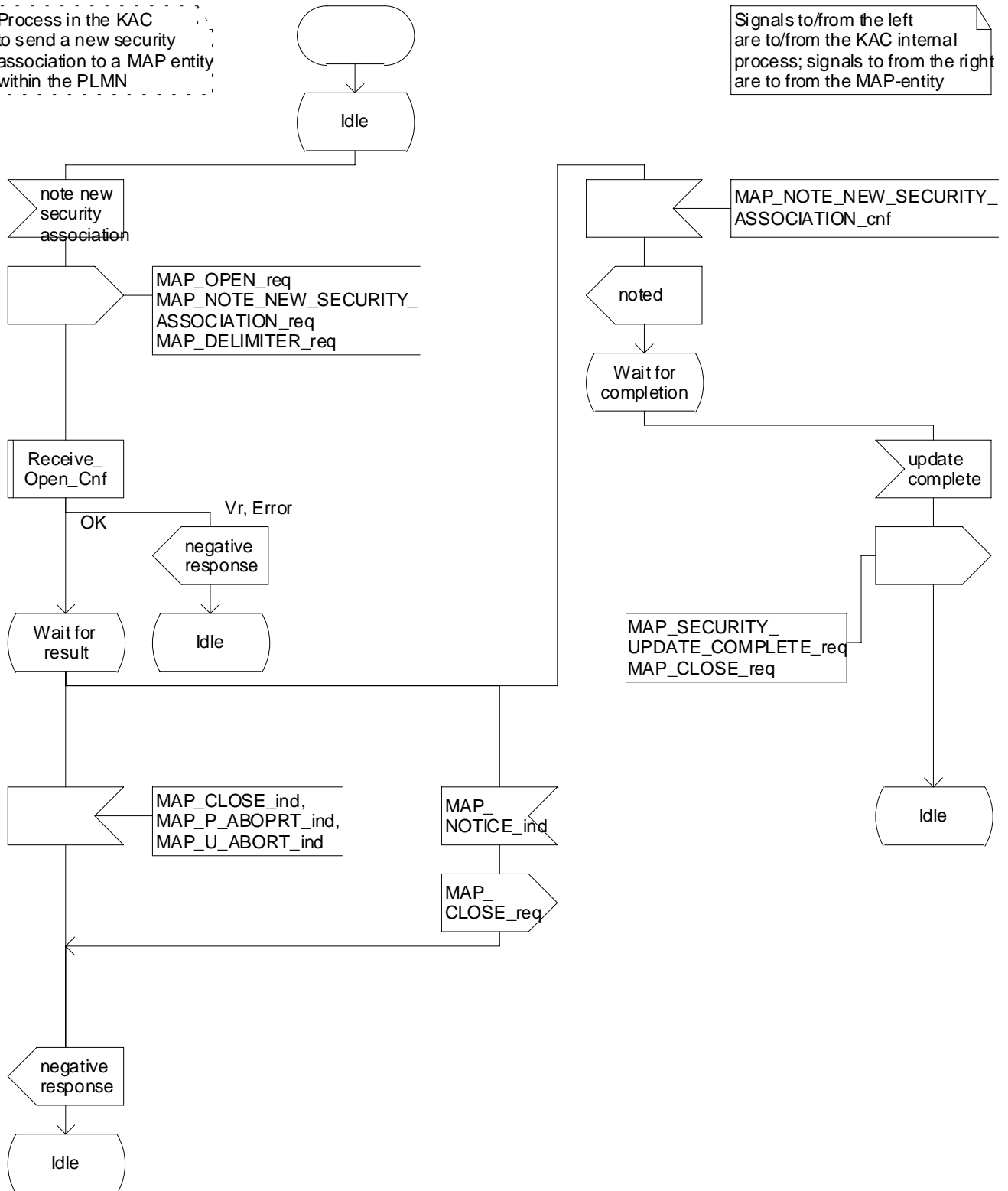
Signals to/from the left are to/from the KAC internal process; signals to/from the right are to/from the remote KAC



# Process NNS\_KAC

Process in the KAC to send a new security association to a MAP entity within the PLMN

Signals to/from the left are to/from the KAC internal process; signals to from the right are to from the MAP-entity



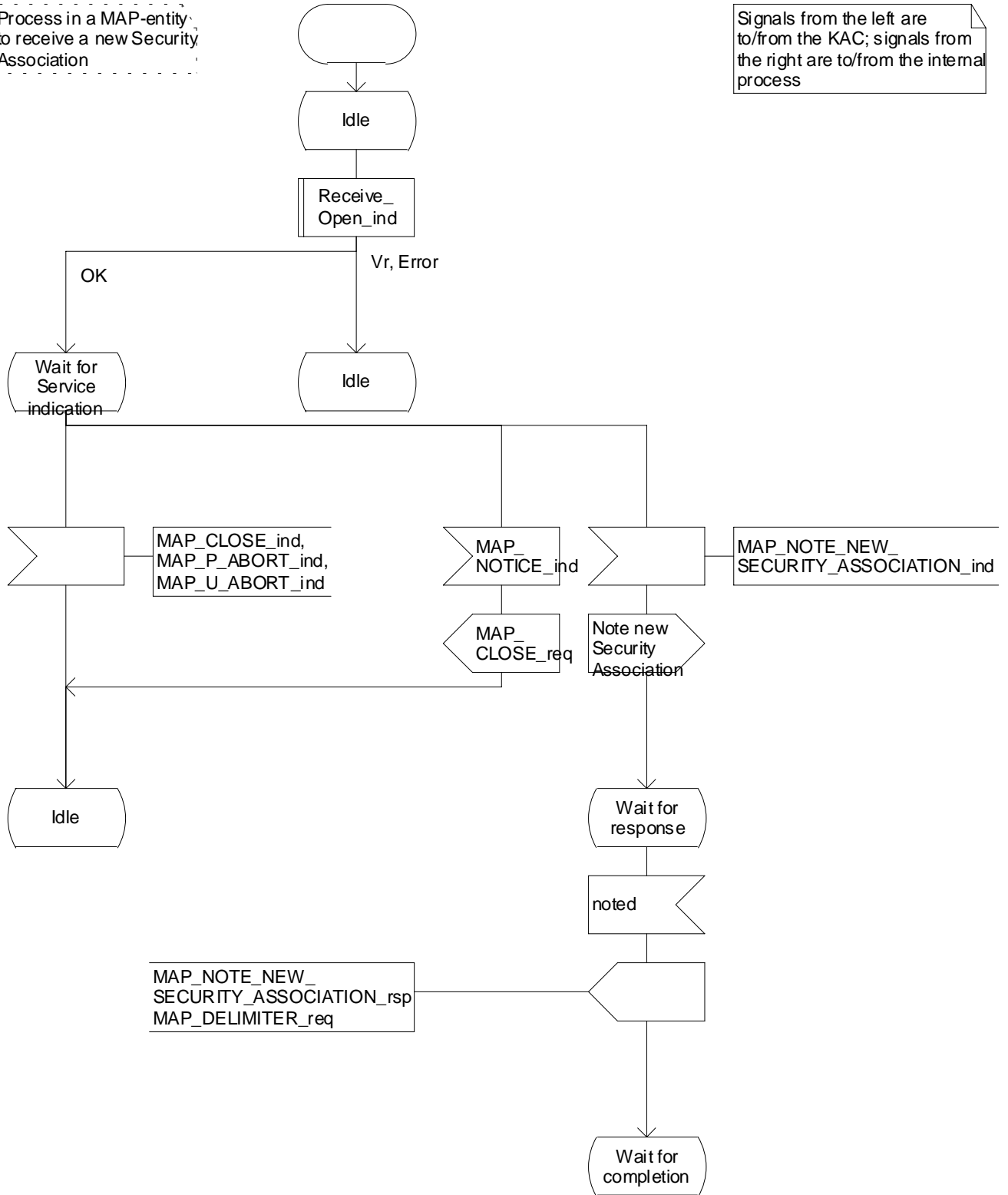
24B.2.2 Process in the relevant MAP entities

1(

Process NNS\_MAP\_ENTITY

Process in a MAP-entity to receive a new Security Association

Signals from the left are to/from the KAC; signals from the right are to/from the internal process



# Process NNS\_MAP\_ENTITY

2(

Process in a MAP-entity to receive a new Security Association

Signals from the left are to/from the KAC; signals from the right are to/from the internal process

