3GPP2 X.S0004-691-E
Version 1.0
Date: July 2005

Annexes

Revision History

Revision		Date
Rev. 1	Initial Publication	July 2005

PART 691

1 ANNEX A: PROCEDURES FOR RANDC VERIFICATION

This annex is informative and is not considered part of this Standard.

This annex describes an algorithm for verifying the value of RANDC received from an MS. The algorithm used to verify RANDC is internal to an MSC and, as such, it does not constitute an intersystem operation.

When an MSC receives RANDC(*ms*), the RANDC value reported by an MS, it shall perform the following:

1 IF the value of RANDC(*ms*) equals *current* RANDC[1]:

1-1 Return to calling task, indicating that RANDC(*ms*) is valid and that the Random Number is *current* RAND.

2 ELSEIF the value of RANDC(*ms*) equals *pending* RANDC:

2-1 Return to calling task, indicating that RANDC(*ms*) is valid and that the Random Number is *pending* RAND.

3 ELSEIF the value of RANDC(*ms*) equals ZERO:

3-1 Return to calling task, indicating that RANDC(*ms*) is valid and that the Random Number is ZERO.

4 ELSE:

4-1 Return to calling task, indicating that RANDC(*ms*) is invalid.

5 ENDIF.

[1] RAND(*n*) is the random number currently transmitted by the MSC in the OMT of the serving cell. RAND(*n*-1) is the random number transmitted in the OMT of the serving cell immediately prior to RAND(*n*). RANDC is the eight most significant bits of RAND.

2 ANNEX B: PROCEDURES FOR SSD MANAGEMENT AT AC

This annex is informative and is not considered part of this Standard.

This annex describes an algorithm for managing updates to the SharedSecretData (SSD) at the AC. The algorithm used to manage updates is internal to an AC and, as such, it does not constitute an intersystem operation.

When an AC initiates updating of an MS's SSD, it performs the following:

1 A new value of SSD is calculated and is stored in the AC's database as the *pending* Shared Secret Data, *pending* SSD; the current value of the Shared Secret Data, *current* SSD, is also retained.

2 IF AUTHR(*ms*), an Authentication Result from the MS, is received through an AuthenticationRequest INVOKE:

2-1 Compute *current* AUTHR using the value of *current* SSD.

2-2 IF *current* SSD is not equal to *pending* SSD:

2-2-1 Compute *pending* AUTHR using the value of *pending* SSD.

2-3 ENDIF.

2-4 IF AUTHR(*ms*) equals *current* AUTHR:

2-4-1 Return to calling task, indicating that authentication was successful.

2-5 ELSEIF *current* SSD is not equal to *pending* SSD and AUTHR(*ms*) equals *pending* AUTHR:

2-5-1 Discard *current* SSD.

2-5-2 Store the value of *pending* SSD as the current value.

2-5-3 Return to calling task, indicating that authentication was successful.

2-6 ELSE:

2-6-1 Return to calling task, indicating that authentication was not successful.

2-7 ENDIF.

3 ENDIF.

4 IF a BaseStationChallenge INVOKE is received:

4-1 Compute an Authentication Result (AUTHBS) using *pending* SSD and the Random Number (RANDBS) received in the BaseStationChallenge INVOKE.

4-2 Return to calling task.

5 ENDIF.

6 IF an AuthenticationStatusReport INVOKE is received indicating that SSD updating was successful:

6-1 Discard *current* SSD.

6-2 Store the value of *pending* SSD as the current value.

6-3 Return to calling task.

7 ENDIF.

8 IF an AuthenticationStatusReport INVOKE is received indicating that SSD updating was not successful:

8-1 Discard *pending* SSD.

8-2 Store *current* SSD in *pending* SSD.

9 ENDIF.

10 Return to calling task.

3 ANNEX C: AUTHENTICATION RESPONSE VERIFICATION

This annex is informative and is not considered part of this Standard.

This annex describes algorithms for verifying authentication responses at the AC and the VLR (when SSD is shared). These algorithms are used to verify responses and are internal to the AC and VLR, and as such, do not constitute intersystem operations.

AUTHR ≠ AUTHR

When an AC/HLR detects that its calculated value of AUTHR using the *current* SSD or *pending* SSD (see Annex B) is not equal to the value received from the MS, then:

1 Optionally perform Unique Challenge.

2 IF the Unique Challenge is not performed OR IF (AuthenticationResponseUnique **AUTHU ≠ AUTHU** :

2-1 Deny or allow service according to the authentication procedures of the service provider.

3 ENDIF.

When the VLR (if SSD is shared with the current serving system) detects that its calculated value of AUTHR is not equal to the value received from the MS, then:

4 IF the MS is marked *pending* SSD update:

4-1 Forward AuthenticationRequest INVOKE to the AC/HLR.

5 ELSE:

5-1 Send the appropriate Security Status Report to the AC/HLR.

5-2 Provide call treatment as specified in the RETURN RESULT.

6 ENDIF.

(Note: if access is denied, the HLR can optionally take additional protective steps; such as, instructing the VLR to change the authorization period of the mobile)

AUTHU ≠ AUTHU

When an AC/HLR detects that its calculated value of AuthenticationResponseUnique (AUTHU) using *current* SSD or *pending* SSD (see Annex B) is not equal to the value received from the MS, then deny or allow service according to the authentication procedures of the service provider at the AC/HLR.

When the MSC detects that its value of AUTHU is not equal to the value received from the MS, then:

1 Send the appropriate AuthenticationStatusReport INVOKE to the AC/HLR.

2 Provide call treatment as specified in the RETURN RESULT. If the Unique Challenge was performed due to an SSD update, the AC/HLR may decide to perform an SSD update when the next transaction occurs.

(Note: if access is denied, the HLR can optionally take additional protective steps; such as, instructing the VLR to change the authorization period of the mobile)

AUTHUBS ≠ AUTHBS

When an SSD update failure occurs, the MS shall notify the MSC. (e.g., the MS indicates an SSD update failure in the SSD update Confirmation/ACK, according to *TDMA*.) The MSC shall send the AC/HLR a failure indicator in the AuthenticationStatusReport INVOKE. Authentication procedures at the AC/HLR may result in one of the following actions:

1 Attempt to update the SSD at the MS.

2 Maintain the old SSD at the AC/HLR and allow service for the MS at the Visited Serving System.

3 Maintain the old SSD at the AC/HLR and deny service for the MS at the Visited Serving System.

4 ANNEX D: SMS AIR INTERFACE DELIVERY POINT-TO-POINT

This annex is informative and is not considered part of this Standard.

The following tables describe the parameters used in the illustrative SMD-REQUEST, SMD-ACK, and SMD-NAK messages. These messages must be converted into the appropriate actual air interface messages. It is further assumed that a given message transaction can be correlated across the air interface, so the address parameters may not be present in the responses depending on the underlying air interface technology.

Table 1 SMD-REQUEST Parameters

SMD-REQUEST Parameters	**Timer: SADT or SAOT**	
Field	**Type**	**Notes**
ElectronicSerialNumber	O	a
MSID	O	a
BearerData	M	
TeleserviceIdentifier	O	b
DestinationAddress	O	c
OriginalDestinationAddress	O	d
OriginalDestinationSubaddress	O	e
OriginalOriginatingAddress	O	f
OriginalOriginatingSubaddress	O	e
OriginatingAddress	O	g
TransactionID	O	h

Notes:

a. Include to identify the MS on the air interface.

b. Include on air interfaces that support more than one teleservice.

c. Include to identify an intermediate destination (e.g., originating MC) if the SMD-REQUEST is mobile originated, if supported by the specific air interface protocol and if not carried by the underlying data transport.

d. Include to identify the destination SME. Required for mobile originated SMD-REQUEST. Include for mobile terminated SMD-REQUEST if supported by the underlying air interface and if different from the air interface destination address (e.g., MSID).

e. Include if applicable.

f. Include to identify the originating SME. Required for mobile terminated SMD-REQUEST. Include for mobile originated SMD-REQUEST if supported by the underlying air interface and if different from the underlying air interface originating address (e.g., MSID).

g. May be included to indicate the MC for mobile terminated SMD-REQ.

h. Include if SMS origination initiated by the MS, for TDMA.

Table 2 SMD-ACK Parameters

SMD-ACK Parameters		
Field	**Type**	**Notes**
BearerData	O	a
CauseCode	O	b
OriginalDestinationAddress	O	b
TransactionID	O	c

Notes:

a. Include if applicable.

b. Include if required by underlying air interface.

c. Include if SMS origination initiated by the MS, for TDMA.

Table 3 SMD-NAK Parameters

SMD-NAK Parameters	**Type**	**Notes**
Field	**Type**	**Notes**
CauseCode	M	
OriginalDestinationAddress	O	a

Notes:

a. Include if required by underlying air interface

The following table reflects a mapping of the SMD-REQUEST parameters with various air interfaces.

Table 4 Mapping of Air Interface Parameters to SMD-REQUEST, SMS-ACK, and SMD-NAK Parameters

Parameter Mapping			
SMD-REQUEST, SMD-ACK or SMD-NAK	***AMPS* Equivalent**	***CDMA* Equivalent**	***TDMA* Equivalent**
ElectronicSerialNumber	carried in lower protocol layers		
MSID	carried in lower protocol layers		
BearerData		SMS Point-to-Point / Bearer Data	R-Data/ R-Data Unit
TeleserviceIdentifier	Not applicable	SMS Point-to-Point / Teleservice ID	R-Data/ R-Data Unit/Higher Leyer Protocol ID
OriginatingAddress	Not applicable	Not applicable	Teleservice Server Address (MS Terminated R-DATA)
DestinationAddress	Not applicable	Not applicable	Teleservice Server Address (MS Terminated R-DATA)
OriginalDestination-Subaddress	Not applicable	SMS Point-to-Point / DestinationSubaddress	R-Data/ User Destination Subaddress
OriginalOriginating Address	Not applicable	SMS Point-to-Point / OriginatingAddress	R-Data/ R-Data Unit User Originating Address
OriginalOriginating Subaddress	Not applicable	SMS Point-to-Point / OriginatingSubaddress	R-Data/ User Originating Subaddress
OriginalDestination-Address	Not applicable	SMS Point-to-Point / DestinationAddress	R-Data/ User Destination Address
CauseCode	Not applicable	SMS Point-to-Point / Destination Address	R-Data/ R-Data Unit

4.1 MSC Initiating SMD-REQUEST Toward an MS-Based SME

Upon request to send an SMS point-to-point message across the air interface, the MSC shall do the following:

1 IF the original message destination address is not the same as the air interface destination address (e.g. MSID) AND the OriginalDestinationAddress is supported by the underlying air interface:

1-1 Include the OriginalDestinationAddress parameter set to the original destination address.

2 ENDIF.

3 Include the OriginalOriginatingAddress parameter set to the original originating address.

4 IF the OriginalDestinationSubaddress is supplied:

4-1 Include the OriginalDestinationSubaddress parameter.

5 ENDIF.

6 IF the OriginalOriginationSubaddress is supplied:

6-1 Include the OriginalOriginationSubaddress parameter.

7 ENDIF.

8 Set the TeleserviceIdentifier to the SMS_TeleserviceIdentifier parameter.

9 Set the BearerData to the SMS_BearerData parameter.

10 IF radio contact cannot be established with the MS:

10-1 Include the SMS_CauseCode parameter indicating *No response to page.*

10-2 Return to calling task.

11 ENDIF.

12 Send a SMD-REQUEST toward the indicated MS.

13 Start the Short Message Air Delivery Timer (SADT).

14 WAIT for an SMD-ACK or SMD-NAK response:

15 WHEN an SMD-ACK is received:

15-1 Stop the timer (SADT).

15-2 IF the message can be processed:

15-2-1 Relay all received parameters.

15-3 ELSE (the message cannot be processed):

15-3-1 Include SMS_CauseCode parameter indicating *Other radio interface problem.*

15-4 ENDIF.

16 WHEN an SMD-NAK is received:

16-1 Stop the timer (SADT).

16-2 IF the message can be processed:

16-2-1 Relay all received parameters.

16-3 ELSE (the message cannot be processed):

16-3-1 Include SMS_CauseCode parameter indicating *Other radio interface problem.*

16-4 ENDIF.

17 WHEN the timer (SADT) expires:

17-1 SMS_CauseCode: *No acknowledgment:* SMD-Request

17-2 Include the SMS_CauseCode parameter indicating *No acknowledgment.*

18 ENDWAIT.

19 Return to the calling task with the included or relayed parameters.

4.2 MS-Based SME Receiving an SMD-REQUEST

Upon receipt of an air interface SMD-REQUEST directed to the mobile, the MS-based SME shall do the following:

1 IF the OriginalDestinationAddress parameter is received:

1-1 Set the original destination address with the address in the received OriginalDestinationAddress parameter.

2 ELSE:

2-1 Set the original destination address with the MSID.

3 ENDIF.

3-1 Set the original originating address with the address in the received OriginalOriginatingAddress parameter.

4 IF the OriginalDestinationSubaddress is received:

4-1 Set the original destination subaddress to the OriginalDestinationSubaddress.

5 ENDIF.

5-1 IF the OriginalOriginationSubaddress is supplied:

5-1-1 Set the original origination subaddress to the OriginalOriginationSubaddress.

5-2 ENDIF.

5-3 IF the teleservice indicated by the TeleserviceIdentifier is unknown or is not supported:

5-3-1 Include the CauseCode parameter appropriately, e.g. indicating *Invalid Teleservice ID*.

5-3-2 Send a negative acknowledgment (SMD-NAK).

5-4 ELSEIF the MS has a resource shortage:

5-4-1 Include the CauseCode parameter appropriately, e.g. indicating *Destination resource shortage*.

5-4-2 Send a negative acknowledgment (SMD-NAK).

5-5 ELSE:

5-5-1 Process the message with the indicated teleservice.

5-5-2 Send a positive acknowledgment (SMD-ACK) or negative acknowledgment (SMD-NAK) as requested by the teleservice.

5-6 ENDIF.

6 ENDIF.

4.3 MSC Receiving an Unexpected SMD-ACK or SMD-NAK

Upon receipt of an unexpected air interface SMD-ACK or SMD-NAK, the Serving MSC shall do the following:

1 Record or report the error according to internal procedures.

2 Discard the message.

4.4 MS-Based SME Initiating SMD-REQUEST Toward an MSC

Upon request to send an SMS point-to-point message across the air interface, the MS-based SME shall do the following:

1 IF the destination address is known:

1-1 Include the DestinationAddress parameter set to that address.

2 ENDIF.

3 Include the OriginalDestinationAddress parameter set to the original destination address.

4 IF the original originating address is different than the MSID of this MS (e.g., more than one address is supported by the MS or address type other than MSID needs to be specified):

4-1 Include the OriginalOriginatingAddress parameter set to the original originating address.

5 ENDIF.

6 IF the OriginalDestinationSubaddress is supplied:

6-1 Include the OriginalDestinationSubaddress parameter.

7 ENDIF.

8 IF the OriginalOriginationSubaddress is supplied:

8-1 Include the OriginalOriginationSubaddress parameter.

9 ENDIF.

10 Set the TransactionID parameter to the SMS transaction identification number.

11 Send a SMD-REQUEST toward the Serving MSC.

12 Start the Short Message Air Origination Timer (SAOT).

13 WAIT for an SMD-ACK or SMD-NAK response:

14 WHEN an SMD-ACK is received:

14-1 Stop the timer (SAOT).

14-2 IF the message is well formed:

14-2-1 Relay all received parameters.

14-3 ELSE:

14-3-1 Include CauseCode parameter appropriately, e.g. indicating *Other radio interface problem.*

14-4 ENDIF.

15 WHEN an SMD-NAK is received:

15-1 Stop the timer (SAOT).

15-2 IF the message is well formed:

15-2-1 Relay all received parameters.

15-3 ELSE:

15-3-1 Include CauseCode parameter indicating *Other radio interface problem.*

15-4 ENDIF.

16 WHEN the timer (SAOT) expires:

16-1 Include CauseCode parameter appropriately, e.g. indicating *Other radio interface problem.*

17 ENDWAIT.

18 Return to the calling task with the included or relayed parameters.

4.5 Serving MSC Receiving an SMD-REQUEST

Upon receipt of an air interface SMD-REQUEST from an MS-based SME, the Serving MSC shall do the following:

1 IF the DestinationAddress parameter is received:

1-1 Set the destination address with the address in the received DestinationAddress parameter.

2 ENDIF.

3 Set the original destination address with the address in the received OriginalDestinationAddress parameter.

4 Set the originating address to the network address of the originating MSC.

5 IF the OriginalOriginatingAddress parameter is received AND is not equal to the MIN or IMSI of the originating MS:

5-1 Set the original originating address with the address in the received OriginalOriginatingAddress parameter.

6 ELSE:

6-1 Set the original originating address with the MDN of the originating MS.

7 ENDIF.

8 IF the OriginalDestinatingSubaddress parameter is received:

8-1 Set the original destinating subaddress to the OriginalDestinatingSubaddress parameter.

9 ENDIF.

10 IF the OriginalOriginationSubaddress parameter is supplied:

10-1 Set the original origination subaddress to the OriginalOriginationSubaddress parameter.

11 ENDIF.

12 Set the transaction identification to the TransactionID parameter.

13 IF the MSC is the Anchor MSC for the indicated MS:

13-1 Execute the “Anchor MSC Initiating SMS Delivery Point-To-Point” task (see Part 641, sec. 3.5).

14 ELSE (the MSC is the Serving MSC):

14-1 Include the InterMSCCircuitID parameter set to the trunk used in the direction toward the Anchor MSC.

14-2 Execute the “MSC Initiating SMS Delivery Backward” task (see Part 641, sec. 1.1).

15 ENDIF.

16 (Get here after the message has been relayed and responded to.)

17 IF the MS is still being served:

17-1 IF the request was *accepted*:

17-1-1 Relay the indicated BearerData.

17-1-2 Send an SMD-ACK to the MS based SME.

17-2 ELSE (the request was *denied*):

17-2-1 Relay the indicated CauseCode.

17-2-2 Send an SMD-NAK to the MS based SME.

17-3 ENDIF.

18 ELSE (the MS is no longer being served):

18-1 Discard the message.

19 ENDIF.

20 Return to calling task .

4.6 MS-Based SME Receiving an Unexpected SMD-ACK or SMD-NAK

Upon receipt of an unexpected Air Interface SMD-ACK or SMD-NAK, the MS-based SME shall do the following:

1 Record or report the error according to internal procedures.

2 Discard the message.

5 ANNEX E: SIGNAL STRENGTH ARBITRATION

This annex is informative and is not considered part of this Standard.

5.1 RegistrationCancellation Race Condition Algorithm

An MS may re-register before the expected next registration time as a result of any one of the following conditions:

1. The RegistrationCancellation Race Condition problem occurs.
2. An MSC within the domain of a VLR overhears a registration or origination which occurs within the domain of a neighboring VLR (i.e., an MS registers on a new Serving MSC outside of the domain of the previous serving VLR and a cochannel co-DCC control channel within the domain of the previous serving VLR overhears the registration).
3. The MS is executing a power-up autonomous registration (e.g., a *TDMA* compliant MS) when the MS is already considered to be in the active state. This implies that either power-down autonomous registration has not been enabled or that the Serving MSC did not receive the power-down registration or power-down release order sent by the MS.
4. The MS did not hear the registration acknowledge message which the Serving MSC sent in response to the MS's autonomous registration, resulting in an MS entering the "Local Recovery Procedures" task and re-executing a registration access after a short delay.

The VLR may determine that the RegistrationCancellation Race Condition has occurred if all of the following conditions are true:

- The MS is in an active state when the RegistrationNotification is received. This activity status check is important because some MSs may execute an autonomous registration each time they power up (e.g., a *TDMA* MS with $PUREG_s = 1$)

AND

- The SystemAccessType parameter indicates that the current RegistrationNotification was sent to the VLR as a result of an autonomous registration which is not a power-down registration and the BorderCellAccess parameter indicates that the registration occurred on a border cell

AND

- The current RegistrationNotification is premature

AND

- The MSCID parameter of the current RegistrationNotification is the same as the one contained in the previous RegistrationNotification INVOKE.

Regardless of whether or not a VLR entry exists for the MS, the VLR will send a RegistrationNotification INVOKE to the HLR for all premature autonomous registrations which occur on border cells. If authentication is active, the VLR will treat the RegistrationNotification as if it were the initial registration within the domain of the VLR and authenticate the MS using the procedures outlined in this Standard or *TSB51*.

If scenario 2 occurs on a border cell, the fact that the VLR sends the HLR a RegistrationNotification will enhance the HLR's ability to determine which of the MSCs which detected the MS's registration is the true serving system. If scenarios 3 or 4 occur on a border cell, the result will simply be a redundant transmission of a RegistrationNotification INVOKE to the HLR.

5.2 Timer Method to Detect Duplicate RegistrationNotification

The HLR will initiate a timer (typically 3 seconds) upon the arrival of a RegistrationNotification INVOKE if the timer is not already running due to the arrival of a previous RegistrationNotification message within the timer period. Upon arrival of the first RegistrationNotification INVOKE, the HLR will also copy the old serving system information into a temporary scratch pad buffer and copy the parameters received in the RegistrationNotification INVOKE into the HLR.

If another RegistrationNotification arrives while the timer is active, the HLR will compare the ReceivedSignalQuality and ControlChannelData parameters of the current RegistrationNotification message with the values stored in the HLR database. If the channel data indicates the same control channel the HLR will perform a signal quality comparison, if possible. If the ReceivedSignalQuality of the current RegistrationNotification is greater than that stored in the HLR, the VLR identification stored in the HLR will be sent a RegistrationNotification RETURN RESULT containing a new AuthorizationDenied and ControlChannelData parameters. The parameters reported by the current RegistrationNotification will then be stored in the HLR.

If the ReceivedSignalQuality of the current RegistrationNotification is less than the value stored in the HLR, the VLR which sent the current RegistrationNotification INVOKE will be sent a RegistrationNotification RETURN RESULT with the AuthorizationDenied, ReceivedSignalQuality and ControlChannelData values stored in the HLR.

When the RegistrationNotification timer expires, a RegistrationCancellation INVOKE is sent to the old serving VLR (identified in the scratch pad buffer) with the ReceivedSignalQuality value, if stored in the HLR, as an INVOKE parameter. If a RegistrationCancellation RETURN RESULT response without the CancellationDenied parameter is received back from the old serving VLR, the HLR will send a RegistrationNotification RETURN RESULT response to the VLR identification stored in the HLR. If a RegistrationCancellation RETURN RESULT with the CancellationDenied parameter is received from the old serving VLR, the HLR will send a RegistrationNotification RETURN RESULT with the AuthorizationDenied parameter set to the VLR identification stored in the HLR. The ReceivedSignalQuality value reported in the RegistrationCancellation RETURN RESULT and ControlChannelData will also be included. The old serving system information stored in the scratch pad buffer will then be copied back into the HLR database

In order to better demonstrate the system’s behavior, we expand the case to a registration access detected by two neighboring systems (Figure 1).

We assume that the old serving system receives the registration access with the best signal strength. The old serving system will then keep serving the MS:

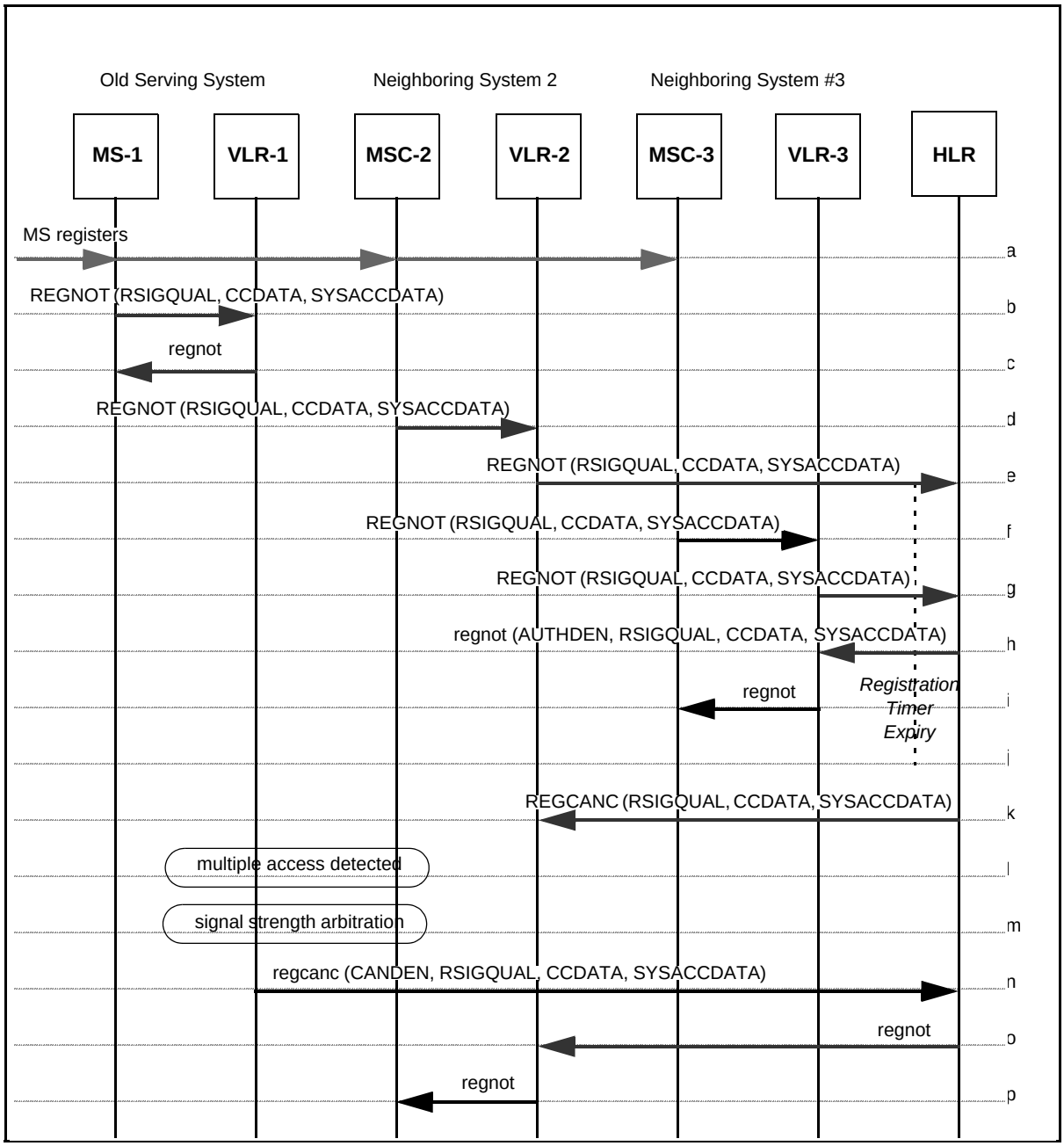

Figure 1 **Multiple registrations received at the HLR (timer method)**

a. MSC-1 (the old serving system) receives a registration access from the MS with a signal strength value of ReceivedSignalQuality.

b. MSC-1 sends a RegistrationNotification INVOKE to VLR-1. Since the MS is already registered in this system, it does not send a RegistrationNotification INVOKE to the HLR. However, it internally marks the registration time, and stores the signal strength value of the registration request.

c. VLR-1 accepts the registration by sending a RegistrationNotification RETURN RESULT to MSC-1.

d. A neighboring system (MSC-2) overhears the same registration access. It sends a RegistrationNotification INVOKE with ReceivedSignalQuality and ControlChannelData to VLR-2.

e. VLR-2 forwards the RegistrationNotification INVOKE to the HLR associated to the MS. The HLR starts a multiple access timer (typically 3 seconds).

f. Another neighboring system (MSC-3) overhears the same registration access. It also sends a RegistrationNotification INVOKE with ReceivedSignalQuality and ControlChannelData to VLR-3.

g. VLR-3 forwards the RegistrationNotification INVOKE to the HLR associated to the MS.

h. While the multiple access timer is running, the ReceivedSignalQuality received with all RegistrationNotifications will be compared. The best will be kept in the HLR, the other will be sent a RegistrationNotification RETURN RESULT with the AuthorizationDenied parameter and ControlChannelData to indicate that their access was invalid.
In our example the HLR determines that system #3 received signal strength is lower than system #2. It sends a RegistrationNotification RETURN RESULT with AuthorizationDenied, ReceivedSignalQuality and, ControlChannelData parameters to system #3.

i. VLR-3 forwards the RegistrationNotification RETURN RESULT to MSC-3.

j. When the multiple access timer expires,…

k. … the HLR sends a RegistrationCancellation INVOKE with the best received ReceivedSignalQuality and ControlChannelData parameters to system 1, the old serving system.

l. The old serving system, by looking at the time stamp of the last received RegistrationNotification, determines that it received the same access.

m. It compares MSC-2's signal strength received in the cancellation request from the HLR to the signal strength received in the last registration (received in step a). If the signal strength received from system #2 is the best, system #1 accepts the RegistrationCancellation.

n. In our example the old serving system receives the best signal strength. It sends to the HLR a RegistrationCancellation RETURN RESULT with the CancellationDenied, ReceivedSignalQuality and ControlChannelData parameters to indicate that it denies the cancellation request.

o. The HLR sends a RegistrationNotification RETURN RESULT with AuthorizationDenied, ReceivedSignalQuality and ControlChannelData to system #2 to indicate that it denies its registration.

p. VLR-2 forwards the RegistrationNotification RETURN RESULT to MSC-2.

5.3 Time Stamp Method to Detect Duplicate RegistrationNotification

The HLR will time stamp the arrival of all RegistrationNotification INVOKE messages. Whenever a RegistrationNotification message is received, the HLR will compare the time stamp of the RegistrationNotification with the time stamp of the previous RegistrationNotification INVOKE. If the time difference is greater than 2 seconds, the HLR will assume that the MS has registered in a new serving system. The HLR will store the RegistrationNotification parameters in the HLR database and send a RegistrationCancellation INVOKE to the old serving VLR.

If the time difference between the current and previous RegistrationNotification is less than 2 seconds, the HLR will use the ReceivedSignalQuality parameter, if available, to determine whether the current or previous RegistrationNotification INVOKE came from the true serving VLR. If the ReceivedSignalQuality parameter of the current RegistrationNotification is greater than the value stored in the HLR database, the HLR will send a RegistrationCancellation INVOKE to the old serving VLR and store the current RegistrationNotification parameters in the HLR database. Whenever a RegistrationCancellation INVOKE is sent to the old serving VLR, the ReceivedSignalQuality value and ControlChannelData reported by the current RegistrationNotification message will be sent as RegistrationCancellation INVOKE parameters. If the ReceivedSignalQuality parameter stored in the HLR is greater than that of the current RegistrationNotification the HLR will send a RegistrationNotification RETURN RESULT with the AuthorizationDenied parameter set to the VLR which sent the current RegistrationNotification.

Since a VLR may reply to a RegistrationCancellation message with a CancellationDenied parameter, it is important for the HLR to save a copy of the old serving VLR information in a scratch pad buffer before overwriting the HLR with the current RegistrationNotification parameters so that the old serving VLR information can be restored into the HLR database if necessary. If the HLR receives a RegistrationCancellation RETURN RESULT with the CancellationDenied parameter, the HLR will store the ReceivedSignalQuality, if available, reported by the RETURN RESULT.

In order to better demonstrate the system's behavior, we expand the case to a registration access detected by two neighboring systems. See Figure 2.

Assuming that neither neighboring system receives the MS with the best signal strength, the old serving system will continue to serve the MS:

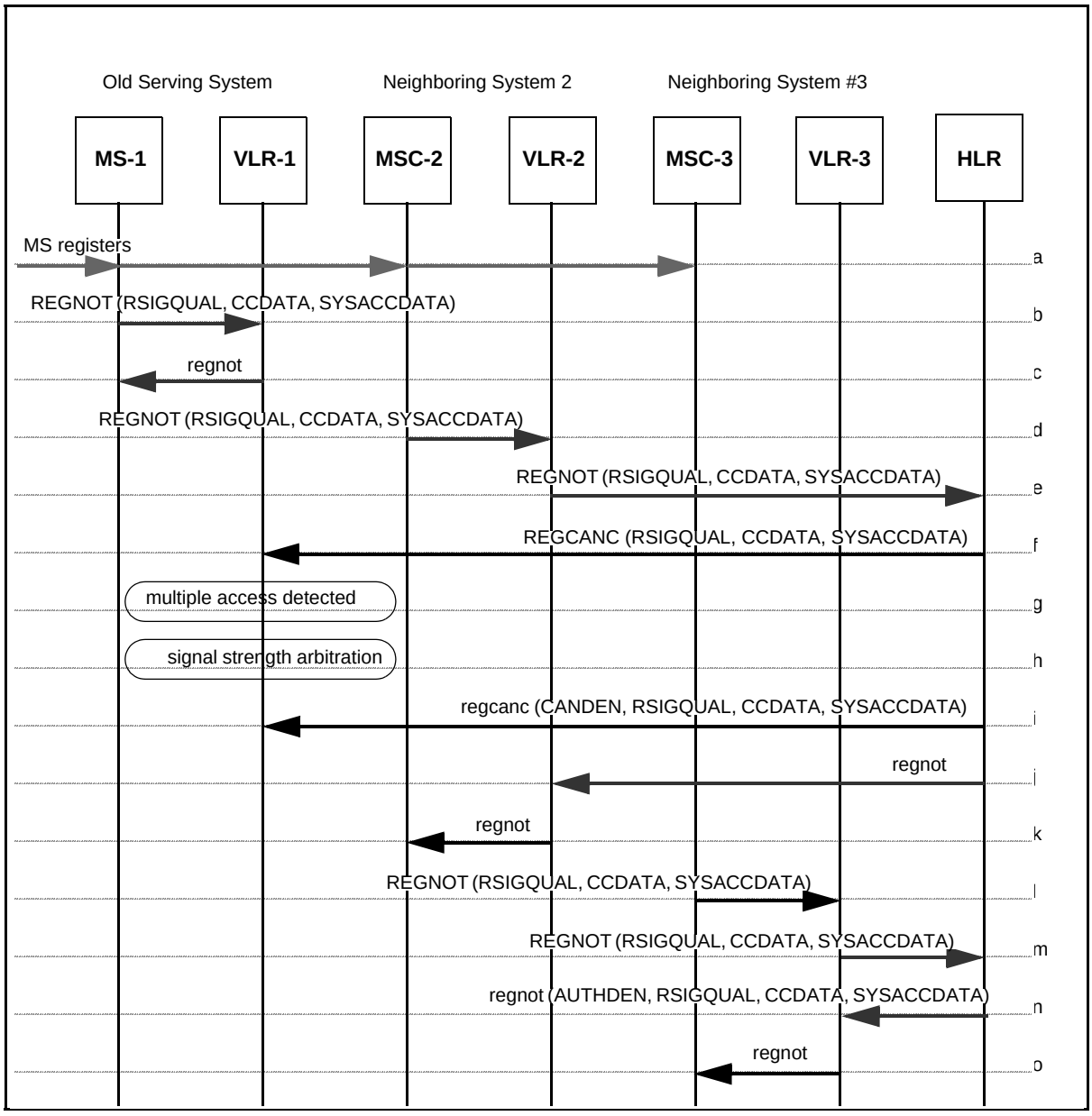

Figure 2 **Multiple registrations received at the HLR (time stamp method)**

a. MSC-1 (the old serving system) receives a registration access from the MS with a signal strength value of ReceivedSignalQuality.

b. MSC-1 sends a RegistrationNotification INVOKE to VLR-1. Since the MS is already registered in this system, it does not send a RegistrationNotification INVOKE to the HLR. However, it internally marks the registration time, and stores the signal strength value of the registration request.

c. VLR-1 accepts the registration by sending a RegistrationNotification RETURN RESULT to MSC-1.

d. A neighboring system (MSC-2) overhears the same registration access. It sends a RegistrationNotification INVOKE with ReceivedSignalQuality and ControlChannelData to VLR-2.

e. VLR-2 forwards the RegistrationNotification INVOKE to the HLR associated to the MS. The HLR marks the time of reception.

f. The HLR sends a RegistrationCancellation INVOKE with the ReceivedSignalQuality and ControlChannelData parameters to the old serving system VLR-1.

g. The old serving system determines that it is a multiple access.

h. VLR-1 determines that the reported ReceivedSignalQuality is not better than the one received from MSC-1.

i. It denies the cancellation, by sending in the RegistrationCancellation RETURN RESULT to the HLR the CancellationDenied, ReceivedSignalQuality and ControlChannelData parameters it received from MSC-1.

j. The HLR updates its stored values of ReceivedSignalQuality and denies the registration from system #2 by sending a RegistrationNotification RETURN RESULT that includes the AuthorizationDenied, ReceivedSignalQuality and ControlChannelData parameters from system #1.

k. VLR-2 forwards the RegistrationNotification RETURN RESULT to MSC-1.

l. A neighboring system (MSC-3) overhears the same registration access. It sends a RegistrationNotification INVOKE with ReceivedSignalQuality and ControlChannelData to VLR-3.

m. VLR-3 forwards the RegistrationNotification INVOKE to the HLR.

n. If it is received within 2 seconds of the last registration, the HLR compares the reported ReceivedSignalQuality with the value stored in its data base. Since the signal strength reported by system #1 is stronger than the one of system #3, the HLR denies the registration by sending a RegistrationNotification RETURN RESULT with the AuthorizationDenied, ReceivedSignalQuality and ControlChannelData parameters from system #1.

o. VLR-3 forwards the RegistrationNotification RETURN RESULT to MSC-3.