

The ATM Forum

Technical Committee

Security Services Renegotiation Addendum to Security Version 1.1

af-sec-0180.000

March, 2002

© 2002 by The ATM Forum. This specification/document may be reproduced and distributed in whole, but (except as provided in the next sentence) not in part, for internal and informational use only and not for commercial distribution. Notwithstanding the foregoing sentence, any protocol implementation conformance statements (PICS) or implementation conformance statements (ICS) contained in this specification/document may be separately reproduced and distributed provided that it is reproduced and distributed in whole, but not in part, for uses other than commercial distribution. All other rights reserved. Except as expressly stated in this notice, no part of this specification/document may be reproduced or transmitted in any form or by any means, or stored in any information storage and retrieval system, without the prior written permission of The ATM Forum.

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum Worldwide Headquarters 572 B Ruger Street San Francisco, CA 94129-0920 Tel: +1.415.561.6110 Fax: +1.415.561.6120

5 Support Services

This section describes the mechanisms that are required to support the security services described in Section 3 and Section 4. Specifically, the following support services are addressed in this section:

- Security message exchange protocols and basic negotiation,
- Security messaging in the control plane,
- Security messaging in the user plane,
- Security messaging in the management plane,
- Key exchange,
- Session key update,
- Certificates.

Security message exchange protocols that support the entity authentication and negotiation services that are described in Section 3 are summarized here and described in detail in Section 5.1.

The three-way security message exchange protocol described in Section 5.1.1.1 may be used for establishing security associations for a point-to-point connection as well as for the first leaf in a point-to-multipoint connection. This protocol is used for security associations that require negotiation of security options. The three-way exchange has the advantage that it does not use time stamps, and therefore does not require clock synchronization.

The two-way security message exchange protocol described in Section 5.1.1.2 may also be used for establishing a security association for a point-to-point connection or a point-to-multipoint connection. This protocol is used for security associations that do not require negotiation of security parameters, and for adding leaves to multipoint connections. A disadvantage of the two-way exchange protocol is that it requires clock synchronization between the party that generates the security information and the party that validates the security information.

This specification defines two-three mechanisms for transporting security information. These are the signaling-based security message exchange mechanism (described in Section 5.1.4), and the in-band security message exchange mechanism (described in Section 5.1.5), and the management-based security message exchange mechanism (described in Section 5.1.5). In both-all cases, the Security Services Information Element (described in Section 5.1.3) is used to carry the security information.

The method for performing the two-way exchange protocol in security-enhanced signaling [4] [5] [6] flows is described in detail in Section 5.1.4. (The three-way message exchange protocol is not supported in signaling in this specification.)

For point-to-multipoint connections after the first leaf is established, subsequent leaves are added with a two-way security message exchange. This is consistent with the fact that negotiation of security options may be performed only when establishing the first leaf—subsequent leaves must accept the options that the root and the first leaf agreed upon.

The method for performing the three-way message exchange protocols in the user plane VCC/VPC is described in detail in Section 5.1.5. This method applies to SVCs, PVCs, and permanent virtual path connections. In order to provide a reliable transport service for in-band message flows, an in-band message exchange protocol is defined in Section 5.1.5.3. As with the signaling-based approach, this protocol uses the Security Services Information Element to convey security-related parameters.

The method for performing the two-way and three-way message exchange protocols in the management plane is described in detail in Section 5.5. This method also applies to SVCs and PVCs, and requires a specific cell-loss recovery protocol which is also presented in Section 5.5.

PVCs (permanent virtual circuits) are provisioned connections. Security services negotiation, authentication, certificate exchange, and key exchange can be done via provisioning at the time PVCs are established, or in-band as described in Sections 5.1.5 and 5.5. Once security services for PVCs are established, the data confidentiality and data integrity services for PVCs are provided the same way as they are provided for SVCs. Likewise, session key update (for data confidentiality and data integrity services) for PVCs is done the same way as it is done for SVCs.

When a SVC or PVC is established, a shared master key and initial session keys may need to be established. To prevent active attacks, key exchange must be bound to strong authentication between the SAs, as explained in Sections 5.1.1 and 5.2.

Once a shared master key and initial session keys are exchanged, there is no need for security message exchanges in the middle of connection after the connection is established. However, session keys for these services may have to be changed periodically. Section 5.3 describes the key update mechanism (for the data confidentiality and data integrity services), which uses OAM cells to perform this function.

During an established connection, a mechanism is provided to allow renegotiation of security services using the security message exchanges. Section 5.5 describes the renegotiation mechanism. The security message exchanges are performed using OAM cells and are referenced later as the Management-Based Security Message Exchange mechanism.

Section 5.4 describes the certification infrastructure and mechanisms for transporting certificates. These certificates can be exchanged during the three-way security message exchange protocol or through some other means that is outside the scope of this specification (e.g., directory servers).

5.1.7.2.1 Non-Real-Time Security OAM Cell Formats

Non-Real-Time (NRT) security OAM cells are defined as having an OAM function type of 0001 (binary). Up to 16 NRT cell types are possible, as defined by the Function ID field. The code points for the Function ID field for NRT security cells are defined in Table 10.

Function ID (binary)	Security Function
0001	Data Confidentiality Session Key Exchange (SKE)
0010	Data Integrity Session Key Exchange (SKE)
<u>0011</u>	Acknowledgment
<u>0100</u>	Negotiation
all others	not defined

	Table 10:	Function	ID Code	Points for	Non-Real	-Time	Security	OAM	Cells.
--	-----------	----------	----------------	-------------------	----------	-------	----------	-----	--------

5.1.7.2.1.3 Negotiation OAM Cell Format

The format of the Negotiation OAM Cell is defined in Figure A.

Bits								
<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	<u>Octets</u>
	<u>GFC/V</u>	PI [11:8]			<u>VPI</u>	[7:4]		<u>1</u>
	VPI	[3:0]			<u>VCI [</u>	<u>15:12]</u>		<u>2</u>
			VCI	[11:4]			_	<u>3</u>
	VCI	<u>[3:0]</u>			<u>PTI</u>		<u>CLP=0</u>	<u>4</u>
<u>HEC [7:0]</u>						<u>5</u>		
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>Function Type = 0001</u>			<u>6</u>	
Relative IDFunction ID = 0100						<u>7</u>		
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	Flow Number				<u>8</u>
SSIE Fragment						<u>9-48</u>		
Sequence Number						<u>49</u>		
Reserved						<u>50-51</u>		
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	CRC	[9:8]	<u>52</u>
			CRC	[7:0]				<u>53</u>

Figure A: Negotiation OAM Cell Format

Notes:

- 1) The use of the Relative ID field is defined in Section 5.1.7.3.
- 2) The Flow Number field is used to identify the flow number to which the negotiation OAM cell belongs. Values taken are 0 (for the first flow), 1 (for the second flow), and 2 (for the third flow).
- 3) The Sequence Number field contains the number of the SSIE fragment belonging to the same SME flow. The first SSIE fragment is assigned 0; the second is assigned 1, and so forth.
- 4) <u>The SSIE Fragment field contains one of the 40-byte fragments composing the SSIE.</u>
- 5) The reserved bytes after the Sequence Number field are provided so that the SSIE fragment field is aligned on a 32 bit boundary, to simplify high speed implementations.

5.1.7.2.1.4 Acknowledgment OAM Cell Format

The format of the Acknowledgment OAM Cell is defined in Figure B.

Bits								
<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	<u>Octets</u>
	<u>GFC/VI</u>	<u>PI [11:8]</u>			VPI	[7:4]		<u>1</u>
	<u>VPI</u>	[3:0]			<u>VCI [</u>	<u>15:12]</u>		<u>2</u>
			VCI	[11:4]				<u>3</u>
	<u>VCI</u>	<u>[3:0]</u>			<u>PTI</u>		<u>CLP=0</u>	<u>4</u>
<u>HEC [7:0]</u>						<u>5</u>		
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>Function Type = 0001</u>			<u>6</u>	
Relative IDFunction ID = 0011						<u>7</u>		
<u>0</u>	<u>0</u> <u>0</u> <u>0</u> <u>Flow Number</u>				<u>8</u>			
Sequence Number						<u>9</u>		
Reserved RSAC MEC						<u>10</u>		
Reserved						<u>11-51</u>		
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	CRC	[9:8]	<u>52</u>
			CRC	[7:0]				<u>53</u>

Figure B: Acknowledgment OAM Cell Format

Notes:

- 1) The use of the Relative ID field is defined in Section 5.1.7.3.
- 2) The Flow Number field is used to identify the flow number to which the acknowledgment refers. Values taken are 0 (for the first flow), 1 (for the second flow), and 2 (for the third flow).
- 3) <u>The Sequence Number field contains the sequence number of the next-expected negotiation OAM cell.</u>
- 4) <u>The RSAC (Ready for Security Association Changeover) bit is set to 1 to indicate that the security association changeover, if required, can start. This guarantees that the last flow has been processed correctly.</u>
- 5) <u>The MEC (Message Exchange Complete) bit indicates either that no other flow should be expected</u> (MEC=1) or that another flow will follow (MEC= 0).

5.1.7.2.2 Real Time Security OAM Cell Formats

Real Time (RT) security OAM cells are defined as having an OAM function type of 0010 (binary). Up to 16 RT cell types are possible, as defined by the Function ID field. The code points for the Function ID field for RT security cells are defined in Table 11.

	Table 11: Function ID Code Foints for Kear Time Security OAM Cens.							
Function ID	Security Function							
(binary)								
0001	Data Confidentiality Session Key Changeover (SKC)							
0010	Data Integrity Session Key Changeover (SKC)							
<u>0011</u>	Security Association Changeover (SAC)							
all others	not defined							

Table 11: Function ID Code Points for Real Time Security OAM Cells.

5.1.7.2.2.3 Security Association Changeover OAM Cell Format

			B	its				
<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	2	<u>1</u>	Octets
	<u>GFC/VI</u>	<u>PI [11:8]</u>			<u>VPI</u>	[7:4]		<u>1</u>
	VPI	<u>[3:0]</u>			VCI [15:12]		<u>2</u>
<u>VCI [11:4]</u>						<u>3</u>		
	<u>VCI</u>	[3:0]			<u>PTI</u>		<u>CLP=0</u>	<u>4</u>
HEC [7:0]						<u>5</u>		
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>]	Function T	ype = 001	<u>0</u>	<u>6</u>
Relative IDFunction ID = 0011						<u>7</u>		
Bank ID Reserved					<u>8</u>			
Reserved						<u>9-51</u>		
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	CRC	[9:8]	<u>52</u>
			CRC	[7:0]				<u>53</u>

The format of the Security Association Changeover OAM Cell is defined in Figure C.

Figure C: Security Association Changeover (SAC) OAM Cell Format

Notes:

- 1) The use of the Relative ID field is defined in Section 5.1.7.3.
- 2) <u>The Bank ID field is an alternating pattern of 0 hex or F hex, for successive security association changeovers.</u>
- 3) <u>Reserved bytes are set to 6A hex, reserved bit fields less than 1 byte in length are set to all zeros.</u>
- 4) <u>The reserved bits included after the 4-bit Bank ID field are provided so that the CKN and IKN fields</u> are aligned on 16 bit boundaries, to simplify high speed implementations.

5.5 Management-Based Security Message Exchange

When the initiator (or responder) wants to modify the security association used to protect its ATM traffic exchanges, it sends negotiation OAM cells within the user data stream to negotiate new security parameters with the remote partner. Up to three SME flows, which are carried by OAM cells, are allowed between partners to negotiate a new security association. Each negotiation OAM cell reception is acknowledged by the receiving partner, which sends back acknowledgment OAM cells.

The negotiation OAM cells and acknowledgment cells belong to end-to-end F4 flows for VPCs and end-toend F5 flows for VCCs, respectively. Their formats are given in Sections 5.1.7.2.1.3 and 5.1.7.2.1.4.

Once the negotiation cell exchanges are done and the negotiation is over, each partner indicates to the other when to start using the new security association by sending Security Association Changeover (SAC) cells whose format is given in Section 5.1.7.2.2.3.

In order to change a security association at high speeds without disrupting service to the end user, two security associations are required: a current-association and a next-association. Parameters for the next-association are delivered using the negotiation cells. The receiver of negotiation cells stores the next-

association in a separate memory location until needed. The actual changeover occurs when the SAC cell is received.

Either peer SA can initiate security renegotiation. If an SA receives FLOW-1 renegotiation cells before it sends FLOW-1 renegotiation cells, then it is the responding SA.

If a security agent receives FLOW-1 renegotiation OAM cells after it has sent FLOW-1 renegotiation OAM cells, the peer SAs determine which is the Initiator as follows:

- 1) If the connection is an SVC, each SA performs an unsigned comparison of its address to its peer SA address, and if its own address is greater than the address of its peer, it terminates renegotiation.
- 2) If the connection is a PVC, each SA waits a randomly chosen duration (according to an exponential distribution with mean of 1 second) before trying a new negotiation.

The security association update assumes that both of the partners exchange negotiation cells. As such, unidirectional connections and point-to-multipoint connections are not supported.

The protocol described in this section is described in further detail in the Finite State Machines in Section 9. If any discrepancies exist between this description and the Finite State Machine description, the Finite State Machine description takes precedence.

5.5.1 The Security Association Renegotiation Process

<u>Negotiation OAM cells are used to transfer security parameters for negotiating the next-association.</u> <u>Renegotiation is performed using either the 2 or 3-way SME protocol described in Section 5.1 and</u> <u>encapsulating the SSIE into the negotiation OAM cells.</u>

The security negotiation OAM cell has 40-bytes available in its SSIE Fragment, so that the SSIE is fragmented into 40-byte blocks before being encapsulated into the negotiation OAM cells. Note that with a 1-byte Sequence Number, the SSIE maximum length is 255*40 = 10,200 bytes.

5.5.1.1 Renegotiation Processing at the Negotiation Update Initiator (First Flow)

When a NUI (Negotiation Update Initiator) needs to update the security association, it constructs a new SSIE with compliance to the 2-way or 3-way SME protocol; it fragments the SSIE of the first flow into 40-byte blocks; it encapsulates them into negotiation OAM cells; and then it sends the negotiation OAM cells.

The format of the negotiation OAM cells is given in Section 5.1.7.2.1.3. The flow number should indicate 0 as the first flow. For instance, if the SSIE is 100 bytes long, the SSIE should be encapsulated into three negotiation OAM cells, carrying 0 in the first negotiation OAM cell sent, 1 in the second negotiation OAM cell, and 2 in the third negotiation OAM cell.

When renegotiation completes (i.e., when SAC cells are sent), any pending session key updates are terminated, and the new master and initial session keys are immediately used for traffic security and subsequent session key updates.

T103 is started when the FLOW-1 SSIE is sent. If T103 expires, the SSIE is sent again. Timer is cancelled when one acknowledgement cell is received (see section 9). (For further details, readers can refer to section 9.)

Note that the old security association remains active until an SAC cell is sent.

5.5.1.2 Renegotiation Processing at the Negotiation Update Responder (First Flow)

<u>Upon receipt of the negotiation OAM cells, the NUR (Negotiation Update Responder) extracts the Flow</u> <u>Number, Sequence Number, and SSIE Fragment, and performs the following steps:</u>

- 1. Verifies the 10-bit OAM cell CRC is correct, and discards the cell if it is not.
- 2. Reassembles the OAM cells to retrieve the first SSIE.
- 3. Verifies the SSIE correctness. This is performed by comparing the SSIE length indicated in the first negotiation OAM cell against the number of OAM cells received. If the SSIE is not fully received and the timer of the NUR (T104) expires, the NUR should send acknowledgment cells to cause a new SSIE transmission.
- 4. If 2-way SME, then the SA stores the new decrypting (confidentiality and/or integrity) session key(s) in memory, until the corresponding SAC cell is received.

5.5.1.3 Renegotiation Processing at the Negotiation Update Responder (Second Flow)

The NUR constructs the SSIE of flow 2-2WE or 2-3WE; it fragments the SSIE into 40-byte blocks ; it encapsulates the fragments into the negotiation OAM cells; and it sends them over the network.

T103 is started when the FLOW-2 SSIE is sent. If T103 expires, the SSIE is sent again. Timer is cancelled when one acknowledgement cell is received (see section 9). (For further details, refer to section 9.)

When renegotiation completes (i.e., when SAC cells are received), any pending session key updates are terminated, and the new master and initial session keys are immediately used for traffic security and subsequent session key updates.

5.5.1.4 Renegotiation Processing at the Negotiation Update Initiator (Second Flow)

<u>Upon receipt of the negotiation OAM cells, the NUI extracts the Flow Number, Sequence Number, and SSIE Fragment, and performs the following steps:</u>

- 1. Verifies the 10-bit OAM cell CRC is correct, and discards the cell if it is not.
- 2. Verifies the SSIE correctness. That consists in comparing the SSIE length indicated in the first negotiation OAM cell against the number of OAM cells received. If the SSIE is not fully received and the timer of the NUI (T104) expires, the NUI should send some acknowledgment cells to cause a new SSIE transmission.
- 3. Reassembles the OAM cells to retrieve the second SSIE.
- 4. Stores the new decrypting (confidentiality and/or integrity) session key(s) in memory, until the corresponding SAC cell is received.
- 5. If a 2-way SME protocol is selected, and the SSIE is fully received, a group of acknowledgment cells with MEC=1 and RSAC=0 are sent. Then when the SSIE is fully processed, and the decrypting key(s) are decrypted, a group of acknowledgment OAM cells with MEC=1 and RSAC=1 are sent.

5.5.1.5 Renegotiation Processing at the Negotiation Update Initiator (Third Flow)

If the 3-way SME protocol is selected, the NUI constructs an SSIE; it fragments the SSIE into 40-byte blocks and encapsulates the fragments into the negotiation cells.

5.5.1.6 Renegotiation Processing at the Negotiation Update Responder (Third Flow)

Upon receipt of the negotiation OAM cells, the NUR extracts the Flow Number, Sequence Number, and SSIE Fragment, and performs the following steps:

- 1. Verifies the 10-bit OAM cell CRC is correct, and discards the cell if it is not.
- 2. Verifies the SSIE correctness. That consists in comparing the SSIE length indicated in the first negotiation OAM cell against the number of OAM cells received. If the SSIE is not fully received and either the timer of the NUR (T104) or the timer of the NUI (T103) expires, the NUI should send some acknowledgment cells to cause a new SSIE transmission.
- 3. Reassembles the OAM cells to retrieve the third SSIE.
- 4. Stores the new decrypting (confidentiality and/or integrity) session key(s) in memory, until the corresponding SAC cell is received.
- 5. If the SSIE is fully received, a group of acknowledgment cells with MEC=1 and RSAC=0 are sent. Then when the SSIE is fully processed, and the decryption (confidentiality and/or integrity) keys are decrypted, a group of acknowledgment OAM cells with MEC=1 and RSAC =1 are sent.

5.5.1.7 The Acknowledgment Process

Acknowledgment cells are used to acknowledge or negatively acknowledge a group of negotiation OAM cells. That is, they acknowledge all the negotiation OAM cells received with a sequence number smaller than the sequence number it carries. In other words, acknowledgment cells include the sequence number of the next-negotiation OAM cells expected.

The format of the Acknowledgment OAM cell is shown in Section 5.1.7.2.1.4.

5.5.1.7.1 Acknowledgment Processing at the Acknowledgment Cell Sender

The acknowledgment procedure enables a number of negotiation OAM cells to be acknowledged at the same time. Actually acknowledgment OAM cells are sent when one of the following events happens:

- <u>The negotiation OAM cells receiver detects a cell loss because the last negotiation OAM cell received</u> <u>carries a sequence number greater than expected.</u>
- <u>No new negotiation OAM cells have been received within a timeout period (see the timers definition in Section 5.5.3.2) and the SSIE is not fully received.</u>
- <u>The SSIE is fully received so the negotiation OAM cells received should acknowledge all the</u> previously received negotiation OAM cells. For the SSIE of the last flow, this is done by transmitting a group of acknowledgment cells with MEC=1 and RSAC=0.
- <u>The SSIE of the last flow is fully processed. This is realized by sending a group of acknowledgment cells with MEC=1 and RSAC=1.</u>

The loss of acknowledgment cells is recovered when a retry timer expires at one partner. To avoid waiting for one timer to expire, and improve the delay for negotiation, the acknowledgment cell sender shall transmit the acknowledgment cell multiple times, as described in the Session Key Update procedures in Section 5.3.

5.5.1.7.2 Acknowledgment Processing at the Acknowledgment Cell Receiver

When an acknowledgment cell is received, the receiver should perform the following steps:

1. It verifies that the Flow Number included into the acknowledgment cell is the same as the negotiation OAM cells most recently sent and discards the cell if it is not. 2. It checks the Sequence Number in the received acknowledgment cell against the sequence number of the last negotiation OAM cell it sent. If the Acknowledgment Sequence Number is greater than the sequence number of the last negotiation OAM cell sent by 1, the transmission of the SSIE is correct. If it is equal or smaller than the sequence number of the last negotiation OAM cell sent, the receiver sends again all the negotiation cells with the sequence number greater than or equal to the acknowledgment sequence number. If it is greater than the sequence number of the last negotiation OAM cell sent by at least 2, then an error has occurred.

5.5.2 The Security Association Changeover Process

After the negotiation is completed, both partners invoke the security association changeover process to indicate to the other partner when to start using the new association and decryption (confidentiality and/or integrity) session key(s) to process the receiving traffic correctly. The format of the SAC cell shall be as shown in Section 5.1.7.2.2.3.

One partner sends the SAC OAM cell that instructs the other partner to start using the new (confidentiality and/or integrity) session key(s) and the new association on the cells following the SAC OAM cell. The SAC OAM cell is sent multiple times to guarantee receipt at the receiver in the presence of cell loss, as described in the Session Key Update procedures in Section 5.3. The SAC OAM cell is not cryptographically protected (since it does not carry any confidential information).

5.5.2.1 SAC Processing at the SAC Cell Sender

The sender performs the following steps:

- If the sender is the negotiation responder and the 2-way SME protocol is selected, or if the sender is the initiator and the 3-way SME protocol is selected, the sender shall wait until it receives an acknowledgement cell with MEC=1 and RSAC=1 before sending the corresponding SAC cell. This provides the remote partner with sufficient time to complete the SSIE processing.
- 2. If the sender is the negotiation initiator and the 2-way SME protocol is selected, or if the sender is the responder and the 3-way SME protocol is selected, the sender shall send multiple acknowledgement cells (according to the procedure in 5.5.1.7) with MEC=1 and RSAC=1 before sending the corresponding SAC cell.
- 3. The sender injects the SAC OAM cell into the connection undergoing security association changeover in such a manner that the encryption algorithm, the integrity mechanism, and session key(s) exchanged by the negotiation OAM cell process are used on the next user cell associated with that connection. This includes the case when the next user cell on that connection immediately follows the SAC cell.

The SAC OAM cell may get lost and the sender will not be able to detect that (since there is no SAC cell acknowledgment). To improve the probability that the security association changeover is successful, the sender shall transmit the SAC OAM cell multiple times, as described in the Session Key Update procedures in Section 5.3.

In the case of the Integrity service, the SAC cell is repeated three (3) times, and each cell shall be inserted between SDUs.

5.5.2.2 SAC Processing at the SAC Cell Receiver

Upon receipt of an SAC OAM cell, the destination performs the following steps:

- 1. The destination verifies that the 10-bit OAM cell CRC is correct, and discards the cell if it is not.
- 2. Processes the SAC OAM cell on the connection undergoing security association changeover in such a manner that the encryption algorithm, the integrity mechanism, and the session key(s) negotiated by

the negotiation cell are used on the next cell received on that connection. This includes the case when this cell immediately follows the SAC cell.

5.5.3 Protocol Details

5.5.3.1 Timer Definitions

The following is a description of the timers that are used for the renegotiation operation. The values of these timers can be found in Section 5.5.3.2.

- <u>T103: This timer is used by one partner to determine whether it needs to resend the full SSIE. The timer is started when it has sent a full or partial SSIE. The timer is stopped when one acknowledgment cell is received. The timer is restarted if SSIE fragments are sent in response to the acknowledgment. If it has not received acknowledgement cells before the timer expires, then it resends the full SSIE and starts the timer again.</u>
- <u>T104: This timer is used by one partner to determine whether it needs to resend acknowledgement cells. The timer is started when it sends acknowledgement cells. If it has not received a negotiation OAM cell before the timer expires, then it resends acknowledgement cells and starts the timer again. The timer is stopped when a negotiation OAM cell is received.
 </u>
- 3. T105: This timer is used by one partner to avoid waiting indefinitely for the remote processing of one SSIE. This is used to determine whether the connection needs to be released. This timer is started after receipt of a complete SSIE is acknowledged. This timer is stopped when a complete SSIE is received from the other partner. If this timer expires, then the connection is released.

The following variables (retry counters) are used in conjunction with the timers used in the protocol.

- 1. <u>I-SSIE-Retry-Count: This variable is used in conjunction with timer T103 by the Initiator of the protocol and counts the number of times that a full SSIE or fragments of it have been sent. A full or partial SSIE may be sent up to a maximum of I-MAX-SSIE-RETRY times.</u>
- 2. <u>I-Ack -Retry-Count: This variable is used in conjunction with timer T104 by the Initiator in the</u> protocol and counts the number of times that a group of acknowledgement cells has been sent. Groups of acknowledgement cells may be sent up to a maximum of I-MAX-ACK-RETRY times.
- 3. <u>R-SSIE-Retry-Count: This variable is used in conjunction with timer T103 by the Responder of the protocol and counts the number of times that a full SSIE or fragments of it have been sent. A full or partial SSIE may be sent up to a maximum of I-MAX-SSIE-RETRY times.</u>
- 4. <u>R-Ack -Retry-Count: This variable is used in conjunction with timer T104 by the Responder in the</u> protocol and counts the number of times that a group of acknowledgement cells has been sent. Groups of acknowledgement cells may be sent up to a maximum of R-MAX-ACK-RETRY times.

If any retry counter is exceeded, then the connection is released as described in Section 5.5.4 with the cause code=«protocol error, unspecified».

The following is a description of the constants that are used in conjunction with the retry counter variables used in the protocol. The values of these constants can be found in Section 5.5.3.2.

- 1. <u>I-MAX-SSIE-RETRY</u>: This constant indicates the maximum number of times that the Initiator may resend a full SSIE or fragments of it to the Responder.
- 2. <u>I-MAX-ACK-RETRY: This constant indicates the maximum number of times that the Initiator may</u> resend a group of acknowledgement cells to the Responder.
- 3. <u>R-MAX-SSIE-RETRY: This constant indicates the maximum number of times that the Responder may</u> resend a full SSIE or fragments of it to the Initiator.
- 4. <u>R-MAX-ACK-RETRY: This constant indicates the maximum number of times that the Responder may</u> resend a group of acknowledgement cells to the Initiator.

5.5.3.2 Timer Values

The protocols for renegotiation through OAM cells use a number of timers in their procedures. These timers are summarized in the following table:

Table I : Timers for Renegotiation					
<u>Timer Name</u>	Timer Value				
<u>T103</u>	10 seconds				
<u>T104</u>	<u>5 seconds</u>				
<u>T105</u>	<u>30 seconds</u>				

In addition, the renegotiation mechanism uses the following constant definitions:

Table II:	Constant	Values for	Renegotiation
Table II.	Constant	values for	Renegotiation

Constant Name	Constant Value
I MAX SSIE RETRY	<u>4</u>
I_MAX_ACK_RETRY	<u>10</u>
R_MAX_SSIE_RETRY	<u>4</u>
R_MAX_ACK_RETRY	<u>10</u>

5.5.4 Protocol Error Handling

Detailed error handling procedures are implementation dependent. However, capabilities facilitating the orderly treatment of error conditions provided for in this section shall be provided in each implementation.

When the Initiator or Responder detects an error, it enters the failed state, and aborts. All errors are defined to be unrecoverable errors. The protocol aborts upon expiration of the timer or when one counter exceeded its maximum.

For SVC-initiated calls in the failed state, the security agent shall clear the call. Error recovery for PVC initiated calls is implementation specific.

For the SVC approach, a «Cause» information element describes the reason for an error and provides diagnostic information. This information element is carried in the RELEASE message.

The following cause codes are defined.

Management-based Message Exchange Cause Codes					
Number	Meaning				
<u>111</u>	protocol error, unspecified (Note 1)				
<u>100</u>	invalid information element contents				
<u>63</u>	service or option not available				
47	resource unavailable				

<u>Note 1:</u>

An indication of the security protocol failure shall be included in the Diagnostics field of the Cause IE when the cause value = "#111, protocol error, unspecified." This indication shall be the rejection reason "security exception" as defined in [4].

7.2.1 Security Service Declaration

The Security Service Declaration Section provides a minimal description of the security services that are requested or supported by a security agent.

The Security Service Declaration shall only be contained in the SSIEs within a 2-way signaling-based exchange. It shall not be generated in 3-way and shall be ignored if received in 3-way.

The Security Service Declaration Section is employed when communicating a security service request to a proxy security agent, or when declaring security capabilities to a peer SA.

Bits								
8	8 7 6 5 4 3 2 1						Octet(s)	
1	1 0 0 0 1 0 1 0							
Security Service Declaration Identifier							x.1	
Security Service Declaration								
х								x.2

Security Service Declaration (Octet x.2, bits 8-1)

Bit	Meaning in Flow-1 from Initiator SA and in Flow-2 from Responder SA	Meaning in Flow-1 from end system	Meaning in Flow-2 from proxy
1		1	
0	Confidentiality Service not supported	not requested	not provided
1	Confidentiality Service supported	requested	provided
2	Terte and the Complexity and the second state		
0	Integrity Service not supported	not requested	not provided
1	Integrity Service supported	requested	provided
3	Authentication Comics not summaried		n at museri de d
0	Authentication Service not supported	not requested	not provided
1	Authentication Service supported	requested	provided
4	Kan Frahan as Camias not annot a		n at museri de d
0	Key Exchange Service not supported	not requested	not provided
1	Key Exchange Service supported	requested	provided
5	Cartificate Ersten as Service act approximated		n at museri de d
0	Certificate Exchange Service not supported	not requested	not provided
I	Certificate Exchange Service supported	requested	provided
0	Section Very Undete Semice not suggested		n at museri de d
0	Session Key Update Service not supported	not requested	not provided
1	Session Key Opdate Service supported	requested	provided
/	Assass Control Service not supported	not requested	not movided
0	Access Control Service not supported	not requested	
1	Access Control Service supported	requested	provided
ð	Management Daged Security Massage	not requested	not movided
<u>U</u>	<u>Management-based Security Message</u>	<u>not requested</u>	<u>not provided</u>
1	Exchange (renegoliation) not supported	an and a d	المراد المراجع
<u>1</u>	Evolution and the second secon	requested	provided
	Exchange (renegonation) supported		

7.2.2.8 Management-based Security Message Exchange Options

Bits								
<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	Octet(s)
<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	
Management-based Security Message Exchange Mechanism Options Identifier							Identifier	<u>x.1</u>
	Management-based SME Mechanism Options							
<u>x</u>	<u>X</u>	<u>X</u>	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	<u>x</u>	<u>x.2</u>

Management-based Security Message Exchange Mechanism Options (Octet x.2)

8	7	6	5	4	3	2	1	Meaning
0	0	0	0	0	0	0	0	Not supported
0	0	0	0	0	0	0	1	Supports Management-based Security Message
E	ccha	ang	e	(Nc	ote)			
1	0	0	0	0	0	0	1	Requires Management-based Security Message
Ex	ccha	ang	e					

Note: This codepoint applies only when used by the initiator in FLOW1-3WE.

<u>9 Management-based Security Message</u> Exchange Finite State Machines

The Finite State Machines (FSMs) described in this section specify the intended behavior for the in-band Security Association Renegotiation protocol. These FSMs correspond to the textual procedures described in Section 5.5 of this specification. If there are any discrepancies between the textual procedures and the FSM tables, the FSM tables shall take precedence.

The FSMs covers two potential configurations:

- 1. Initiator of security association negotiation,
- 2. <u>Responder to security association negotiation.</u>

The FSMs are described in five sections:

- 1. <u>The FSM Graphical Views are shown in Section 9.1.</u>
- 2. <u>All FSM States are described in section 9.2.</u>
- 3. <u>All FSM Events are described in section 9.3.</u>
- 4. <u>All FSM Actions are described in section 9.4.</u>
- 5. <u>The FSM Summary Tables are shown in section 9.5.</u>

9.1 FSM Graphical View

The notations used are the following:

- <u>SIE1, SIE2 are the SSIE of the first and second flow.</u>
- <u>SIE1FragTotalNb</u>, <u>SIE2FragTotalNb</u> are the number of fragments of <u>SIE1</u> and <u>SIE2</u>, that is the number of negotiation cells necessary to transport the <u>SSIE</u>.
- <u>SIE1FragNb</u>, SIE2FragNb are the numbers indicated in the SIE1 and SIE2 fragment received.
- <u>WrongFragNb is the number of the first unexpected fragment (with a wrong number) belonging to the same SIE.</u>
- SeqNb is the sequence number indicated in the acknowledgement cells.
- AckNb is the sequence number of the last acknowledgement cell received.
- FragNb is the number of the next SSIE fragment to be received.

During initialization in the S1-R Idle state, the R-Ack Retry count counter and the WrongFragNb are assumed to be set to 0.



Figure 6. Initiator FSM.



Figure 7. Responder FSM.

9.2 FSM States

INITIATOR	<u>-1</u>		
Number	<u>Name</u>	Messages	Description
		Outstanding	
<u>S1–I</u>	<u>Idle</u>	None, or	A new security association negotiation is required
		<u>SAC cells +</u>	
		<u>Acknowledge</u>	
		ment cells	
		$\frac{\text{MEC}=1}{\text{PSAC}=1}$	
\$2 I	Waiting for	$\frac{\text{KSAC}=1}{\text{Eull SIE1}}$	Initiator has initiated a security association pagatistion
<u>52-1</u>	<u>Ack</u>	<u>Full SIET</u>	hy sending a SSIE 1 to the responder. It is waiting for
	$\frac{MEK}{MEC=0}$		the Responder to acknowledge the SIF1
	$\frac{MEC=0}{RSAC=0}$		the responder to demovied by the SHST
<u>S3–I</u>	Waiting for	Acknowledge	Responder sends the first fragment of the SIE2
	the 1 st	ment cells	
	fragment of	<u>MEC=0,</u>	
	<u>SIE2</u>	<u>RSAC =0</u>	
<u>S4–I</u>	Waiting for	<u>None</u>	Responder sends the next fragments of the SIE2
	the next		
	<u>SIE2</u>		
85 I	<u>Iraginents</u> Weiting for	Aaknowladga	Initiator waits until the SIE2 is processed locally
<u>33-1</u>	the SIE2	<u>Acknowledge</u>	initiator waits until the STE2 is processed locarry
	local process	MFC-1	
	<u>iocar process</u>	$\frac{MLC=1}{RSAC} = 0$	
S6–I	Failed		Error occurred during the security association
			negotiation.
RESPONDER	<u>-R</u>		
<u>Number</u>	<u>Name</u>	Messages	Description
		<u>Outstanding</u>	
<u>S1-R</u>	<u>ldle</u>	None, or	No requests for a new security association negotiation
C2 D	Waiting for	SAC cells	Deep on dee sucito for the event frequencies of the SIE1 if
<u>52-K</u>	waiting for	INONE	Responder waits for the flext fragments of the SIE1 fl
	fragment		
S3-R	Waiting for	None	Responder waits until the SIE2 is locally constructed
<u></u>	SIE2 local	<u>010</u>	responder mans until the SHS2 is rocarry constructed
	construction		
<u>S4-R</u>	Waiting for	Full SIE2	Responder sends a SIE2 to the initiator. It is waiting for
	Acknowledg		the initiator to acknowledge the SIE2
	ement cells		
	MEC=1		
05 D	<u>RSAC =0</u>	NT.	
<u>85-K</u>	Waiting for	None	Responder waits for the acknowledgement cells which
	ACKNOWIEdg		association changeover
	MEC-1		association changeover
	$\frac{1}{RSAC} - 1$		
S6-R	Failed	None	Error occurred during the security association
~~~~			negotiation.

## 9.3 FSM Events

		Table 5: Events
	<b>INITIATOR</b>	
Number	Name	Description
E1	Security Context Update Request	The initiator has been requested to negotiate new security parameters
E2	Valid Ack	Acknowledgement cells have been received
	(MEC=0, RSAC	• this event is expected
	=0,	
	SeqNb=SIE1Frag	
	TotalNb)	
	Received	
E3	T103 expires	<u>Timer T103 has exceeded time shown in section 5.5.3.1.</u>
		<ul> <li><u>this event results in a new SIE1 transmission</u></li> </ul>
E4	I-SSIE-Retry-	Initiator has sent SSIE of FLOW1 the maximum number of times.
	Count Exceeded	• <u>This event results in a fault at the initiator</u>
E5	Valid Ack	One acknowledgement cell has been received but indicates that cells were
	(MEC=0, RSAC	lost. If a group of similar acknowledgement cells is received, event E5
	$\underline{=0,}$	occurs only once when receiving the first acknowledgement cell.
	SeqNb<=SIE1Fra	• this event results in the new transmission of the SIE1 fragments
	<u>grotaind</u>	numbered from SeqNb
	AND	
	$\frac{AND}{(SeaNb + AckNb)}$	
F6	Valid SIF2 first	The initiator receives the first fragment of the SIF2
LU	fragment	• this event results in new fragments being expected
	Received	• <u>uns event results in new tragments being expected</u>
	(SIE2FragTotalN	
	<u>b&gt;1)</u>	
E7	Valid SIE2	The initiator receives the full SIE2 in one fragment
	received	<ul> <li>this event results in no more fragments being expected</li> </ul>
	(SIE2FragTotalN	
	<u>b=1)</u>	
E8	Valid SIE2 fully	The initiator receives the full SIE2 in SIE2FragTotalNb fragments
	received	• this event is expected if the SIE2 requires more than one
	(FragNb=SIE2Fra	<u>fragment.</u>
EO	<u>g1otalNb)</u>	The T104 has seen a latting the main section 5.5.2.1
E9	<u>1104 expires</u>	<u>11mer 1104 has exceeded time shown in section 5.5.5.1.</u>
		• Inis event results in Acknowledgement cells MEC=0, KSAC =0, SecNb=EregNb 1 or in Acknowledgement cells MEC=0, RSAC
		-0. SeqNb-0
E10	I-Ack Retry	Initiator has sent a group of acknowledgement cells (MEC=0, RSAC =0)
LIU	Count Exceeded	the maximum number of times
		• this event results in a fault at the initiator
E11	One SIE2	The initiator receives one more fragment
2	fragment received	• this event is expected if the SIE2 requires more than one
		fragment.
E12	Local process	The initiator processes the SIE2
	over	• this event is expected.
E13	Unexpected SIE2	The initiator receives one fragment with the wrong sequence number
	fragment received	(implying that at least one cell is lost). If several fragments of the same
	AND	SIE are received, event E13 occurs only once when the first fragment is
	((WrongFragNb=	received out of sequence.

	0)OR(SIE2FragN	• this event results in sending a group of Acknowledgement cells
	<u>b=<wrongfragnb< u=""></wrongfragnb<></u>	<u>MEC=0, RSAC=0, SeqNb=0</u>
E14	<u>))</u> T105 avrinas	Timer T105 has avaaded time shown in section 5.5.2.1
E14	<u>1105 expires</u>	<u>Timer 1105 has exceeded time shown in section 5.5.5.1.</u>
	RESPONDER	• <u>uns event results in a fault at the initiator</u>
Number	Name	Description
E1	Valid SIE1 first	The responder receives the first fragment of the SIE1
11	fragment	this event results in new fragments being expected
	Received	
	(SIE1FragTotalN	
	<u>b&gt;1)</u>	
E2	Valid SIE1	The responder receives the full SIE1 in one fragment
	Received	<ul> <li>this event results in no more fragments being expected</li> </ul>
	(SIE1FragTotalN	
E2	$\frac{D=1}{Valid SIE1 fully}$	The responder receives the full SIE1 in SIE1FreeTotelNh freements
ЕJ	received	this event is expected if the SIE1 requires more than one
	(FragNb=SIE1Fra	fragment
	gTotalNb)	<u>Indgittetta</u>
E4	T104 Expires	Timer T104 has exceeded time shown in section 5.5.3.1.
		• <u>This event results in a fault at the responder</u>
E5	<u>R-Ack Retry</u>	Responder has sent a group of acknowledgement cells (MEC=0, RSAC
	Count Exceeded	=0) the maximum number of times.
		<u>This event results in a fault at the initiator</u>
E6	One SIE1	The responder receives one more fragment
	fragment received	• <u>this event is expected if the SIE1 requires more than one</u>
<b>F7</b>	T 1	<u>fragment.</u>
E/	Local process	<u>The responder processes the SIET</u>
F8	Valid Ack	<u>uns event is expected.</u> Acknowledgement cells have been received
Lo	$\frac{\text{Valid ACK}}{\text{(MEC=1)}}$	this event is expected
	RSAC=0.	
	SeqNb=SIE2Frag	
	TotalNb)	
	Received	
E9	T103 expires	<u>Timer T103 has exceeded time shown in section 5.5.3.1.</u>
		this event results in a new SIE2 transmission
E10	<u>I-SSIE Retry</u>	Initiator has sent SSIE of FLOW2 the maximum number of times.
<b>F11</b>	<u>Count Exceeded</u>	• <u>This event results in a fault at the initiator</u>
EII	$\frac{\text{Valid Ack}}{(\text{MEC}=0)}$	One acknowledgement cell has been received but indicates that cells were lost. If a group of similar acknowledgement cells is received, event E11
	$\frac{(\text{WIEC}=0,}{\text{RSAC}=0}$	occurs only once when receiving the first acknowledgement cell
	SeqNb<=SIE2Fra	• this event results in the new transmission of the SIE2 fragments
	gTotalNb)	numbered from SeqNb
	Received	<u>.</u>
	AND	
	<u>(SeqNb≠AckNb)</u>	
E12	Invalid Ack	Responder receives Acknowledgement cells MEC=1, RSAC =1
	$\frac{(\text{MEC}=1,}{\text{DSAC}=1}$	• <u>this event is expected</u>
	$\frac{KSAU=1}{SocNb-SUE2Error}$	
	TotalNb)	
	Received	
	10001100	

E13	Unexpected SIE1 fragment received AND ((WrongFragNb= 0)OR(SIE1FragN b= <wrongfragnb ))</wrongfragnb 	The responder receives one fragment with the wrong sequence number         (implying that at least one cell is lost). If several fragments of the same         SIE are received, event E13 occurs only once when the first fragment is         received out of sequence.         •       this event results in sending a group of Acknowledgement cells         MEC=0, RSAC=0, SeqNb=0
E14	T105 expires	Timer T105 has exceeded time shown in section 5.5.3.1.         • this event results in a fault at the responder

## 9.4 FSM Actions

		Table 6: Actions
	<u>INITIATOR</u>	
Number	Name	Description
A1	Send SIE1	Send a SIE1 encapsulated into SIE1FragTotalNb fragments, from initiator
		to responder
A2	Start T103 Timer	Start timer T103
A3	Initialize I-SSIE	Set I-SSIE retry counter to 0
	retry count	
A4	Initialize I-Ack	Set I-Ack retry counter to 0
	retry count	
A5	Stop T103 Timer	Stop timer T103
A6	Increment I-SSIE	Increment the counter I-SSIE-Retry-Count
	retry	
A7	Send partial SIE1	Send the SIE1 fragments numbered between SeqNb and
		SIE1FragTotalNb from initiator to responder
A8	Start T104 Timer	Start timer T104
A9	Initialize FragNb	Set FragNb=1
	counts	
A10	Send Ack	Send a group of acknowledgement cells with MEC=1, RSAC=0, and
	MEC=1, RSAC	SeqNb=SIE2FragTotalNb from initiator to responder
	<u>=0,</u>	
	SeqNb=SIE2Frag	
	<u>TotalNb</u>	
A11	Stop T104 Timer	Stop timer T104
A12	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC =0, and
	MEC=0, RSAC	SeqNb=FragNb from initiator to responder
	<u>=0,</u>	
	SeqNb=FragNb-1	
A13	Increment I-Ack	Increment I-Ack counter
A14	Increment FragNb	Increment FragNb counter
A15	Send Ack	Send a group of acknowledgement cells with MEC=1, RSAC=1, and
	$\frac{\text{MEC}=1}{\text{DGAG}=1}$	SeqNb=SIE2FragTotaINb from initiator to responder
	$\frac{\text{RSAC=1}}{\text{RSAC=1}}$	
	SeqNb=SIE2Frag	
A 1 C	<u>10talino</u> OK fer ser din s	It is now accommon do d that the initiation can be a grown of CAC calls to the
A10	OK for sending	It is now recommended that the initiator sends a group of SAC cells to the
A 17	SAC Sand Asla	Send a group of coloroulodgement colle with MEC-0, BSAC-0, and
AI/	<u>Send Ack</u> MEC-0	Send a group of acknowledgement cents with MEC=0, KSAC=0, and SagNb=0 from initiator to responder to make the responder send the full
	$\frac{\text{NILC}=0}{\text{RSAC}=0}$	SIE2 again
	$\frac{NSNC=0}{SeaNb=0}$	<u>SH22 agam</u>
A18	Start T105	Start timer T105
A19	Stop T105	Stop timer T105
Δ20	Initialize	Set WrongFragNh=0
7120	WrongFragNh=0	
A21	WrongFragNh-SI	Set WrongFragNh=SIF2FragNh
1121	E2FragNh	See through rughtemailer rughte
A22	Initialize	Set AckNb=-1
1122	AckNh=-1	
A23	AckNb=SeaNb	Set AckNb=SeqNb
- 120	RESPONDER	
Number	Name	Description

A1	Start T104 Timer	Start timer T104
A2	Initialize FragNb	Set FragNb to 1
	<u>count</u>	
A3	Initialize R-SSIE	Set R-SSIE retry counter to 0
	retry count	
A4	Initialize R-Ack	Set R-Ack retry counter to 0
	retry count	
A5	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	<u>MEC=0,</u>	SeqNb=1 from responder to initiator
	<u>RSAC=0,</u>	
	SeqNb=1	
A6	Stop T102 Timer	Stop timer T102
A7	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	<u>MEC=0,</u>	SeqNb=SIE1FragTotalNb from responder to initiator
	$\frac{\text{KSAC}=0}{\text{SecNib}}$	
	SeqIND=SIE1Frag	
10	<u>Totalino</u> Sond Aak	Sand a group of asknowledgement calls with MEC-0, BSAC-0, and
Ao	MEC-0	Send a group of acknowledgement cents with MEC-0, KSAC-0, and
	$\frac{NILC=0}{RSAC=0}$	Sequo-11agivo-1 nom responder to initiator
	<u>Nonc=0,</u> SeaNh=FragNh-1	
A9	Increment R-Ack	Increment R-Ack counter
A10	Increment FragNh	Increment FragNb counter
A11	Send SIE2	Send the SIE2 fragments numbered between SeqNb and
		SIE2FragTotalNb from responder to initiator
A12	Start T103 Timer	Start timer T103
A13	Stop T103 Timer	Stop timer T103
A14	Increment R-SSIE	Increment R-SSIE-Retry-Count
A15	Send partial SIE2	Send the SIE2 fragments numbered between SeqNb and
		SIE2FragTotalNb from responder to initiator
A16	OK for sending	It is now recommended that the responder sends a group of SAC cells to
	SAC	the initiator
A17	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	<u>MEC=0,</u>	SeqNb=0 from responder to initiator to make the initiator send the full
	<u>RSAC=0,</u>	<u>SIE1 again</u>
	<u>SeqNb=0</u>	
A18	<u>Start T105</u>	Start timer T105
A19	<u>Stop T105</u>	Stop timer T105
A20	<u>Initialize</u>	<u>Set WrongFragNb=0</u>
	WrongFragNb=0	
A21	WrongFragNb=SI	Set WrongFragNb=SIE1FragNb
1.22	<u>ElFragNb</u>	
A22	Initialize	Set AckIND=-1
1.00	ACKIND=-1	
A23	<u>AckNb=SeqNb</u>	<u>Set AckNb=SeqNb</u>

## 9.5 FSM Summary Table

States x Events:S1-IS2-IS3-IS4-IS2E1A1, A2, A3 A4, A22 S2-IN/AN/AN/AN/AE2N/AA5, A18, A20 S3-IN/AN/AN/AE3N/AA1, A2, A6 S2-IN/AN/AN/AE4N/AS6-IN/AN/AN/AE5N/AA7, A2, A6, S2-IN/AN/AN/AE6N/AN/AA1, A2, A6, S2-IN/AN/AN/AE7N/AA7, A2, A6, S2-IA19, A8, A9, S2-IN/AN/AE7N/AN/AA10, A18, S5-IN/AN/A	S6-I           N/A           N/A           N/A           N/A	
E1A1, A2, A3, A4, A22 S2-IN/AN/AN/AN/AE2N/AA5, A18, A20 S3-IN/AN/AN/AE3N/AA1, A2, A6 S2-IN/AN/AN/AE4N/AS6-IN/AN/AN/AE5N/AA7, A2, A6, A23 S2-IN/AN/AN/AE6N/AN/AA1, A2, A6, S6-IN/AN/AN/AE7N/AN/AA1, A2, A6, S2-IA19, A8, A9, A20 S4-IN/AN/AE7N/AN/AA10, A18 S5-IN/AN/A	<u>N/A</u> <u>N/A</u> <u>N/A</u>	
E2 $N/A$ $A5, A18, A20$ $S3-I$ $N/A$ $N/A$ $N/A$ $N/A$ E3 $N/A$ $A1, A2, A6$ $S2-I$ $N/A$ $N/A$ $N/A$ $N/A$ E4 $N/A$ $S6-I$ $N/A$ $N/A$ $N/A$ $N/A$ E5 $N/A$ $A7, A2, A6, A23$ $S2-I$ $N/A$ $N/A$ $N/A$ E6 $N/A$ $N/A$ $A19, A8, A9, A20$ $S4-I$ $N/A$ $N/A$ E7 $N/A$ $N/A$ $A10, A18, S5-I$ $N/A$ $N/A$ E8 $N/A$ $N/A$ $N/A$ $N/A$ $N/A$	<u>N/A</u> <u>N/A</u>	
E3N/AA1, A2, A6 S2-IN/AN/AN/AE4N/AS6-IN/AN/AN/AE5N/AA7, A2, A6, A23 S2-IN/AN/AN/AE6N/AN/AA19, A8, A9, 	<u>N/A</u>	
E4N/AS6-IN/AN/AN/AE5N/AA7, A2, A6, A23 S2-IN/AN/AN/AE6N/AN/AA19, A8, A9, A20 S4-IN/AN/AE7N/AN/AA10, A18 S5-IN/AN/AE8N/AN/AN/AA11, A10 S5-IN/A	NT/A	
E5N/AA7, A2, A6, A23 S2-IN/AN/AN/AE6N/AN/AA19, A8, A9, A20 S4-IN/AN/AE7N/AN/AA10, A18 S5-IN/AN/AE8N/AN/AN/AA11,A10 S5-IN/A	<u>N/A</u>	
E6N/AN/AA19, A8, A9, A20 S4-IN/AN/AE7N/AN/AA10, A18 S5-IN/AN/AE8N/AN/AN/AA11,A10 S5-IN/A	<u>N/A</u>	
E7 $N/A$ $N/A$ $A10, A18$ $\underline{S5-I}$ $N/A$ $N/A$ E8 $N/A$ $N/A$ $N/A$ $A11,A10$ $\underline{S5-I}$ $N/A$	<u>N/A</u>	
E8 $N/A$ $N/A$ $A11,A10$ $N/A$ E8 $N/A$ $A11,A10$ $N/A$	<u>N/A</u>	
	<u>N/A</u>	
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	<u>N/A</u>	
E10         N/A         S6-I         N/A	<u>N/A</u>	
E11 <u>N/A</u> <u>N/A</u> <u>N/A</u> <u>A14, A8</u> <u>N/A</u> <u>S4-I</u>	<u>N/A</u>	
E12         N/A         N/A         N/A         A15, A           S1-I	<u>.16</u> <u>N/A</u>	
E13 $\underline{N/A}$ $\underline{N/A}$ $\underline{A19, A17, A12, A8, A13, A/A}$ $\underline{A13, A8, A21}$ $\underline{A21}$ $\underline{S3-I}$ $\underline{S4-I}$	<u>N/A</u>	
E14 $\underline{N/A}$ $\underline{N/A}$ $\underline{S6-I}$ $\underline{N/A}$ $\underline{N/A}$	37/4	

Table 8: Responder Summary.									
<u>States x</u>	<u>S1-R</u>	<u>S2-R</u>	<u>S3-R</u>	<u>S4-R</u>	<u>S5-R</u>	<u>S6-R</u>			
<u>Events:</u>									
E1	<u>A1, A2, A3,</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
	<u>A20</u>								
	<u>S2-R</u>								
E2	<u>A5, A18</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
	<u>S3-R</u>								
E3	<u>N/A</u>	<u>A6, A7</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
		<u>S3-R</u>							
E4	<u>A17 A9, A1</u>	<u>A8, A9, A1</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
	<u>S1-R</u>	<u>S2-R</u>							
E5	<u>S6-R</u>	<u>S6-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
E6	<u>N/A</u>	<u>A10, A1</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
		<u>S2-R</u>							
E7	<u>N/A</u>	<u>N/A</u>	<u>A11, A12,</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
			<u>A22</u>						
			<u>S4-R</u>						
E8	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A13, A18</u>	<u>N/A</u>	<u>N/A</u>			
				<u>S5-R</u>					
E9	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A11, A12,</u>	<u>N/A</u>	<u>N/A</u>			
				<u>A14</u>					
				<u>S4-R</u>					
E10	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>S6-R</u>	<u>N/A</u>	<u>N/A</u>			
E11	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A15, A12,</u>	<u>N/A</u>	<u>N/A</u>			
				<u>A14, A23</u>					
				<u>S4-R</u>					
E12	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A19, A16, A4,</u>	<u>N/A</u>			
					<u>A20</u>				
					<u>S1-R</u>				
E13	<u>A17, A9, A1,</u>	<u>A8, A9, A1,</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>			
	<u>A21</u>	<u>A21</u>							
	<u>S1-R</u>	<u>S2-R</u>							
E14	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>S6-R</u>	<u>N/A</u>			

Т

### 9.6 <u>Renegotiation Finite State Machines (FSMs) when the three-</u> way SME protocol is used

The Finite State Machines (FSMs) described in this section specify the intended behavior for the Renegotiation protocol. These FSMs correspond to the textual procedures described in Section 5.4 of this specification. If there are any discrepancies between the textual procedures and the FSM tables, the FSM tables shall take precedence.

The FSMs covers two potential configurations:

- 3. Initiator of security association negotiation,
- 4. Responder to security association negotiation.

The FSMs are described in five sections:

- 6. <u>The FSM Graphical Views are shown in Section 9.6.1.</u>
- 7. <u>All FSM States are described in section 9.6.2.</u>
- 8. <u>All FSM Events are described in section 9.6.3.</u>
- 9. <u>All FSM Actions are described in section</u> 9.6.4.
- 10. The FSM Summary Tables are shown in section 9.6.5.

#### 9.6.1 FSM Graphical View

The notations used are the following:

- SIE1, SIE2, SIE3 are the SSIE of the first, second and third flow.
- <u>SIE1FragTotalNb</u>, <u>SIE2FragTotalNb</u>, <u>SIE3FragTotalNb</u> are the number of fragments of <u>SIE1</u>, <u>SIE2</u>, and <u>SIE3</u>, that is the number of negotiation cells necessary to transport the <u>SSIE</u>.
- <u>SIE2FragNb</u>, and <u>SIEFragNb</u> are the number indicated respectively in the SIE2, and either the SIE1 or <u>SIE3 fragment received</u>.
- WrongFragNb is the number of the first unexpected fragment (with a wrong number) belonging to the same SIE.
- <u>SeqNb is the sequence number indicated in the acknowledgement cells.</u>
- <u>AckNb is the sequence number of the last acknowledgement cell received.</u>
- FragNb is the number of the next SSIE fragment to be received.

During initialization in the S1-R Idle state, the R-Ack Retry count counter and the WrongFragNb are assumed to be set to 0.



Figure 8. Initiator FSM.

S1-R : IDLE for next fragmen	aiting S3-R:W SIE1 for SIE t process	Vaiting 2 local	S4-R:W for Ack	Vaiting a 0,0	S5-R: W for 1 st fr of SIE3	Vaiting agment	S6-R: V for nex fragme	Waiting t SIE3 nt	S7-R: Waiting for SIE3 local process	S8
E1: 1 st fragment of SIE1 SIEFragment>1 A1 : Start T104 A2 : Initialize FragNb=1 A3 : Initialize R-SIE Retry count=0 A22 : Initialize WrongFragNb=0	E3: full SIE1 FragNb=SIE1FragTota INb A6 : StopT104 A7 : Send Ack 0,0,SIE1FragTotalNb E4: T104 expires A8: Send Ack 0,0,FragNb-1 A9 : Increment R- Ack retry count A1 : Start T104	E7: Local pr over A11 : Send 3 A12 : Start 7 A3 : Initializ SIE retry co A2 : Initializ FragNb=1 A24 : Initial AckNb=-1	SIE2 F103 ze R- unt=0 ze	E8: Ack 0,0,SIE2Fra A13 : Stop A20 : Start 7 A22 : Initia WrongFrag E9: T103 0 A11 : Star A12 : Star A14 : Incr SIE Retry E10: R	agTotalNb T103 F105 lize Nb=0 expires d SIE2 t T103 ement R- count -SIE retry co	E1: SIE3 SIE3Frag [°] A21 : Stop A1 : Start A22 : Initi WrongFra E2: SIE3 <u>SIE3Frag[°]</u> A16 : Sen 1,0,FragN A20 :Start unt exceede	FotalNb>1 o T105 T104 alize gNb=0 FotalNb=1 d Ack b T105 ed max	E3: full SII FragNb=SI gTotalNb A6 : Stop 7 A16 : Send 1,0,SeqNb: ragTotalNt E5: R-Ack 1	E3 E3Fra F104 Adk =SIE3F Petry count exceede	ed max
SIE1FragTotalNb=1 A5 : Send Ack 0,0,1 A20 :Start T105	E5: R-Ack retry cou	nt exceeded M	ax Max					E5: R-Acl A21 : Stop	c retry count excee p T105	ded ma
E4 : T104 expire A19: Send Ack 0[0,0 A9 : Increment R-Ack retry count A1 : Start T104 E13 : Unexpected SIE AND((WrongFragNb= 0)0R(SIEFragNb= WrongFragNb)) A19: Send Ack 0,0,0 A9 : Increment R-Ack retry count A1 : Start T104 A23 : WrongFragNb= SIEFragNb	A20 :Start T105 E3: R-Ack retry count E4: T104 expire A19: Send Ack 00,0 A9 : Increment R-Ack retry count A1 : Start T104 E13: Unexpected SIE AND((WrongFragNb= 0)OR(SIEFragNb= E6: SIE1 fragment FragNb <sie1fragtotal Nb A10: Increment FragNb A10: Increment FragNb A10: Increment FragNb= 0)OR(SIEFragNb= E13: Unexpected SIE AND((WrongFragNb= 0)OR(SIEFragNb= VrongFragNb)) A19: Send Ack 0,0,0 A9 : Increment R-Ack retry count A1 : Start T104 A23 : WrongFragNb= SIEFragNb E13: Unexpected SIE AND((WrongFragNb= 0)OR(SIEFragNb= VrongFragNb)) A19: Send Ack 0,0,0 A9 : Increment R-Ack retry count A1 : Start T104 A23 : WrongFragNb= SIEFragNb E13: Unexpected SIE AND((WrongFragNb= 0)OR(SIEFragNb= SIEFragNb</sie1fragtotal 		t exceeded Max int exceeded Max E12: SIE3 local p A17 : Send Ack 1, A18 : OK for send		E11: Ack 0.0.SeqNb<=SIE2Hra gTotalNb AND (SeqNb≠AckNb) A15: Send SIE2 fragment from SeqNb to SIE2FragTotalNb-1 A12: Start T103 A14: Increment R- SIE Retry count A25: AckNb=SeqNb SIE3FragTotalNb bire SAC		E4 : T104 extrices A19 : Send Ack 0,0,0 A21 : Stop T105 A1 : Start T104 A9 : Increment R- Ack retry count E13 : Unexpedted SE AND ((WrongFragNb=0) OR(SIEFragNb=0) OR(SIEFragNb= WrongFragNb)) A21 : Stop T105 A19: Send Ack 0,0,0 A9 : Increment R- Ack retry count A1 : Start T104 A23:WrongFragNb = SIEFragNb		expres Ack -11 T104 B3FragT ment 104 Cted gNb=0) gNb=0) gNb=(Nb)) ck 0,0, ent R- punt 104 FragNb b	

#### Figure 9. Responder FSM.

## 9.6.2 FSM States

<b>INITIATOR</b>	<u>-1</u>		
Number	<u>Name</u>	Messages Outstanding	Description
S1 <u>–I</u>	Idle	<u>None, or</u> SAC cells	A new security parameters negotiation is required

S2 <u>–I</u>	<u>Waiting for</u> <u>Ack</u> <u>MEC=0,</u> <u>RSAC =0</u>	<u>Full SIE1</u>	Initiator has initiated a security association negotiation by sending a SIE1 to the responder. It is waiting for the Responder to acknowledge the SIE1
S3 <u>–I</u>	Waiting for the 1 st fragment of SIE2	None	Responder sends the first fragment of the SIE2
S4 <u>–I</u>	Waiting for the next SIE2 fragments	None	Responder sends the next fragments of the SIE2
S5 <u>-I</u>	Waiting for the SIE3 local construction	<u>Acknowledge</u> <u>ment cells</u> <u>MEC=1.</u> <u>RSAC=0</u>	Initiator waits until the SIE3 is constructed and ready to be sent.
S6 <u>-I</u>	Waiting for Acknowledg ement cells MEC=1 RSAC=0	<u>Full SIE3</u>	Initiator waits for the acknowledgement cells which acknowledge the receipt of the SIE3
S7 <u>-I</u>	Waiting for Acknowledg ement cells MEC=1 RSAC=1	<u>None</u>	Initiator waits for the acknowledgement cells which informs it that the responder is ready for security association changeover
S8 <u>-I</u>	Failed		Error occurred during the security association
			negotiation.
<u>RESPONDER</u>	<u>-R</u>		negotiation.
<u>RESPONDER</u> Number	<u>-R</u> <u>Name</u>	Messages Outstanding	Description
<u>RESPONDER</u> <u>Number</u> <u>S1-R</u>	<u>-R</u> <u>Name</u> <u>Idle</u>	Messages Outstanding None, or SAC cells + Acknowledge ment cells MEC=1, RSAC=1	Description         No requests for a new security parameters association negotiation
<u>RESPONDER</u> <u>Number</u> <u>S1-R</u> <u>S2-R</u>	<u>-R</u> Name Idle Waiting for next SIE1 fragment	Messages Outstanding None, or SAC cells + Acknowledge ment cells MEC=1, RSAC=1 None	Description         No requests for a new security parameters association negotiation         Responder waits for the next fragments of the SIE1 if any.
<u>RESPONDER</u> <u>Number</u> <u>S1-R</u> <u>S2-R</u> <u>S3-R</u>	<u>-R</u> Name Idle Waiting for next SIE1 fragment Waiting for SIE2 local construction	Messages         Outstanding         None, or         SAC cells +         Acknowledge         ment cells         MEC=1,         RSAC=1         None	Description         No requests for a new security parameters association negotiation         Responder waits for the next fragments of the SIE1 if any.         Responder waits until the SIE2 is locally constructed
RESPONDER         Number         S1-R         S2-R         S3-R         S4-R	<u>R</u> Name         Idle         Mating for         next SIE1         fragment         Waiting for         SIE2 local         construction         Waiting for         Acknowledg         ement cells         MEC=0         RSAC=0	Messages         Outstanding         None, or         SAC cells +         Acknowledge         ment cells         MEC=1,         RSAC=1         None         None         Full SIE2	Description         No requests for a new security parameters association negotiation         Responder waits for the next fragments of the SIE1 if any.         Responder waits until the SIE2 is locally constructed         Responder sends a SIE2 to the initiator. It is waiting for the initiator to acknowledge the SIE2
RESPONDER         Number         \$1-R         \$2-R         \$2-R         \$3-R         \$3-R         \$5-R	<u>R</u> Name         Idle         Waiting for         next SIE1         fragment         Waiting for         SIE2 local         construction         Waiting for         Acknowledg         ement cells         MEC=0         RSAC=0         Waiting for         fragment of         SIE3	Messages         Outstanding         None, or         SAC cells +         Acknowledge         ment cells         MEC=1,         RSAC=1         None         Full SIE2         None	Description         No requests for a new security parameters association negotiation         Responder waits for the next fragments of the SIE1 if any.         Responder waits until the SIE2 is locally constructed         Responder sends a SIE2 to the initiator. It is waiting for the initiator to acknowledge the SIE2         Initiator sends the first fragment of the SIE3

<u>S7-R</u>	Waiting for SIE3 local process	Acknowledge ment cells MEC=1, RSAC=0	Responder waits until the SIE3 is processed locally
<u>S8-R</u>	<u>Failed</u>	None	Error occurred during the security association negotiation.

#### 9.6.3 FSM Events

	Table 9: Events				
	<b>INITIATOR</b>				
Number	<u>Name</u>	Description			
E1	Security	The initiator has been requested to negotiate a new security association			
	Association				
	Update Request				

E2	Valid Ack ( <u>MEC=0,</u> <u>RSAC=0,</u> <u>SeqNb=SIEFragT</u> otalNb) Received	Acknowledgement cells have been received and indicate that the SSIE fragmented into SIEFragTotalNb fragments is fully received • this event is expected
E3	T103 expires	Timer T103 has exceeded time shown in section 5.4.3         • this event results in a new SSIE transmission
E4	<u>I-SSIE-Retry-</u> Count Exceeded	Initiator has sent SSIE the maximum number of times.           • This event results in a fault at the initiator
E5	Valid Ack (MEC=0, RSAC=0, SeqNb<=SIEFrag TotalNb) Received AND (SeqNb≠AckNb)	<ul> <li><u>One acknowledgement cell has been received but indicates that cells were lost. If a group of similar acknowledgement cells is received, event E5 occurs only once when receiving the first acknowledgement cell.</u></li> <li><u>this event results in the new transmission of the SSIE fragments numbered from SeqNb</u></li> </ul>
E6	Valid SIE2 first fragment Received (SIE2FragTotalN b>1)	<ul> <li><u>The initiator receives the first fragment of the SIE2</u></li> <li><u>this event results in new fragments being expected</u></li> </ul>
E7	<u>Valid SIE2 fully</u> <u>Received</u> ( <u>SIE2FragTotalN</u> <u>b=1)</u>	<ul> <li><u>The initiator receives the full SIE2 in one fragment</u></li> <li><u>this event results in no more fragments being expected</u></li> </ul>
E8	Valid SIE2 fully Received (FragNb=SIE2Fra gTotalNb)	<ul> <li><u>The initiator receives the full SIE2 in SIE2FragTotalNb fragments</u></li> <li><u>this event is expected if the SIE2 requires more than one fragment.</u></li> </ul>
E9	T104 expires	Timer T104 has exceeded time shown in section 5.5.3.1.         • this event results in Acknowledgement cells MEC=0, RSAC=0, SeqNb=FragNb-1
E10	I-Ack Retry Count Exceeded	Initiator has sent a group of acknowledgement cells (MEC=0, RSAC=0) the maximum number of times. • this event results in a fault at the initiator
E11	One SIE2 fragment Received	<ul> <li><u>The initiator receives one more fragment</u></li> <li><u>this event is expected if the SIE2 requires more than one fragment.</u></li> </ul>
E12	Local construction of SIE3 over	The initiator processes SIE2 and constructs the SIE3 <ul> <li>this event is expected.</li> </ul>
E13	Valid Ack (MEC=1, RSAC=0, SeqNb=SIEFragT otalNb) Received	<u>Acknowledgement cells have been received</u> <u>this event is expected</u>
E14	Valid Ack (MEC=1, RSAC=1, SeqNb=SIEFragT otalNb) Received	Initiator receives Acknowledgement cells MEC=1, RSAC=1         • this event is expected
E15	Unexpected SIE2 fragment received AND ((WrongFragNb=	The initiator receives one fragment with the wrong sequence number (implying that at least one cell is lost). If several fragments of the same SIE are received, event E15 occurs only once when the first fragment is received out of sequence.

	0)OR(SIE2FragN	• This event results in sending a group of Acknowledgement cells
	<u>b=<wrongfragnb< u=""></wrongfragnb<></u>	MEC=0, RSAC=0, SeqNb=0
	<u>))</u>	
E16	T105 expires	Timer T105 has exceeded time shown in section 5.5.3.1.
		• this event results in a fault at the initiator
	<u>RESPONDER</u>	
Number	Name	Description
E1	Valid SSIE first	The responder receives the first fragment of the SSIE
	fragment	• this event results in new fragments being expected
	Received	
	(SIEFragTotalNb	
	<u>&gt;1)</u>	

E2	Valid SSIE	The responder receives the full SSIE in one fragment
	received	• this event results in no more fragments being expected
	(SIEFragTotalNb	
	<u>=1)</u>	
E3	Valid SSIE fully	The responder receives the full SSIE in SIEFragTotalNb fragments
	received	• this event is expected if the SSIE requires more than one
	(FragNb=SIEFrag	fragment.
	<u>TotalNb)</u>	
E4	T104 Expires	Timer T104 has exceeded time shown in section 5.5.3.1.
		• This event results in a fault at the responder
E5	R-Ack Retry	Responder has sent a group of acknowledgement cells (MEC=0,
	Count Exceeded	RSAC=0) the maximum number of times.
		• This event results in a fault at the initiator
E6	One SSIE	Responder receives one more fragment
	fragment received	• this event is expected if the SSIE requires more than one
		fragment.
E7	Local construction	Responder processes SIE1 and constructs the SIE2
	of SIE2	• this event is expected.
E8	Valid Ack	Acknowledgement cells have been received
	(MEC=0,	• this event is expected
	RSAC=0,	<u>.</u>
	SeqNb=SIE2Frag	
	TotalNb)	
	Received	
E9	T103 expires	Timer T103 has exceeded time shown in section 5.5.3.1.
		• this event results in a new SIE2 transmission
E10	I-SSIE Retry	Initiator has sent SSIE of FLOW2 the maximum number of times.
	Count Exceeded	• This event results in a fault at the initiator
E11	Valid Ack	One acknowledgement cell has been received but indicates that cells were
	(MEC=0, RSAC	lost. If a group of similar acknowledgement cells is received, event E11
	=0,	occurs only once when receiving the first acknowledgement cell.
	SeqNb<=SIE2Fra	• this event results in the new transmission of the SIE2 fragments
	gTotalNb)	numbered from SeqNb
	Received	
	AND	
	<u>(SeqNb≠AckNb)</u>	
E12	Local process of	Responder processes SIE3 locally and is ready for security association
	SIE3 over	changeover
		• this event is expected
E13	Unexpected SIE	The responder receives one fragment with the wrong sequence number
	fragment received	(implying that at least one cell is lost). If several fragments of the same
		SIE are received, event E13 occurs only once when the first fragment is
		received out of sequence.
1		
		<ul> <li>this event results in sending a group of Acknowledgement cells</li> </ul>
		<ul> <li>this event results in sending a group of Acknowledgement cells MEC=0, RSAC=0, SeqNb=0</li> </ul>
E14	T105 expires	this event results in sending a group of Acknowledgement cells <u>MEC=0, RSAC=0, SeqNb=0</u> Timer T105 has exceeded time shown in section 5.5.3.1.

## 9.6.4 FSM Actions

Table 10: Actions					
	<b>INITIATOR</b>				
Number	Name	Description			
A1	Send SSIE	Send a SSIE from initiator to responder using SIEFragTotalNb fragments			
A2	Start T103 Timer	Start timer T103			
A3	Initialize I-SSIE	Set I-SSIE retry counter to 0			
	retry counter				
A4	Initialize I-Ack	Set I-Ack retry counter to 0			
	retry counter				
A5	Stop T103 Timer	Stop timer T103			
A6	Increment I-SSIE	Increment the counter I-SSIE-Retry-Count			
	<u>retry</u>				
A7	Send partial SSIE	Send the SSIE fragments numbered between SeqNb and SIEFragTotalNb			
		from initiator to responder			
A8	Start T104 Timer	Start timer T104			
A9	Initialize FragNb	Set FragNb=1			
	<u>counts</u>				
A10	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and			
	<u>MEC=0,</u>	SeqNb=SIE2FragTotalNb from initiator to responder			
	<u>RSAC=0,</u>				
	SeqNb=SIE2Frag				
	TotalNb				
All	Stop T104 Timer	Stop timer T104			
A12	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and			
	$\frac{\text{MEC}=0}{\text{DSAC}=0}$	Seqind=Fraging from initiator to responder			
	$\frac{\text{KSAC}=0}{\text{Soc}\text{Nb}-\text{Erog}\text{Nb}} 1$				
A13	Increment I Ack	Increment I. Ack counter			
A14	Increment FreeNh	Increment FragNh counter			
A14	OK for sonding	It is now recommended that the initiator sends a group of $SAC$ calls to the			
AIJ	SAC	It is now recommended that the initiator sends a group of SAC cens to the responder			
A16	Sand Ack	Sand a group of acknowledgement calls with MEC-0. PSAC-0 and			
AIU	MEC=0 RSAC	SeaNh=0 from initiator to responder to make the responder send the full			
	=0 SeaNb=0	SIF2 again			
A17	Start T105	Start timer T105			
A18	Stop T105	Stop timer T105			
A19	Initialize	Set WrongFragNh=0			
1117	WrongFragNb=0	<u>ber wrongringrio-o</u>			
A20	WrongFragNb=SI	Set WrongFragNb=SIE2FragNb			
	E2FragNb				
A21	Initialize	Set AckNb=-1			
	AckNb=-1				
A22	AckNb=SeqNb	Set AckNb=SeqNb			
	RESPONDER				
Number	Name	Description			
Al	Start T104 Timer	Start timer T104			

A2	Initialize FragNb	Set FragNb to 1
A3	Initialize R-SSIE	Set R-SSIE retry counter to 0
	retry counter	
A4	Initialize R-Ack	Set R-Ack retry counter to 0
	retry counter	
A5	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	<u>MEC=0,</u>	SeqNb=1 from responder to initiator
	<u>RSAC=0,</u>	
	<u>SeqNb=1</u>	
A6	Stop T104 Timer	Stop timer T104
A7	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	<u>MEC=0,</u>	SeqNb=SIE1FragTotalNb from responder to initiator
	<u>RSAC=0,</u>	
	SeqNb=SIE1Frag	
	TotalNb	
A8	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	<u>MEC=0,</u>	SeqNb=FragNb-1 from responder to initiator
	$\frac{\text{KSAC}=0}{\text{KSAC}=0}$	
4.0	SeqIND=FragIND-1	The second D. A. I. Second on
A9	Increment R-Ack	Increment R-Ack counter
A10	Increment Fraginb	Increment Fragino counter
AII	Send SIE2	Send the SIE2 fragments numbered between SeqNb and
A 10	Quert T102 T'mere	SIE2FragTotalNb from responder to initiator
A12	Start 1103 Timer	Start timer 1103
A13	Stop 1103 Timer	Stop timer T103
Al4	Increment R-SSIE	Increment R-SSIE-Retry-Count
A15	Send partial SIE2	Send the SIE2 fragments numbered between SeqNb and
110	0 1 4 1	SIE2Frag1 otalNb from responder to initiator
A16	Send Ack	Send a group of acknowledgement cells with MEC=1, RSAC=0, and
	$\frac{\text{MEC}=1}{\text{DSAC}=0}$	Sequo=SIEFrag1 otaling from responder to initiator
	<u>NDAC-0,</u> SogNb-SIFErogT	
	otalNb	
Δ17	Send Ack	Send a group of acknowledgement cells with MEC-1 $RSAC-1$ and
A17	MEC-1	SeaNb-SIFFragTotalNb from responder to initiator
	$\frac{\text{MLC}=1}{\text{RSAC}=1}$	<u>beqrio-bibing roun to non responder to initiator</u>
	SeaNb=SIEFragT	
	otalNb	
A18	OK for sending	It is now recommended that the responder sends a group of SAC cells to
-	SAC	the initiator
A19	Send Ack	Send a group of acknowledgement cells with MEC=0, RSAC=0, and
	MEC=0,	SeqNb=0 from responder to initiator to make the initiator send the full
	<u>RSAC=0,</u>	SIE again
	<u>SeqNb=0</u>	
A20	Start T105	Start timer T105
A21	<u>Stop T105</u>	Stop timer T105
A22	Initialize	Set WrongFragNb=0
	WrongFragNb=0	
A23	WrongFragNb=SI	Set WrongFragNb=SIEFragNb (either SIE1FragNb or SIE3FragNb)
	<u>EFragNb</u>	
A24	Initialize	Set AckNb=-1
	AckNb=-1	
A25	AckNb=SeqNb	Set AckNb=SeqNb

## 9.6.5 FSM Summary Table

Table 11: Initiator Summary.								
<u>States x</u> Events:	<u>S1-I</u>	<u>S2-I</u>	<u>83-I</u>	<u>S4-I</u>	<u>S5-I</u>	<u>S6-I</u>	<u>87-I</u>	<u><b>S8-I</b></u>
E1	<u>A1, A2,</u> <u>A3, A4,</u> <u>A21</u> <u>S2-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E2	<u>N/A</u>	<u>A5, A17,</u> <u>A19</u> <u>S3-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E3	<u>N/A</u>	<u>A1, A2,</u> <u>A6</u> <u>S2-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A1, A2,</u> <u>A6</u> <u>S6-I</u>	<u>N/A</u>	<u>N/A</u>
E4	<u>N/A</u>	<u>S8-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>S8-I</u>	<u>N/A</u>	<u>N/A</u>
E5	<u>N/A</u>	<u>A7, A2,</u> <u>A6, A22</u> <u>S2-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A7, A2,</u> <u>A6, A22</u> <u>S6-I</u>	<u>N/A</u>	<u>N/A</u>
E6	<u>N/A</u>	<u>N/A</u>	<u>A18, A8,</u> <u>A9, A19</u> <u>S4-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E7	<u>N/A</u>	<u>N/A</u>	<u>A10, A17</u> <u>S5-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E8	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A11, A10</u> <u>S5-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E9	<u>N/A</u>	<u>N/A</u>	<u>A18,</u> <u>A16,</u> <u>A13, A8</u> <u>S3-I</u>	<u>A12, A8,</u> <u>A13</u> <u>S4-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E10	<u>N/A</u>	<u>N/A</u>	<u>A18</u> <u>S8-I</u>	<u>S8-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E11	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A14, A8</u> <u>S4-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E12	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A1, A2,</u> <u>A3, A9,</u> <u>A21</u> <u>S6-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E13	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A4, A17</u> <u>S7-I</u>	<u>N/A</u>	<u>N/A</u>
E14	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A18, A15</u> <u>S1-I</u>	<u>N/A</u>
E15	<u>N/A</u>	<u>N/A</u>	<u>A18,</u> <u>A16,</u> <u>A13, A8,</u> <u>A20</u> <u>S3-I</u>	<u>A12, A8,</u> <u>A13, A20</u> <u>S4-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E16	<u>N/A</u>	<u>N/A</u>	<u>S8-I</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>S8-I</u>	<u>N/A</u>

Table 12: Responder Summary.								
<u>States x</u> Events:	<u>S1-R</u>	<u>S2-R</u>	<u>S3-R</u>	<u>S4-R</u>	<u>S5-R</u>	<u>S6-R</u>	<u>S7-R</u>	<u>S8-R</u>
E1	<u>A1, A2,</u> <u>A3, A22</u> <u>S2-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A21, A1,</u> <u>A22</u> <u>S6-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E2	<u>A5, A20</u> <u>S3-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A16, A20</u> <u>S7-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E3	<u>N/A</u>	<u>A6, A7</u> <u>S3-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A6, A16</u> <u>S7-R</u>	<u>N/A</u>	<u>N/A</u>
E4	<u>A19, A9,</u> <u>A1</u> <u>S1-R</u>	<u>A8, A9,</u> <u>A1</u> <u>S2-R</u>	<u>N/A</u>	<u>N/A</u>	<u>A21, A19,</u> <u>A9, A1</u> <u>S5-R</u>	<u>A8, A9,</u> <u>A1</u> <u>S6-R</u>	<u>N/A</u>	<u>N/A</u>
E5	<u>S8-R</u>	<u>S8-R</u>	<u>N/A</u>	<u>N/A</u>	<u>A21</u> <u>S8-R</u>	<u>S8-R</u>	<u>N/A</u>	<u>N/A</u>
E6	<u>N/A</u>	<u>A10, A1</u> <u>S2-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A10, A1</u> <u>S6-R</u>	<u>N/A</u>	<u>N/A</u>
E7	<u>N/A</u>	<u>N/A</u>	<u>A11, A12,</u> <u>A3, A2,</u> <u>A24</u> <u>S4-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E8	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A13, A20,</u> <u>A22</u> <u>S5-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E9	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A11, A12,</u> <u>A14</u> <u>S4-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E10	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>S8-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E11	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A15, A12,</u> <u>A14, A25</u> <u>S4-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
E12	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>A17, A18,</u> <u>A4, A22</u> <u>S1-R</u>	<u>N/A</u>
E13	<u>A19, A9,</u> <u>A1, A23</u> <u>S1-R</u>	<u>A8, A9,</u> <u>A1, A23</u> <u>S2-R</u>	<u>N/A</u>	<u>N/A</u>	<u>A21, A19,</u> <u>A9, A1,</u> <u>A23</u> <u>S5-R</u>	<u>A8, A9,</u> <u>A1, A23</u> <u>S6-R</u>	<u>N/A</u>	<u>N/A</u>
E14	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>S8-R</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>

## **IV.2 Identifiers**

Table 18 provides a list of identifiers defined in this specification. They are used to identify octet groups in both in-band and signalling messages. The table shows:

- the 8 bit identifier value
- reference section in the specification where the format is defined
- the number of octets in the length field (or "-" if no length field is used)
- the use of the value field (see main body of specification for details)
- the name of the octet group which it identifies

Codepoint	Reference	Size of Length field	Use of Value field	Name
0010 0001*	5.1.5.3.2.1	2	Complex	FLOW1-3WE
0010 0010*	515001		(SSIE)	
0010 0010*	5.1.5.3.2.1	2	(SSIE)	FLOW2-3WE
0010 0011*	5.1.5.3.2.1	2	Complex (SSIE)	FLOW3-3WE
0010 0100	5.1.3.2.10.1	_	_	(Not used - reserved)
0010 0101	5.1.3.2.10.1	_	Sections for: Security Service Confidentiality Authorization	Security Message Exchange Format (FLOW2-2WE optional)
0010 0110	5.1.3.2.10.1	_	Sections for: Security Service Confidentiality Authorization	Security Message Exchange Format (FLOW2-2WE required)
0010 0111	5.1.3.2.10.1	_	Sections for: Security Service Confidentiality Authorization	Security Message Exchange Format (3-way)
0010 1000	5.1.3.2.10.2	1	type + data	Label Based Access Control
0011 0001*	5.1.5.3.2.1	2	None	CONFIRM-AP
0011 0010*	5.1.5.3.2.1	2	Cause value	FAULT
1000 0010	7.1.1	1	name type + value	Initiator Distinguished Name
1000 0011	7.1.2	1	name type + value	Responder Distinguished Name
1000 0100	7.1.3	1	name type $+$ value	Security Agent Distinguished Name
1000 1000	7.2	_	Complex: Sections for: Security Service Declarations Security Service Options, Security Service Algorithm	Security Service Specification Section
1000 1010	7.2.1	-	<u>8</u> 7 Boolean values	Security Service Declaration
1001 0000	7.2.2.1	-	1 of 3 values	Data Confidentiality Service Option
1001 0010	7.2.2.2	-	1 of 5 values	Data Integrity Service Options
1001 0011	7.2.2.3	-	1 of 3 values	Authentication Service Options
1001 0100	7.2.2.4	-	1 of 3 values	Key Exchange Service Options
1001 0101	7.2.2.5	-	1 of 3 values	Session Key Update Service Options

#### Table 18: Security Message Identifiers

Codepoint	Reference	Size of	Use of Value field Name	
_		Length field		
1001 0110	7.2.2.6	-	1 of 3 values	Access Control Service Options
<u>1001 0111</u>	<u>7.2.2.7</u>	-	<u>1 of 3 values</u>	Certificate Exchange Service Options
<u>1001 1000</u>	<u>7.2.2.8</u>	=	<u>1 of 3 values</u>	Management-based Security Message
				Exchange Options
1010 0000	7.2.3.1	1	Algorithm, mode	Data Confidentiality Algorithm
1010 0010	7.2.3.2	1	Algorithm	Data Integrity Algorithm
1010 0100	7.2.3.3	1	Algorithm	Hash Algorithm
1010 0110	7.2.3.4	1	Algorithm	Signature Algorithm
1010 1000	7.2.3.5	1	Algorithm	Key Exchange Algorithm
1010 1010	7.2.3.6	1	Algorithm	Session Key Update Algorithm
1010 1100	1.2.3.1	1	Complex:	Authentication Algorithm Group
			Signature	
			Hash	
1010 1110	7.2.3.8	1	Complex:	Integrity Algorithm Group
		-	Algorithms for:	B
			MAC	
			Signature	
			Key Exchange	
			Key Update	
			Hash	
1011 0000	7.2.3.9	1	Complex:	Confidentiality Algorithm Group
			Algorithms for:	
			Signatura	
			Key Exchange	
			Key Undate	
			Hash	
1011 0010	7.2.3.7/	1	Algorithm details	Signature Algorithm Details
	7.2.3.8/			
	7.2.3.9			
1011 0100	7.2.3.7/	1	Algorithm details	Hash Algorithm Details
1011.0110	7.2.3.8	1	Algorithm details	MAC Algorithm details
1011 1000	7.2.3.8	1	Algorithm details	Key Exchange Algorithm Details
1011 1000	7.2.3.9	1	Algoriumi detans	Key Exchange Augorunni Detans
1011 1010	7.2.3.9	1	Algorithm details	Key Update Algorithm Details
1011 1110	7.2.3.9	1	Algorithm details	Encryption Algorithm Details
1100 0000	7.3	2	Encrypted data	Confidential Parameters Section
1100 0100	7.3.1	-	Complex:	Confidential Parameters
			(Next 3):	
			Master Key	
			1st data Conf Session key	
1100 1000	7.2.2	1	1st data Integ Session key	Martin IZ
1100 1000	7.3.2	1	value	Master Key
1100 1010	1.5.5	1	value	First Data Confidentiality Session Key
1100 1100	1.3.4	1	value	First Data Integrity Session Key

Codepoint	Reference	Size of Length field	Use of Value field	Name
1101 0000	7.4	_	Complex: (next 5): Initiator Random Number Responder Random Number Time-variant stamp Credentials SME or SAS Digital signature	Authentication Section
1101 0100	7.4.1	-	4 octet value	Initiator Random Number
1101 0101	7.4.2	-	4 octet value	Responder Random Number
1101 0110	7.4.3	-	4 octet time stamp 4-octet sequence #	Time-variant Time Stamp
1101 1000	7.4.4	2	type + value	Credentials
1101 1010	7.4.5	1	values (algorithm specific)	Security Message Exchange Digital Signature
1101 1100	7.4.6	2	values (algorithm specific)	SAS Digital Signature

* Although these message types are for in-band only, the codepoints are assigned so as not to overlap with the codepoints used in signaling.

"Complex" indicates the value field contains other fields which are individually identified with one of these identifiers.

## Appendix V OAM Cell Policing Considerations for Security Renegotiation

(This Appendix does not form an integral part of this specification)

The Management-Based Security Message Exchange approach may suffer from the point that some switches police the OAM cells at a fraction of the bandwidth allocated for the virtual circuit (or path). This policing activity could result in degradation of the renegotiation protocol. This feature implies that the minimum of the negotiation period (Tmin) should be a function of this fraction (n%), as follows:

Tmin=(M*53*8)/(n*B)

Where M is the maximum number of OAM cells necessary for negotiation, and B is the bandwidth (in bps) allocated to the VC (or VP).