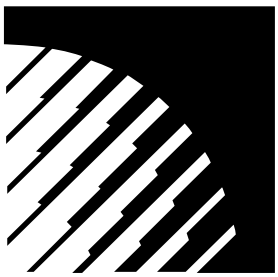

The ATM Forum

Technical Committee

Security Services Renegotiation Addendum to Security Version 1.1

af-sec-0180.000

March, 2002

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 5.5). 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.

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

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

5.1.7.2.1.3 Negotiation OAM Cell Format

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

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

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) The SSIE Fragment field contains one of the 40-byte fragments composing the SSIE.
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								
8	7	6	5	4	3	2	1	Octets
GFC/VPI [11:8]				VPI [7:4]				1
VPI [3:0]				VCI [15:12]				2
VCI [11:4]								3
VCI [3:0]				PTI			CLP=0	4
HEC [7:0]								5
1	1	1	1	Function Type = 0001				6
Relative ID				Function ID = 0011				7
0	0	0	0	Flow Number				8
Sequence Number								9
Reserved						RSAC	MEC	10
Reserved								11-51
0	0	0	0	0	0	CRC [9:8]		52
CRC [7:0]								53

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) The Sequence Number field contains the sequence number of the next-expected negotiation OAM cell.
4) 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.
5) The MEC (Message Exchange Complete) bit indicates either that no other flow should be expected (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 Points for Real Time Security OAM Cells.

Function ID (binary)	Security Function
0001	Data Confidentiality Session Key Changeover (SKC)
0010	Data Integrity Session Key Changeover (SKC)
0011	Security Association Changeover (SAC)
all others	not defined

5.1.7.2.2.3 Security Association Changeover OAM Cell Format

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

Bits								
8	7	6	5	4	3	2	1	Octets
GFC/VPI [11:8]				VPI [7:4]				1
VPI [3:0]				VCI [15:12]				2
VCI [11:4]								3
VCI [3:0]				PTI			CLP=0	4
HEC [7:0]								5
1	1	1	1	Function Type = 0010				6
Relative ID				Function ID = 0011				7
Bank ID				Reserved				8
Reserved								9-51
0	0	0	0	0	0	CRC [9:8]		52
CRC [7:0]								53

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) The Bank ID field is an alternating pattern of 0 hex or F hex, for successive security association changeovers.
3) Reserved bytes are set to 6A hex, reserved bit fields less than 1 byte in length are set to all zeros.
4) The reserved bits included after the 4-bit Bank ID field are provided so that the CKN and IKN fields 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-to-end 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

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

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)

Upon receipt of the negotiation OAM cells, the NUR (Negotiation Update Responder) 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. 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)

Upon receipt of the negotiation OAM cells, the NUI 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 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:

- The negotiation OAM cells receiver detects a cell loss because the last negotiation OAM cell received carries a sequence number greater than expected.
- 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.
- The SSIE is fully received so the negotiation OAM cells received should acknowledge all the 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.
- 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.

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:

1. 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.

1. 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.
2. 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.
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. 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.
2. I-Ack -Retry-Count: This variable is used in conjunction with timer T104 by the Initiator in the 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. 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.
4. R-Ack -Retry-Count: This variable is used in conjunction with timer T104 by the Responder in the 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. I-MAX-SSIE-RETRY: This constant indicates the maximum number of times that the Initiator may resend a full SSIE or fragments of it to the Responder.
2. I-MAX-ACK-RETRY: This constant indicates the maximum number of times that the Initiator may resend a group of acknowledgement cells to the Responder.
3. R-MAX-SSIE-RETRY: This constant indicates the maximum number of times that the Responder may resend a full SSIE or fragments of it to the Initiator.
4. R-MAX-ACK-RETRY: This constant indicates the maximum number of times that the Responder may 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

Timer Name	Timer Value
T103	10 seconds
T104	5 seconds
T105	30 seconds

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

Table II: Constant Values for Renegotiation

Constant Name	Constant Value
I_MAX_SSIE_RETRY	4
I_MAX_ACK_RETRY	10
R_MAX_SSIE_RETRY	4
R_MAX_ACK_RETRY	10

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**
111	protocol error, unspecified (Note 1)
100	invalid information element contents
63	service or option not available
47	resource unavailable

Note 1:
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	7	6	5	4	3	2	1	Octet(s)
1	0	0	0	1	0	1	0	
Security Service Declaration Identifier								x.1
Security Service Declaration								
x	x	x	x	x	x	x	x	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			
0	Confidentiality Service not supported	not requested	not provided
1	Confidentiality Service supported	requested	provided
2			
0	Integrity Service not supported	not requested	not provided
1	Integrity Service supported	requested	provided
3			
0	Authentication Service not supported	not requested	not provided
1	Authentication Service supported	requested	provided
4			
0	Key Exchange Service not supported	not requested	not provided
1	Key Exchange Service supported	requested	provided
5			
0	Certificate Exchange Service not supported	not requested	not provided
1	Certificate Exchange Service supported	requested	provided
6			
0	Session Key Update Service not supported	not requested	not provided
1	Session Key Update Service supported	requested	provided
7			
0	Access Control Service not supported	not requested	not provided
1	Access Control Service supported	requested	provided
8			
0	Management-Based Security Message Exchange (renegotiation) not supported	not requested	not provided
1	Management-Based Security Message Exchange (renegotiation) supported	requested	provided

7.2.2.8 Management-based Security Message Exchange Options

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

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 Exchange (Note)
1 0 0 0 0 0 0 1	Requires Management-based Security Message Exchange

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

9 Management-based Security Message 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. Responder to security association negotiation.

The FSMs are described in five sections:

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

9.1 FSM Graphical View

The notations used are the following:

- SIE1, SIE2 are the SSIE of the first and second flow.
- SIE1FragTotalNb, SIE2FragTotalNb are the number of fragments of SIE1 and SIE2, that is the number of negotiation cells necessary to transport the SSIE.
- SIE1FragNb, SIE2FragNb are the numbers indicated in the SIE1 and SIE2 fragment received.
- WrongFragNb is the number of the first unexpected fragment (with a wrong number) belonging to the same SIE.
- 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.

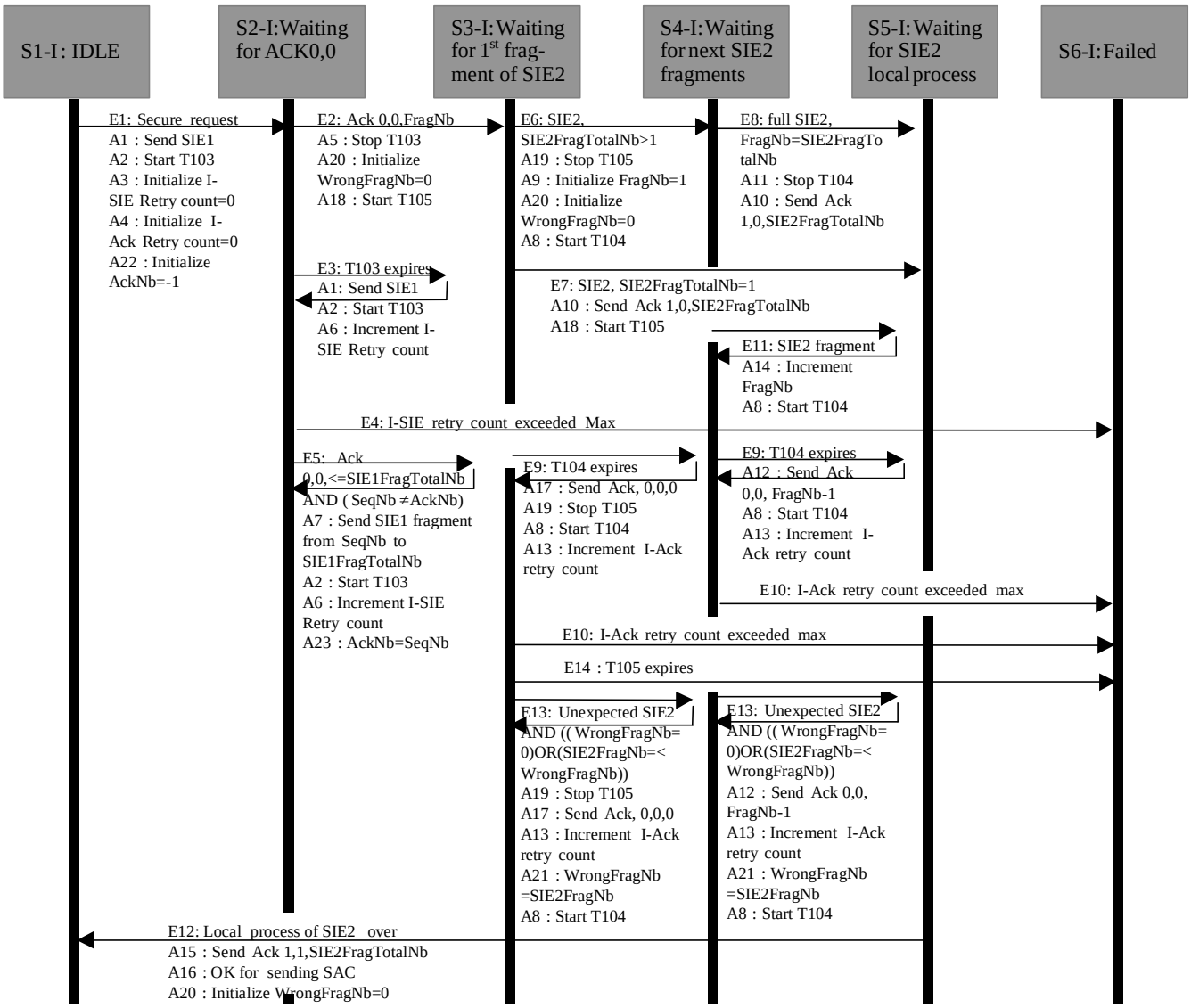

Figure 6. Initiator FSM.

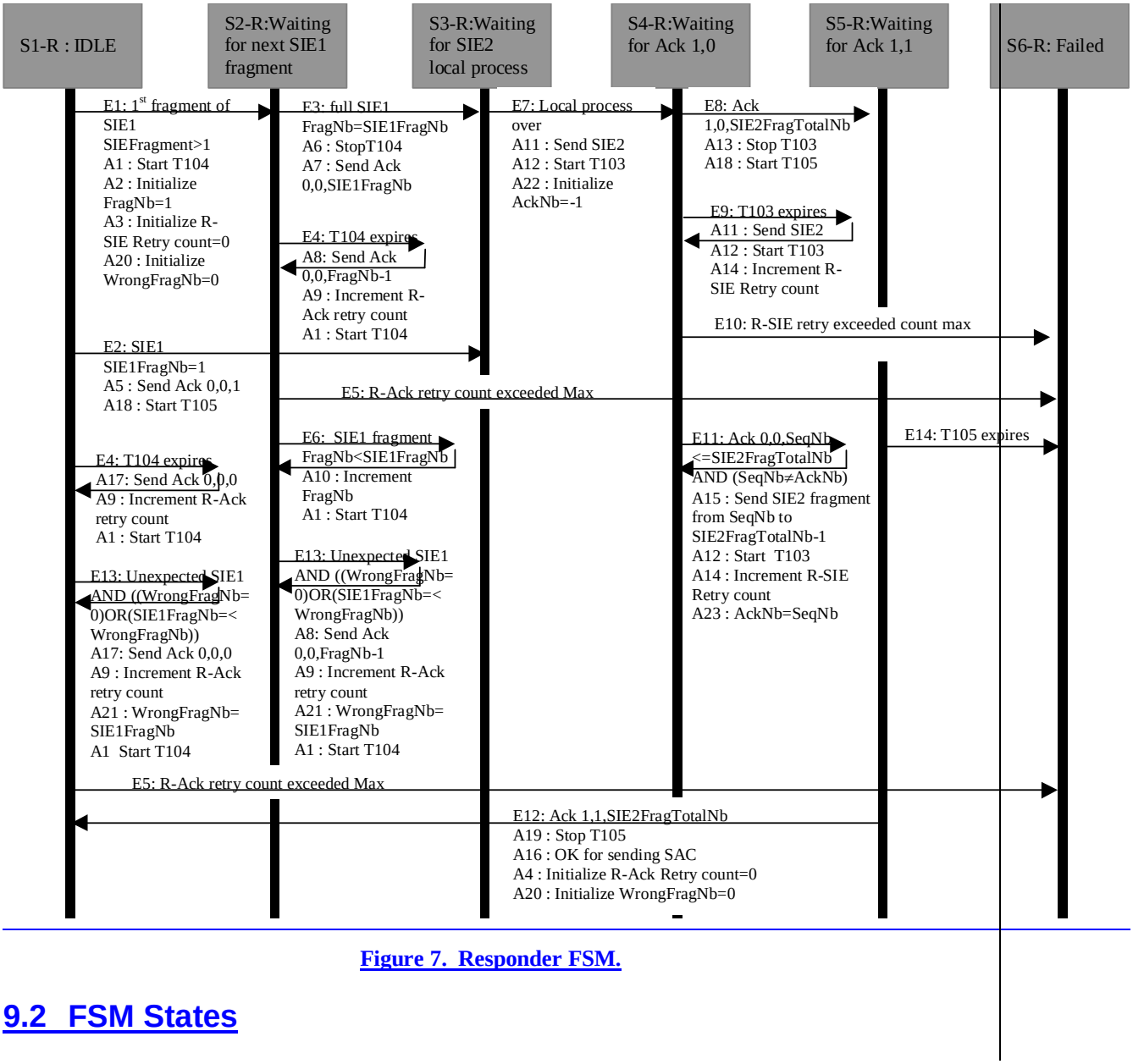

Figure 7. Responder FSM.

9.2 FSM States

INITIATOR	***-I***		
Number	**Name**	**Messages Outstanding**	**Description**
S1–I	Idle	None, or SAC cells + Acknowledgement cells MEC=1, RSAC =1	A new security association negotiation is required
S2–I	Waiting for Ack MEC=0, RSAC =0	Full SIE1	Initiator has initiated a security association negotiation by sending a SSIE 1 to the responder. It is waiting for the Responder to acknowledge the SIE1
S3–I	Waiting for the 1st fragment of SIE2	Acknowledgement cells MEC=0, RSAC =0	Responder sends the first fragment of the SIE2
S4–I	Waiting for the next SIE2 fragments	None	Responder sends the next fragments of the SIE2
S5–I	Waiting for the SIE2 local process	Acknowledgement cells MEC=1, RSAC =0	Initiator waits until the SIE2 is processed locally
S6–I	Failed		Error occurred during the security association negotiation.
RESPONDER	***-R***		
Number	**Name**	**Messages Outstanding**	**Description**
S1-R	Idle	None, or SAC cells	No requests for a new security association negotiation
S2-R	Waiting for next SIE1 fragment	None	Responder waits for the next fragments of the SIE1 if any
S3-R	Waiting for SIE2 local construction	None	Responder waits until the SIE2 is locally constructed
S4-R	Waiting for Acknowledgement cells MEC=1 RSAC =0	Full SIE2	Responder sends a SIE2 to the initiator. It is waiting for the initiator to acknowledge the SIE2
S5-R	Waiting for Acknowledgement cells MEC=1 RSAC =1	None	Responder waits for the acknowledgement cells which informs it that the initiator is ready for security association changeover
S6-R	Failed	None	Error occurred during the security association negotiation.

9.3 FSM Events

Table 5: Events

	INITIATOR	
Number	**Name**	**Description**
E1	Security Context Update Request	The initiator has been requested to negotiate new security parameters
E2	Valid Ack (MEC=0, RSAC =0, SeqNb=SIE1FragTotalNb) Received	Acknowledgement cells have been received • this event is expected
E3	T103 expires	Timer T103 has exceeded time shown in section 5.5.3.1. • this event results in a new SIE1 transmission
E4	I-SSIE-Retry-Count Exceeded	Initiator has sent SSIE of FLOW1 the maximum number of times. • This event results in a fault at the initiator
E5	Valid Ack (MEC=0, RSAC =0, SeqNb<=SIE1FragTotalNb) Received AND (SeqNb≠AckNb)	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. • this event results in the new transmission of the SIE1 fragments numbered from SeqNb
E6	Valid SIE2 first fragment Received (SIE2FragTotalNb>1)	The initiator receives the first fragment of the SIE2 • this event results in new fragments being expected
E7	Valid SIE2 received (SIE2FragTotalNb=1)	The initiator receives the full SIE2 in one fragment • this event results in no more fragments being expected
E8	Valid SIE2 fully received (FragNb=SIE2FragTotalNb)	The initiator receives the full SIE2 in SIE2FragTotalNb fragments • this event is expected if the SIE2 requires more than one fragment.
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 or in Acknowledgement cells MEC=0, RSAC =0, SeqNb=0
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	The initiator receives one more fragment • this event is expected if the SIE2 requires more than one fragment.
E12	Local process over	The initiator processes the SIE2 • this event is expected.
E13	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 E13 occurs only once when the first fragment is received out of sequence.

	0)OR(SIE2FragNb=<WrongFragNb))	• 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 initiator
	RESPONDER	
Number	**Name**	**Description**
E1	Valid SIE1 first fragment Received (SIE1FragTotalNb>1)	The responder receives the first fragment of the SIE1 • this event results in new fragments being expected
E2	Valid SIE1 Received (SIE1FragTotalNb=1)	The responder receives the full SIE1 in one fragment • this event results in no more fragments being expected
E3	Valid SIE1 fully received (FragNb=SIE1FragTotalNb)	The responder receives the full SIE1 in SIE1FragTotalNb fragments • this event is expected if the SIE1 requires more than one fragment.
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 Count Exceeded	Responder 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
E6	One SIE1 fragment received	The responder receives one more fragment • this event is expected if the SIE1 requires more than one fragment.
E7	Local process over	The responder processes the SIE1 • this event is expected.
E8	Valid Ack (MEC=1, RSAC=0, SeqNb=SIE2FragTotalNb) Received	Acknowledgement cells have been received • this event is expected
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 Count Exceeded	Initiator has sent SSIE of FLOW2 the maximum number of times. • This event results in a fault at the initiator
E11	Valid Ack (MEC=0, RSAC=0, SeqNb<=SIE2FragTotalNb) Received AND (SeqNb≠AckNb)	One acknowledgement cell has been received but indicates that cells were lost. If a group of similar acknowledgement cells is received, event E11 occurs only once when receiving the first acknowledgement cell. • this event results in the new transmission of the SIE2 fragments numbered from SeqNb
E12	Invalid Ack (MEC=1, RSAC=1, SeqNb=SIE2FragTotalNb) Received	Responder receives Acknowledgement cells MEC=1, RSAC =1 • this event is expected

E13	Unexpected SIE1 fragment received AND ((WrongFragNb=0)OR(SIE1FragNb=<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

	INITIATOR	
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 retry count	Set I-SSIE retry counter to 0
A4	Initialize I-Ack retry count	Set I-Ack retry counter to 0
A5	Stop T103 Timer	Stop timer T103
A6	Increment I-SSIE retry	Increment the counter I-SSIE-Retry-Count
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 counts	Set FragNb=1
A10	Send Ack MEC=1, RSAC =0, SeqNb=SIE2Frag TotalNb	Send a group of acknowledgement cells with MEC=1, RSAC=0, and SeqNb=SIE2FragTotalNb from initiator to responder
A11	Stop T104 Timer	Stop timer T104
A12	Send Ack MEC=0, RSAC =0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RSAC =0, and SeqNb=FragNb from initiator to responder
A13	Increment I-Ack	Increment I-Ack counter
A14	Increment FragNb	Increment FragNb counter
A15	Send Ack MEC=1, RSAC=1, SeqNb=SIE2Frag TotalNb	Send a group of acknowledgement cells with MEC=1, RSAC=1, and SeqNb=SIE2FragTotalNb from initiator to responder
A16	OK for sending SAC	It is now recommended that the initiator sends a group of SAC cells to the responder
A17	Send Ack MEC=0, RSAC=0, SeqNb=0	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=0 from initiator to responder to make the responder send the full SIE2 again
A18	Start T105	Start timer T105
A19	Stop T105	Stop timer T105
A20	Initialize WrongFragNb=0	Set WrongFragNb=0
A21	WrongFragNb=SI E2FragNb	Set WrongFragNb=SIE2FragNb
A22	Initialize AckNb=-1	Set AckNb=-1
A23	AckNb=SeqNb	Set AckNb=SeqNb
	RESPONDER	
Number	**Name**	**Description**

A1	Start T104 Timer	Start timer T104
A2	Initialize FragNb count	Set FragNb to 1
A3	Initialize R-SSIE retry count	Set R-SSIE retry counter to 0
A4	Initialize R-Ack retry count	Set R-Ack retry counter to 0
A5	Send Ack MEC=0, RSAC=0, SeqNb=1	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=1 from responder to initiator
A6	Stop T102 Timer	Stop timer T102
A7	Send Ack MEC=0, RSAC=0, SeqNb=SIE1FragTotalNb	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=SIE1FragTotalNb from responder to initiator
A8	Send Ack MEC=0, RSAC=0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=FragNb-1 from responder to initiator
A9	Increment R-Ack	Increment R-Ack counter
A10	Increment FragNb	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 SAC	It is now recommended that the responder sends a group of SAC cells to the initiator
A17	Send Ack MEC=0, RSAC=0, SeqNb=0	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=0 from responder to initiator to make the initiator send the full SIE1 again
A18	Start T105	Start timer T105
A19	Stop T105	Stop timer T105
A20	Initialize WrongFragNb=0	Set WrongFragNb=0
A21	WrongFragNb=SIE1FragNb	Set WrongFragNb=SIE1FragNb
A22	Initialize AckNb=-1	Set AckNb=-1
A23	AckNb=SeqNb	Set AckNb=SeqNb

9.5 FSM Summary Table

Table 7: Initiator Summary.

States x Events:	**S1-I**	**S2-I**	**S3-I**	**S4-I**	**S5-I**	**S6-I**	
E1	A1, A2, A3, A4, A22 S2-I	N/A	N/A	N/A	N/A	N/A	
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	N/A	
E6	N/A	N/A	A19, A8, A9, A20 S4-I	N/A	N/A	N/A	
E7	N/A	N/A	A10, A18 S5-I	N/A	N/A	N/A	
E8	N/A	N/A	N/A	A11,A10 S5-I	N/A	N/A	
E9	N/A	N/A	A19, A17, A13, A8 S3-I	A12, A8, A13 S4-I	N/A	N/A	
E10	N/A	N/A	S6-I	S6-I	N/A	N/A	
E11	N/A	N/A	N/A	A14, A8 S4-I	N/A	N/A	
E12	N/A	N/A	N/A	N/A	A15, A16 S1-I	N/A	
E13	N/A	N/A	A19, A17, A13, A8, A21 S3-I	A12, A8, A13, A21 S4-I	N/A	N/A	
E14	N/A	N/A	S6-I	N/A	N/A	N/A	

Table 8: Responder Summary.

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

9.6 Renegotiation Finite State Machines (FSMs) when the three-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. The FSM Graphical Views are shown in Section 9.6.1.
7. All FSM States are described in section 9.6.2.
8. All FSM Events are described in section 9.6.3.
9. All FSM Actions are described in section 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.
- SIE1FragTotalNb, SIE2FragTotalNb, SIE3FragTotalNb are the number of fragments of SIE1, SIE2, and SIE3, that is the number of negotiation cells necessary to transport the SSIE.
- SIE2FragNb, and SIEFragNb are the number indicated respectively in the SIE2, and either the SIE1 or SIE3 fragment received.
- WrongFragNb is the number of the first unexpected fragment (with a wrong number) belonging to the same SIE.
- 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.

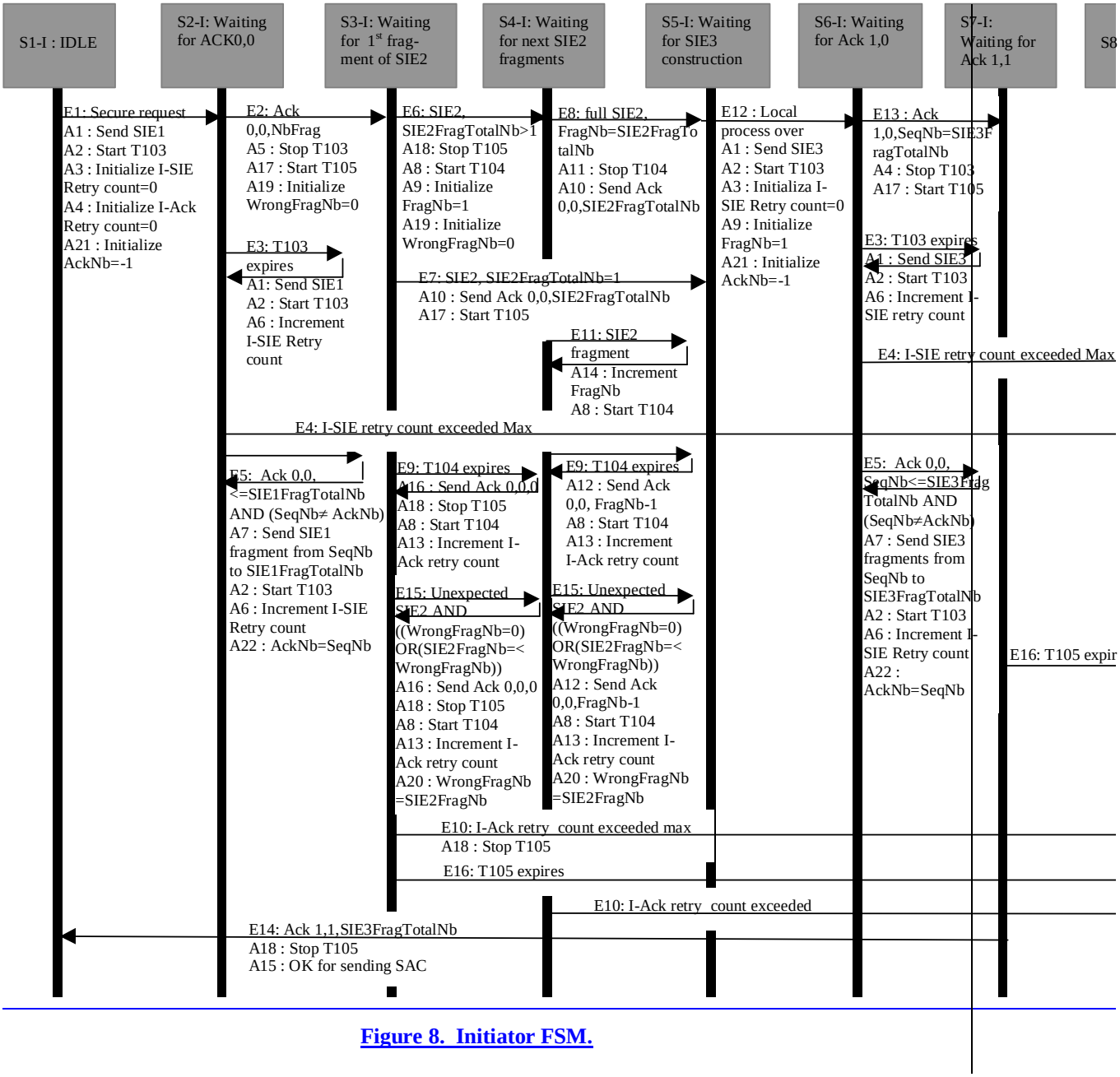

Figure 8. Initiator FSM.

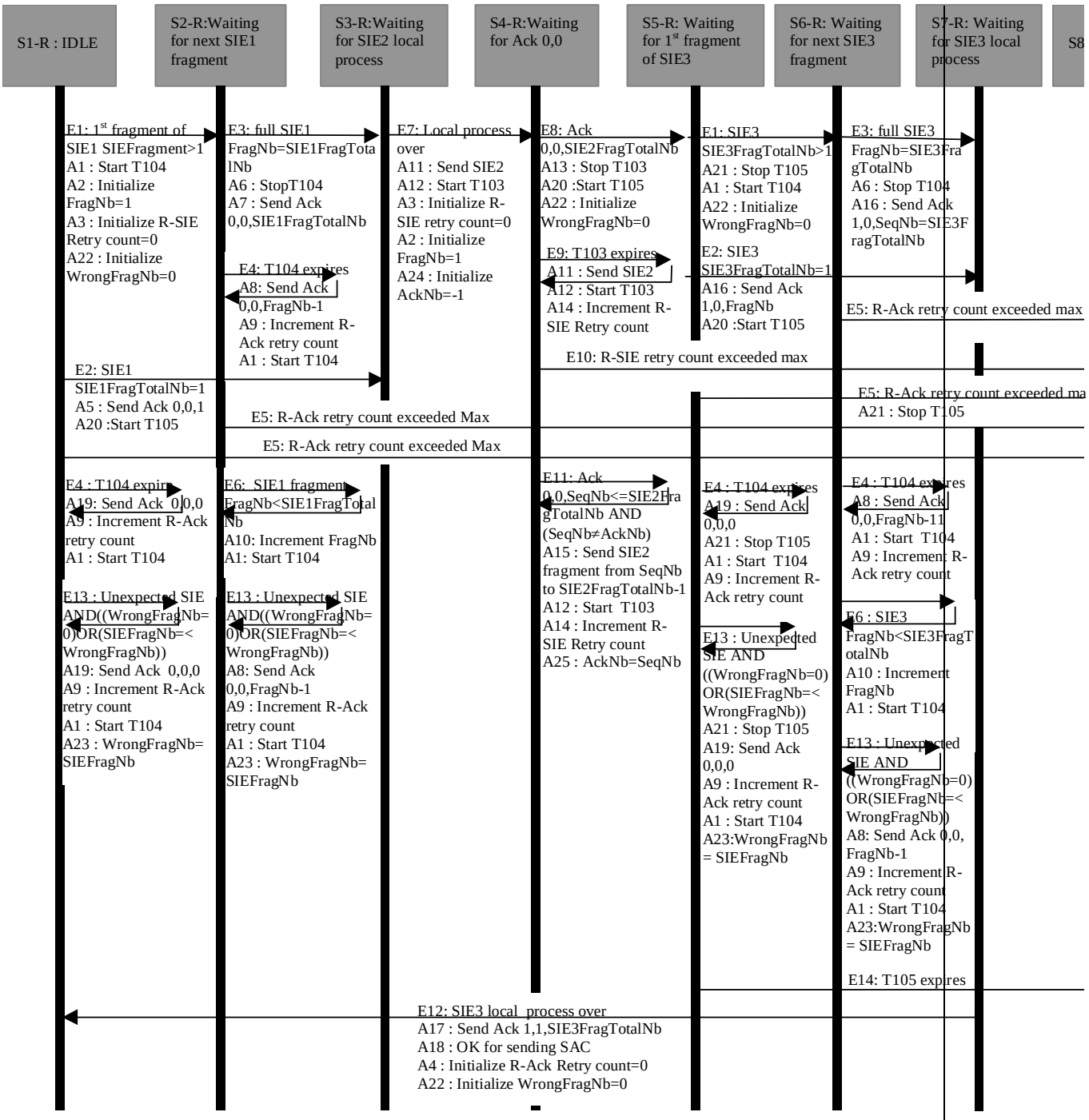

Figure 9. Responder FSM.

9.6.2 FSM States

INITIATOR	***-I***		
Number	**Name**	**Messages Outstanding**	**Description**
S1–I	Idle	None, or SAC cells	A new security parameters negotiation is required

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

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

9.6.3 FSM Events

Table 9: Events

	INITIATOR	
Number	**Name**	**Description**
E1	Security Association Update Request	The initiator has been requested to negotiate a new security association

E2	Valid Ack (MEC=0, RSAC=0, SeqNb=SIEFragTotalNb) 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	I-SSIE-Retry-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<=SIEFragTotalNb) Received AND (SeqNb≠AckNb)	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. • this event results in the new transmission of the SSIE fragments numbered from SeqNb
E6	Valid SIE2 first fragment Received (SIE2FragTotalNb>1)	The initiator receives the first fragment of the SIE2 • this event results in new fragments being expected
E7	Valid SIE2 fully Received (SIE2FragTotalNb=1)	The initiator receives the full SIE2 in one fragment • this event results in no more fragments being expected
E8	Valid SIE2 fully Received (FragNb=SIE2FragTotalNb)	The initiator receives the full SIE2 in SIE2FragTotalNb fragments • this event is expected if the SIE2 requires more than one fragment.
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	The initiator receives one more fragment • this event is expected if the SIE2 requires more than one fragment.
E12	Local construction of SIE3 over	The initiator processes SIE2 and constructs the SIE3 • this event is expected.
E13	Valid Ack (MEC=1, RSAC=0, SeqNb=SIEFragTotalNb) Received	Acknowledgement cells have been received • this event is expected
E14	Valid Ack (MEC=1, RSAC=1, SeqNb=SIEFragTotalNb) 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(SIE2FragNb=<WrongFragNb))	• This event results in sending a group of Acknowledgement cells MEC=0, RSAC=0, SeqNb=0
E16	T105 expires	Timer T105 has exceeded time shown in section 5.5.3.1.. • this event results in a fault at the initiator
	RESPONDER	
Number	**Name**	**Description**
E1	Valid SSIE first fragment Received (SIEFragTotalNb >1)	The responder receives the first fragment of the SSIE • this event results in new fragments being expected

E2	Valid SSIE received (SIEFragTotalNb =1)	The responder receives the full SSIE in one fragment • this event results in no more fragments being expected
E3	Valid SSIE fully received (FragNb=SIEFrag TotalNb)	The responder receives the full SSIE in SIEFragTotalNb fragments • this event is expected if the SSIE requires more than one fragment.
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 Count Exceeded	Responder 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
E6	One SSIE fragment received	Responder receives one more fragment • this event is expected if the SSIE requires more than one fragment.
E7	Local construction of SIE2	Responder processes SIE1 and constructs the SIE2 • this event is expected.
E8	Valid Ack (MEC=0, RSAC=0, SeqNb=SIE2Frag TotalNb) Received	Acknowledgement cells have been received • this event is expected
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 Count Exceeded	Initiator has sent SSIE of FLOW2 the maximum number of times. • This event results in a fault at the initiator
E11	Valid Ack (MEC=0, RSAC =0, SeqNb<=SIE2Fra gTotalNb) Received AND (SeqNb≠AckNb)	One acknowledgement cell has been received but indicates that cells were lost. If a group of similar acknowledgement cells is received, event E11 occurs only once when receiving the first acknowledgement cell. • this event results in the new transmission of the SIE2 fragments numbered from SeqNb
E12	Local process of SIE3 over	Responder processes SIE3 locally and is ready for security association changeover • this event is expected
E13	Unexpected SIE fragment received	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

9.6.4 FSM Actions

Table 10: Actions

	INITIATOR	
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 retry counter	Set I-SSIE retry counter to 0
A4	Initialize I-Ack retry counter	Set I-Ack retry counter to 0
A5	Stop T103 Timer	Stop timer T103
A6	Increment I-SSIE retry	Increment the counter I-SSIE-Retry-Count
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 counts	Set FragNb=1
A10	Send Ack MEC=0, RSAC=0, SeqNb=SIE2FragTotalNb	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=SIE2FragTotalNb from initiator to responder
A11	Stop T104 Timer	Stop timer T104
A12	Send Ack MEC=0, RSAC=0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=FragNb from initiator to responder
A13	Increment I-Ack	Increment I-Ack counter
A14	Increment FragNb	Increment FragNb counter
A15	OK for sending SAC	It is now recommended that the initiator sends a group of SAC cells to the responder
A16	Send Ack MEC=0, RSAC=0, SeqNb=0	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=0 from initiator to responder to make the responder send the full SIE2 again
A17	Start T105	Start timer T105
A18	Stop T105	Stop timer T105
A19	Initialize WrongFragNb=0	Set WrongFragNb=0
A20	WrongFragNb=SIE2FragNb	Set WrongFragNb=SIE2FragNb
A21	Initialize AckNb=-1	Set AckNb=-1
A22	AckNb=SeqNb	Set AckNb=SeqNb
	RESPONDER	
Number	**Name**	**Description**
A1	Start T104 Timer	Start timer T104

A2	Initialize FragNb	Set FragNb to 1
A3	Initialize R-SSIE retry counter	Set R-SSIE retry counter to 0
A4	Initialize R-Ack retry counter	Set R-Ack retry counter to 0
A5	Send Ack MEC=0, RSAC=0, SeqNb=1	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=1 from responder to initiator
A6	Stop T104 Timer	Stop timer T104
A7	Send Ack MEC=0, RSAC=0, SeqNb=SIE1FragTotalNb	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=SIE1FragTotalNb from responder to initiator
A8	Send Ack MEC=0, RSAC=0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=FragNb-1 from responder to initiator
A9	Increment R-Ack	Increment R-Ack counter
A10	Increment FragNb	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	Send Ack MEC=1, RSAC=0, SeqNb=SIEFragTotalNb	Send a group of acknowledgement cells with MEC=1, RSAC=0, and SeqNb=SIEFragTotalNb from responder to initiator
A17	Send Ack MEC=1, RSAC=1, SeqNb=SIEFragTotalNb	Send a group of acknowledgement cells with MEC=1, RSAC=1, and SeqNb=SIEFragTotalNb from responder to initiator
A18	OK for sending SAC	It is now recommended that the responder sends a group of SAC cells to the initiator
A19	Send Ack MEC=0, RSAC=0, SeqNb=0	Send a group of acknowledgement cells with MEC=0, RSAC=0, and SeqNb=0 from responder to initiator to make the initiator send the full SIE again
A20	Start T105	Start timer T105
A21	Stop T105	Stop timer T105
A22	Initialize WrongFragNb=0	Set WrongFragNb=0
A23	WrongFragNb=SIEFragNb	Set WrongFragNb=SIEFragNb (either SIE1FragNb or SIE3FragNb)
A24	Initialize AckNb=-1	Set AckNb=-1
A25	AckNb=SeqNb	Set AckNb=SeqNb

9.6.5 FSM Summary Table

Table 11: Initiator Summary.

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

Table 12: Responder Summary.

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

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

Table 18: Security Message Identifiers

Codepoint	Reference	Size of Length field	Use of Value field	Name
0010 0001*	5.1.5.3.2.1	2	Complex (SSIE)	FLOW1-3WE
0010 0010*	5.1.5.3.2.1	2	Complex (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	–	87 ~~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

Codepoint	Reference	Size of Length field	Use of Value field	Name
1001 0110	7.2.2.6	–	1 of 3 values	Access Control Service Options
1001 0111	7.2.2.7	–	1 of 3 values	Certificate Exchange Service Options
1001 1000	7.2.2.8	–	1 of 3 values	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	7.2.3.7	1	Complex: Algorithms for: Signature Hash	Authentication Algorithm Group
1010 1110	7.2.3.8	1	Complex: Algorithms for: MAC Signature Key Exchange Key Update Hash	Integrity Algorithm Group
1011 0000	7.2.3.9	1	Complex: Algorithms for: Encryption Signature Key Exchange Key Update Hash	Confidentiality Algorithm Group
1011 0010	7.2.3.7/ 7.2.3.8/ 7.2.3.9	1	Algorithm details	Signature Algorithm Details
1011 0100	7.2.3.7/ 7.2.3.8	1	Algorithm details	Hash Algorithm Details
1011 0110	7.2.3.8	1	Algorithm details	MAC Algorithm details
1011 1000	7.2.3.8 7.2.3.9	1	Algorithm details	Key Exchange Algorithm Details
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: (Next 3): Master Key 1st data Conf Session key 1st data Integ Session key	Confidential Parameters
1100 1000	7.3.2	1	value	Master Key
1100 1010	7.3.3	1	value	First Data Confidentiality Session Key
1100 1100	7.3.4	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).