ATM Forum Technical Committee
ATM Forum/99-XXXX

****

TITLE: Security services negotiation through OAM cells

****

SOURCE:

Maryline Laurent
ENST de Bretagne
2, rue de la châtaigneraie
35512 Cesson Sévigné, France
Phone: 33 2 99 12 70 20
Fax: 33 2 99 12 70 30
E-mail: Maryline.Laurent@enst-bretagne.fr

Thomas D. Tarman[*]
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0451
Phone: (505)844-4975
Fax: (505)844-9641
E-mail: tdtarma@sandia.gov

****

DATE: May 2000, San Francisco, California

****

DISTRIBUTION: SEC Working Group

****

ABSTRACT:

As described in contribution AF99-0335, it is interesting that new security services and mechanisms are allowed to be negotiated during a connection in progress. To do that, new "negotiation OAM cells" dedicated to security should be defined, as well as some acknowledgment cells allowing negotiation OAM cells to be exchanged reliably. Remarks which were given at the New Orleans meeting regarding those cells formats are taken into account.

This contribution presents some baseline text describing the format of the negotiation and acknowledgment cells, and the using of those cells. All the modifications brought to the specifications 1.0 are visible using the Word tools.

****

NOTICE:

This contribution has been prepared to assist the ATM Forum. This proposal is made by the authors as a basis of discussion. This contribution should not be construed as a binding proposal on the authors or their companies. Specifically, the authors and their companies reserve the right to amend or modify the statements contained herein.

****

Introduction

The following sections in this contribution provide proposed modifications (indicated in bold face) to the Security 1.0 baseline document, af-sec-0100.01. All section references in this contribution pertain to the Security 1.0 document.

[*] This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy.

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.

In order to negotiate parameters for the security services described in Section 3, and to directly support the entity authentication service described in Section 3.1, a set of security message exchange protocols is required. These protocols are described in Section 5.1 and are summarized below.

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

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

This specification defines **three** mechanisms for transporting security information. These are the signaling-based security message exchange mechanism (described in Section 5.1.2), the in-band security message exchange mechanism (described in Section 5.1.5)**, and the management-based security message exchange mechanism (described in Section** 5.4). In **all** cases, the Security Services Information Element (described in Section 5.1.5.3.6) is used to carry the security information.

The method for performing the two-way exchange protocol in Signaling 4.0 flows is described in detail in Section 5.1.2. (The three-way message exchange protocol is not supported in Signaling 4.0 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 because 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 two-way and three-way message exchange protocols in the user plane VCC/VPC is described in detail in Section 5.1.5. This method applies to both SVCs and PVCs. In order to provide a reliable transport service for inband message flows, an inband 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 5.4. This method also applies to SVCs and PVCs, and requires a specific cell-loss recovery protocol which is presented in Section** 5.4..

PVCs (permanent virtual connections) are provisioned connections. Security services negotiation, authentication, certificate exchange, and key exchange can be done via provisioning at the time PVCs are established, or inband as described in Sections 5.1.4.4 **and** 5.4. 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, confidentiality and/or integrity (data origin authentication) keys need to be exchanged in order to provide these services. In order to prevent «man in the middle» attacks, key exchange must be accompanied by strong authentication between the user endpoints, as addressed in Sections 5.1.1.1 and 5.1.1.2. Other details associated with the key exchange service are described in Section 6.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Once confidentiality and/or integrity keys are exchanged, session keys for these services must be changed periodically. Section 5.3 describes the key update mechanism (for the data confidentiality and data integrity services), which is performed **first** by exchanging a «master key» at the connection setup time **or during the connection** and **second by** exchanging subsequent «session» keys under the master key. Session key update messages are exchanged using OAM cells.

During a connection in progress, possibility is given to users to negotiate new security services during the connection, using the security message exchanges. Section 5.4 describes the renegotiation mechanism. The security message exchanges are done using OAM cells.

Section REF5.5 describes the certification infrastructure and mechanisms for transporting certificates and certificate chains. 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).

5.3.1. Security OAM Cells

5.3.1.1.1. 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 1.

Table 1: 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

5.3.1.1.1.1. Data Confidentiality SKE OAM Cell Format

The format of the Data Confidentiality SKE OAM Cell is defined in Figure 1.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Byte #	Descrip
GFC/VPI[11:8]				VPI[7:4]				1	ATM addr
VPI[3:0]				VCI[15:12]				2	ATM addr
VCI[11:4]								3	ATM Addr
VCI[3:0]				PTI			CLP	4	Addr,PTI
HEC[7:0]								5	Header Ck
1	1	1	1	0	0	0	1	6	OAM+Ftype
Relative ID				0	0	0	1	7	RID,FID
Bank ID				Reserved				8	BID,RES
Reserved								9	Reserved
Key Number								10-13	KN
Encrypted Session Key (ESK)								14-45	ESK
Reserved								46-51	Reserved
0	0	0	0	0	0	CRC[9:8]		52	0 , CRC
CRC-10 [7:0]								53	CRC-10

Figure 1. Session Key Exchange (SKE) 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 key updates.

3) The key number (KN) field, a 32-bit field, indicates the key number associated with the new session key. Each session key is assigned a number. The first session key to be exchanged by the session key update protocol is assigned 1; the second session key is assigned 2, and so forth. The KN is a cryptographically protected field which is used by the key update protocol to ensure freshness, uniqueness, and ordering of session key updates and to synchronize the initiator and responder to the same session key.

4) The Encrypted Session Key (ESK), a 256-bit field, contains the next session key encrypted using the master key for the connection. For cases when the session key being transported is less than 256 bits in length, it is contained in the least significant bits of the ESK field, with the most significant bits being padded with zeros (before encryption).

5) Reserved bytes are set to 6A hex, reserved bit fields less than 1 byte in length are set to all zeros.

6) The reserved bits included after the 4-bit Bank ID field are provided so that the KN and ESK fields are aligned on 16 bit boundaries, to simplify high speed implementations.

5.3.1.1.1.2. Data Integrity SKE OAM Cell Format

The format of the Data Integrity SKE OAM Cell is defined in Figure 1, except that the Function ID field is set to 0010 (binary).

5.3.1.1.1.3. Negotiation OAM cell Format

The format of the Negotiation OAM cell is defined in Figure 2.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Byte #	Descrip
GFC/VPI[11:8]				VPI[7:4]				1	ATM addr
VPI[3:0]				VCI[15:12]				2	ATM addr
VCI[11:4]								3	ATM Addr
VCI[3:0]				PTI			CLP	4	Addr,PTI
HEC[7:0]								5	Header Ck
1	1	1	1	0	0	0	1	6	OAM+Ftype
Relative ID				0	1	0	0	7	RID,FID
0	0	0	0	Flow Number				8	OAM+FNb
Sequence Number								9	Sqce Nb
SSIE Fragment								10-51	SSIE Frag.
0	0	0	0	0	0	CRC[9:8]		52	0 , CRC
CRC-10 [7:0]								53	CRC-10

Figure 2. Negotiation OAM Cell Format.

7) The use of the Relative ID field is defined in Section 5.1.7.3.

8) 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).

9) 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.

10) The SSIE Fragment field contains one of the 42-byte fragments composing the SSIE.

5.3.1.1.1.4. Acknowledgment OAM cell Format

The format of the Acknowledgment OAM cell is defined in Figure 3.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Byte #	Descrip
GFC/VPI[11:8]				VPI[7:4]				1	ATM addr
VPI[3:0]				VCI[15:12]				2	ATM addr
VCI[11:4]								3	ATM Addr
VCI[3:0]				PTI			CLP	4	Addr,PTI
HEC[7:0]								5	Header Ck
1	1	1	1	0	0	0	1	6	OAM+Ftype
Relative ID				0	0	1	1	7	RID,FID
0	0	0	0	Flow Number				8	OAM+FNb
Sequence Number								9	Sqce Nb
0	0	0	0	0	0	RKC	MEC	10	Reserved RKC, MEC
0	0	0	0	0	0	0	0	11-51	Reserved
0	0	0	0	0	0	CRC[9:8]		52	0 , CRC
CRC-10 [7:0]								53	CRC-10

Figure 3. Acknowledgment OAM Cell Format.

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 negotiation OAM cell which is acknowledged.**

4) **The RKC (Ready for Key Changeover) bit set to 1 indicates that the session keys 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 SME flow should be expected (MEC=1) or that another SME flow will come (MEC= 0).**

5.3.1.1.2. 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 2.

Table 2: 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

5.3.1.1.2.1. Data Confidentiality SKC OAM Cell Format

The format of the Data Confidentiality SKC OAM Cell is defined in **Figure 6.**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Byte #	Descrip
GFC/VPI[11:8]				VPI[7:4]				1	ATM addr
VPI[3:0]				VCI[15:12]				2	ATM addr
VCI[11:4]								3	ATM Addr
VCI[3:0]				PTI			CLP	4	Addr,PTI
HEC[7:0]								5	Header Ck
1	1	1	1	0	0	1	0	6	OAM+F type
Relative ID				0	0	0	1	7	RID,FID
Bank ID				Reserved				8	BID,RES
Reserved								9	Reserved
Key Number								10-13	KN
State Vector (SV)								14-21	SV
Reserved								22-51	Reserved
0	0	0	0	0	0	CRC[9:8]		52	0 , CRC
CRC-10 [7:0]								53	CRC-10

Figure 4. Session Key Changeover (SKC) 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 key updates.

3) The key number (KN) field, a 32-bit field, indicates the key number associated with the new session key. Each session key is assigned a number. The first session key to be exchanged by the session key update protocol is assigned 1; the second session key is assigned 2, and so forth.

4) The State Vector (SV) contains the new state vector when using the counter mode of the Data Confidentiality service, or it is set to all zeros otherwise.

5) Reserved bytes are set to 6A hex, reserved bit fields less than 1 byte in length are set to all zeros.

6) The reserved bits included after the 4-bit Bank ID field are provided so that the KN and SV fields are aligned on 16 bit boundaries, to simplify high speed implementations.

5.3.1.1.2.2. Data Integrity SKC OAM Cell Format

The format of the Data Integrity SKC OAM Cell is defined in Figure 4REFMERGEFORMAT, except that the Function ID field is set to 0010 (binary).

5.3.1.1.2.3. Security Association Changeover SAC OAM Cell Format

The format of the Security Association SAC OAM Cell is defined in Figure 5.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Byte #	Descrip
GFC/VPI[11:8]				VPI[7:4]				1	ATM addr
VPI[3:0]				VCI[15:12]				2	ATM addr
VCI[11:4]								3	ATM Addr
VCI[3:0]				PTI			CLP	4	Addr,PTI
HEC[7:0]								5	Header Ck
1	1	1	1	0	0	1	0	6	OAM+F type
Relative ID				0	0	1	1	7	RID,FID
Bank ID				Reserved				8	BID,RES
Reserved								9	Reserved
Confidentiality Key Number								10-13	CKN
Integrity Key Number								14-17	IKN
Reserved								18-51	Reserved
0	0	0	0	0	0	CRC[9:8]		52	0 , CRC
CRC-10 [7:0]								53	CRC-10

Figure 5. 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 and key updates.**

3) **The Confidentiality key number (KN) field, a 32-bit field, indicates the key number associated with the new data confidentiality session key. Each session key is assigned a number. The first session key to be exchanged by the session key update protocol is assigned 1; the second session key is assigned 2, and so forth. If confidentiality session keys are not required over the connection, the CKN field is set to all zeros.**

4) **The Integrity key number (KN) field, a 32-bit field, indicates the key number associated with the new data integrity session key. Each session key is assigned a number. The first session key to be exchanged by the session key update protocol is assigned 1; the second session key is assigned 2, and so forth. If integrity session keys are not required over the connection, the IKN field is set to all zeros.**

5) **Reserved bytes are set to 6A hex, reserved bit fields less than 1 byte in length are set to all zeros.**

6) **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.**

5.4. In-band Security Association Renegotiation

When the initiator (or responder) wants to modify the security association used to protect its ATM traffic exchanges, it sends some negotiation OAM cells within the user data stream to negotiate new security parameters with the remote partner. Up to three exchanges of negotiation OAM cells are allowed between partners to negotiate a new security association. Each negotiation OAM cells transfer reception is acknowledged by the receiving partner which sends back some 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 5.3.1.1.1.3and 5.3.1.1.1.4.

Once the negotiation cells exchanges are done and the negotiation is over, each partner indicates to the other when to start using the new security association by sending some Security Association Changeover SAC cells whose format is given in Section 5.3.1.1.2.3.

In order to change security association at high speeds, without disrupting service to the end user, two security associations are required: a current-association and a next-association. The next-association is delivered using the negotiation cells. The negotiation cells receiver stores the next-association in a separate memory location until needed. The actual changeover occurs when the SAC cell is sent.

It is likely that collisions between negotiations, and between renegotiation and session key update occur. The solution is that each partner has a collision manager which:

- precludes that a session key update be initiated locally when a security association negotiation is in progress.
- stops the session key update currently in progress when it receives a security association negotiation request from its partner, provided that it did not send SKC session key activation cell yet. That is, if the session key update has just been initiated sending only SKE cells, the collision manager stops the session key update mechanism to realize the requested negotiation. If at least one SKC cell has been sent, the session key update is completed and the negotiation is processed.
- stops the negotiation being initiated when a negotiation request arrives. That is, if no negotiation OAM cells have been sent, the collision manager stops the security association negotiation being locally initiated to satisfy the incoming negotiation request. Otherwise, if the connection is an SVC connection, it should stop the negotiation if the ATM address of the station where it is located is greater than this of the connection partner, otherwise the negotiation goes on. If the connection is a PVC connection, it should wait for a randomly chosen duration 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.

5.4.1. The Renegotiation Process

The renegotiation OAM cells are used to transfer security parameters for negotiating the next-association. Renegotiation is done 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 renegotiation OAM cells has 42-byte available in its SSIE Fragment, so that the SSIE is fragmented into 42-byte blocks, before being encapsulated into the negotiation OAM cells. Note that with a 1-byte Sequence Number, the SSIE maximum length is 255*42 = 10710 bytes.

5.3.1.1. Negotiation Processing at the Negotiation Update Initiator (1st 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 42-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 5.3.1.1.1.3. 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.

The NUI increments its confidentiality key number (CKNnui) and/or its integrity key number (IKNnui) by one, and it is assigned to its new data confidentiality session key and/or data integrity session key which have been sent through the negotiation cells. These CKNnui and/or IKNnui numbers will be used when activating the new association through the SAC cell.

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

5.3.1.2. Negotiation Cells Processing at the Negotiation Update Responder (1st 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:

Verifies the 10-bit OAM cell CRC is correct, and discards the cell if it is not.

Reassembles the OAM cells to retrieve the first SSIE.

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 NUR should send some acknowledgment cells to cause a new SSIE transmission.

Stores the new decrypting (confidentiality and/or integrity) session key(s) in memory, until the corresponding SAC cell is received.

5.3.1.3. Negotiation Cells Processing at the Negotiation Update Responder (2nd flow)

The NUR constructs the SSIE of flow 2-2WE or 2-3WE ; it assigns a confidentiality key number CKNnur and/or an integrity key number IKNnur to the new data confidentiality/integrity session key(s) by incrementing the old key numbers by one ; it fragments the SSIE into 42-byte blocks ; it encapsulates them into the negotiation OAM cells; it sends them over the network.

5.3.1.4. Negotiation Processing at the Negotiation Update Initiator (2nd flow)

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

Verifies the 10-bit OAM cell CRC is correct, and discards the cell if it is not.

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 (T103) or the timer of the NUI (T104) expires, the NUI should send some acknowledgment cells to cause a new SSIE transmission.

Reassembles the OAM cells to retrieve the second SSIE.

Stores the new decrypting (confidentiality and/or integrity) session key(s) in memory, until the corresponding SAC cell is received.

If a 2-way SME protocol is selected, and the SSIE is fully received, a group of acknowledgment cells with MEC=1 and RKC=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 RKC=1 are sent.

5.3.1.5. Negotiation Processing at the Negotiation Update Initiator (3rd flow)

If the 3-way SME protocol is selected, the NUI constructs a SSIE ; it fragments the SSIE into 42-byte blocks and encapsulates them into the negotiation cells.

The NUI increments its confidentiality key number (CKNnui) and/or integrity key number (IKNnui) by one, and it is assigned to its new encrypting data confidentiality/integrity session key sent through the negotiation cells. This CKNnui/IKNnui number will be used when activating the new association through the SAC cell.

5.3.1.6. Negotiation Processing at the Negotiation Update Responder (3^{rd} flow)

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

Verifies the 10-bit OAM cell CRC is correct, and discards the cell if it is not.

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.

Reassembles the OAM cells to retrieve the third SSIE.

Stores the new decrypting (confidentiality and/or integrity) session key(s) in memory, until the corresponding SAC cell is received.

If the SSIE is fully received, a group of acknowledgment cells with MEC=1 and RKC=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 RKC=1 are sent.

5.3.1.7. The Acknowledgment Process

Acknowledgment cells are used to acknowledge or disacknowledge 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. That is, acknowledgment cells include the sequence number of the next negotiation OAM cells expected.

5.3.1.7.1. Acknowledgment Processing at the Acknowledgment Cell sender

The acknowledgment 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 for a long time (see the timers definition in section 5.3.1.1) 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 RKC=0.
- The SSIE of the last flow is fully processed. This is realized by sending a group of acknowledgment cells with MEC=1 and RKC=1.

The loss of acknowledgment cells is recovered thanks to one partner timer expiration. To avoid waiting for one timer to expire, and improve the delay for negotiation, the acknowledgment cell sender shall transmit the acknowledgment cell at least three (3) times for L_{Ack} seconds. The value of L_{Ack} is an implementation parameter and need not be standardized.

5.3.1.7.2. Acknowledgment Processing at the Acknowledgment Cell receiver

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

Verifies that the Flow Number included into the acknowledgment cell is the same as the negotiation OAM cells being sent and discards the cell if it is not.

Checks the Sequence Number against the sequence number of the last negotiation OAM cell sent. If the Acknowledgment Sequence Number is equal to the sequence number of the last negotiation OAM cell sent, the transmission of the SSIE is correct. If it is 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, then an error has occurred.

5.4.2. 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 correctly the receiving traffic. The format of the SAC cell shall be as shown in Section 5.3.1.1.2.3.

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 carries the key number(s) associated with the new (confidentiality and/or integrity) session key(s) to which the SAC cell receiver shall switch. The SAC OAM cell is sent multiple times to guarantee receipt at the receiver in the presence of cell loss. The SAC OAM cell is not cryptographically protected (since it does not carry any confidential information).

5.3.1.1. 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 the acknowledgement cells with MEC=1 and RKC=1 before sending the corresponding SAC cell. This provides the remote partner with sufficient time to complete the SSIE processing.

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 some acknowledgement cells with MEC=1 and RKC=1 before sending the corresponding SAC cell.

The sender sets the *CKN* and *IKN* of the SAC cell to the values stored with its next-association (see Section 5.3.1.1.2.3).

The sender injects the SAC OAM cell into the connection undergoing key 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 session key changeover is successful, the sender shall transmit the SAC OAM cell at least three (3) times for L_{SAC} seconds. The value of L_{SAC} is an implementation parameter and need not be standardized. It is also permissible to insert several SAC cells back-to-back. The value of L_{SAC} is an implementation parameter and need not be standardized.

5.3.1.2. SAC Processing at the SKC Cell receiver

Upon receipt of an SAC OAM cell, the destination extracts the key number and performs the following steps:

The destination verifies that the 10-bit OAM cell CRC is correct, and discards the cell if it is not.

The destination shall process the SAC OAM cell on the connection undergoing key 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.

The destination shall check that the key number(s) is greater than the current key number(s) relative to the confidentiality and/or integrity services, that is either greater by one or more.

5.4.3. Protocol Details

5.3.1.1. 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 5.4.3, and Section TBD.

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 sends the full SSIE. If it has not received acknowledgement cells before the timer expires, then the Initiator resends the full SSIE and starts the timer again. The timer is stopped when one acknowledgement cell is received.
2. T104: This timer is used by one partner to determine whether it needs to resend some 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 some acknowledgement cells and starts the timer again. The timer is stopped when a negotiation OAM cell is received.

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 an SSIE has been sent. An 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. Group 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 an SSIE has been sent. An 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. Group of acknowledgement cells may be sent up to a maximum of R-MAX-ACK-RETRY times.

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 5.4.3,Table 4.

1. I-MAX-SSIE-RETRY: This constant indicates the maximum number of times that the Initiator may resend an SSIE 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 an SSIE 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.

5.3.1.2. 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 3: Timers for Renegotiation

Timer Name	Timer Value
T103	10 seconds
T104	5 seconds

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

Table 4: 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

6. Normative Annex

6.3. Renegotiation Finite State Machines (FSMs) when the two-way SME protocol is used

The Finite State Machines (FSMs) described in this section specify the intended behavior for the in-band Security Association Negotiation 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:

1. **Initiator of security association negotiation,**
2. **Responder to security association negotiation.**

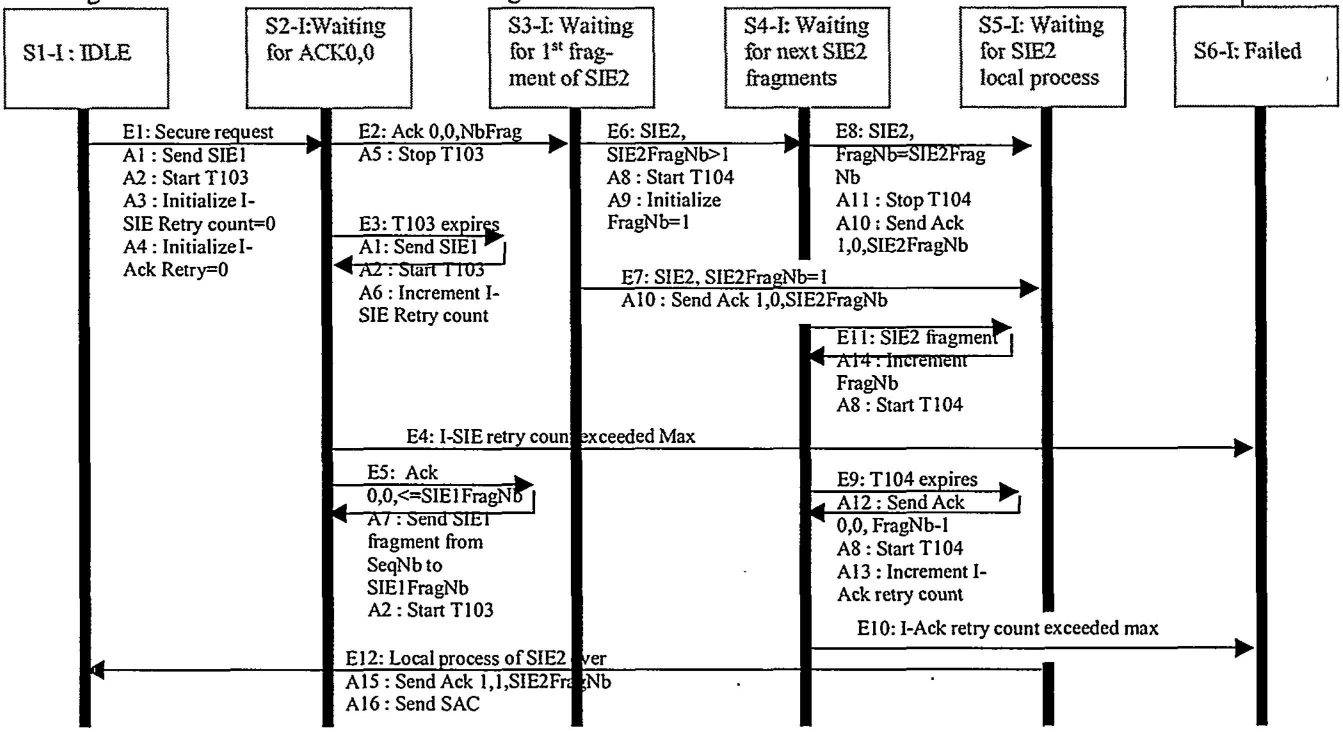
The FSMs are described in five sections:

1. **The FSM Graphical Views are shown in Section 5.3.1.**
2. **All FSM States are described in section 5.3.1.**
3. **All FSM Events are described in section 5.3.1.**
4. **All FSM Actions are described in section 5.3.1.**
5. **The FSM Summary Tables are shown in section 5.3.1.**

5.3.1. FSM Graphical View

The notations used are the following:

- SIE1, SIE2 are the SSIE of the first and second flow.
- SIE1FragNb, SIE2FragNb are the number of fragments of SIE1 and SIE2, that is the number of negotiation cells necessary to transport the SSIE.
- SeqNb is the sequence number indicated in the acknowledgement cells.
- FragNb is the number of the next SSIE fragment to be received.



S1-I : IDLE
S2-I:Waiting for ACK0,0
S3-I: Waiting for 1st frag-ment of SIE2
S4-I: Waiting for next SIE2 fragments
S5-I: Waiting for SIE2 local process
S6-I: Failed
E1: Secure request
A1 : Send SIE1
A2 : Start T103
A3 : Initialize I-SIE Retry count=0
A4 : Initialize I-Ack Retry=0
E2: Ack 0,0,NbFrag
A5 : Stop T103
E6: SIE2, SIE2FragNb>1
A8 : Start T104
A9 : Initialize FragNb=1
E8: SIE2, FragNb=SIE2FragNb
A11 : Stop T104
A10 : Send Ack 1,0,SIE2FragNb
E3: T103 expires
A1: Send SIE1
A2 : Start T103
A6 : Increment I-SIE Retry count
E7: SIE2, SIE2FragNb=1
A10 : Send Ack 1,0,SIE2FragNb
E11: SIE2 fragment
A14 : Increment FragNb
A8 : Start T104
E4: I-SIE retry coun exceeded Max
E5: Ack 0,0,<=SIE1FragNb
A7 : Send SIE1 fragment from SeqNb to SIE1FragNb
A2 : Start T103
E9: T104 expires
A12 : Send Ack 0,0, FragNb-1
A8 : Start T104
A13 : Increment I-Ack retry count
E10: I-Ack retry count exceeded max
E12: Local process of SIE2 ver
A15 : Send Ack 1,1,SIE2Fr gNb
A16 : Send SAC

Figure 6. Initiator FSM.

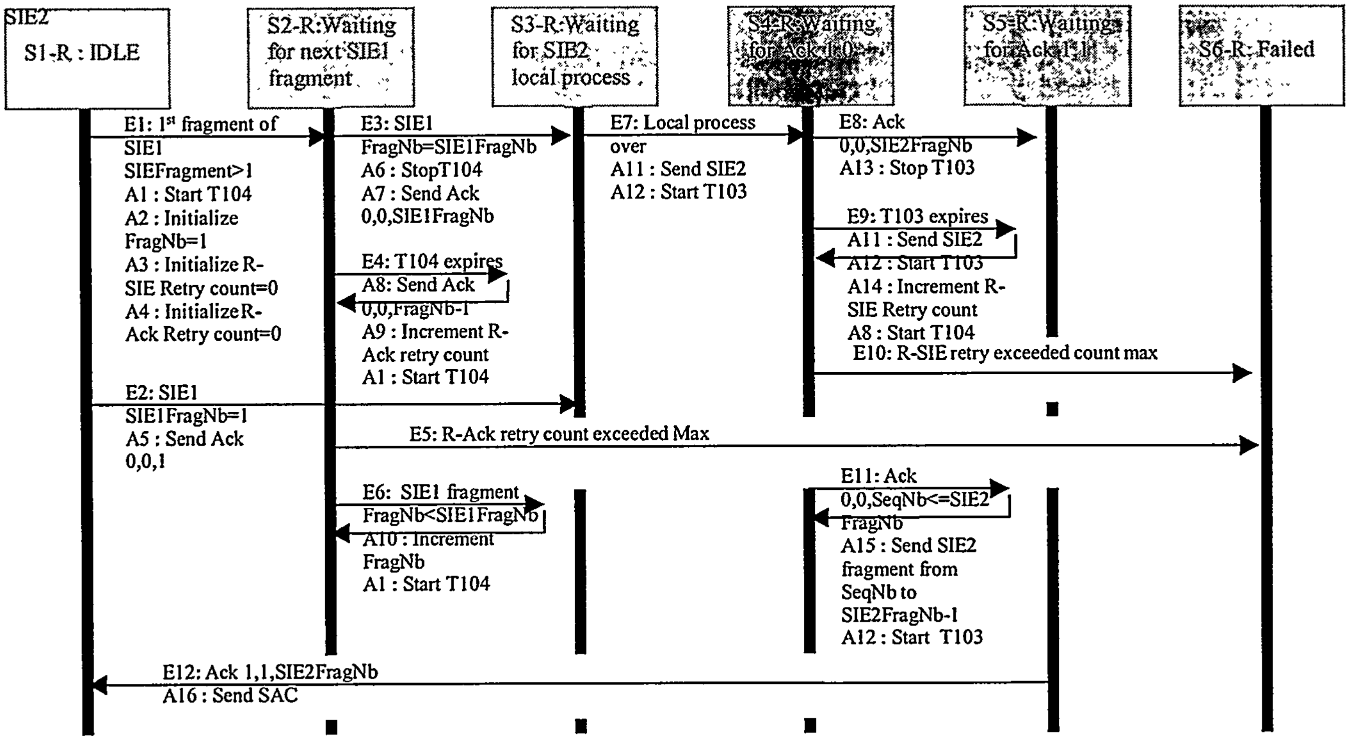


Figure 7. Responder FSM.

5.3.2. FSM States

INITIATOR -I			
Number	Name	Messages Outstanding	Description
S1 –I	Idle	None, or SAC cells + Acknowledgement cells MEC=1, RKC=1	A new security association negotiation is required
S2 –I	Waiting for Ack MEC=0, RKC=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 1^{st} fragment of SIE2	Acknowledgement cells MEC=0, RKC=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, RKC=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 RKC=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 RKC=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.

5.3.3. 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, RKC=0, SeqNb=SIE1FragNb) Received	Acknowledgement cells have been received • this event is expected
E3	T103 expires	Timer T103 has exceeded time shown in section 5.3.1.1 • 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, RKC=0, SeqNb<=SIE1FragNb) Received	Acknowledgement cells have been received but indicates that cells were lost • this event results in the new transmission of the SIE1 fragments numbered from SeqNb
E6	Valid SIE2 first fragment Received (SIE2FragNb>1)	The initiator receives the first fragment of the SIE2 • this event results in new fragments being expected
E7	Valid SIE2 received (SIE2FragNb=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=SIE2FragNb)	The initiator receives the full SIE2 in SIE2FragNb fragments • this event is expected if the SIE2 requires more than one fragment.
E9	T104 expires	Timer T104 has exceeded time shown in section 5.3.1.1 • this event results in Acknowledgement cells MEC=0, RKC=0, SeqNb=FragNb-1
E10	I-Ack Retry Count Exceeded	Initiator has sent a group of acknowledgement cells (MEC=0, RKC=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.
	RESPONDER	
Number	Name	Description
E1	Valid SIE1 first fragment Received (SIE1FragNb>1)	The responder receives the first fragment of the SIE1 • this event results in new fragments being expected
E2	Valid SIE1 Received (SIE1FragNb=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=SIE1FragNb)	The responder receives the full SIE1 in SIE1FragNb.fragments • this event is expected if the SIE1 requires more than one fragment.
E4	T104 Expires	Timer T104 has exceeded time shown in section 5.3.1.1 • 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, RKC=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, RKC=0, SeqNb=SIE2FragNb) Received**	**Acknowledgement cells have been received** • **this event is expected**
E9	**T103 expires**	**Timer T103 has exceeded time shown in section** 5.3.1.1 • **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, RKC=0, SeqNb<=SIE2FragNb) Received**	**Acknowledgement cells have been received but indicates that cells were lost** • **this event results in the new transmission of the SIE2 fragments numbered from SeqNb**
E12	**Invalid Ack (MEC=1, RKC=1, SeqNb=SIE2FragNb) Received**	**Responder receives Acknowledgement cells MEC=1, RKC=1** • **this event is expected**

5.3.4. FSM Actions

Table 6: Actions

	INITIATOR	
Number	Name	Description
A1	Send SIE1	Send a SIE1 encapsulated into FragNbSIE1 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 FragNbSIE1 from initiator to responder
A8	Start T104 Timer	Start timer T104
A9	Initialize FragNb counts	Set FragNb=1
A10	Send Ack MEC=1, RKC=0, SeqNb=FragNbSIE2	Send a group of acknowledgement cells with MEC=1, RKC=0, and SeqNb=FragNbSIE2 from initiator to responder
A11	Stop T104 Timer	Stop timer T104
A12	Send Ack MEC=0, RKC=0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RKC=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, RKC=1, SeqNb=FragNbSIE2	Send a group of acknowledgement cells with MEC=1, RKC=1, and SeqNb=FragNbSIE2 from initiator to responder
A16	Send SAC	Send a group of SAC cells from initiator to responder
	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, RKC=0, SeqNb=1	Send a group of acknowledgement cells with MEC=0, RKC=0, and SeqNb=1 from responder to initiator
A6	Stop T102 Timer	Stop timer T102
A7	Send Ack	Send a group of acknowledgement cells with MEC=0, RKC=0, and

	MEC=0, RKC=0, SeqNb=FragNbSIE1	SeqNb=FragNbSIE1 from responder to initiator
A8	Send Ack MEC=0, RKC=0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RKC=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 FragNbSIE2 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 FragNbSIE2 from responder to initiator
A16	Send SAC	Send a group of SAC cells from responder to initiator

5.3.5. 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 S2-I	N/A	N/A	N/A	N/A	N/A
E2	N/A	A5 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 S2-I	N/A	N/A	N/A	N/A
E6	N/A	N/A	A8, A9 S4-I	N/A	N/A	N/A
E7	N/A	N/A	A10 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	N/A	A12, A8, A13 S4-I	N/A	N/A
E10	N/A	N/A	N/A	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

Table 8: Responder Summary.

States x Events:	S1-R	S2-R	S3-R	S4-R	S5-R	S6-R
E1	A1, A2, A3, A4 S2-R	N/A	N/A	N/A	N/A	N/A
E2	A5 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	N/A	A8, A9, A1 S2-R	N/A	N/A	N/A	N/A
E5	N/A	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 S4-R	N/A	N/A	N/A
E8	N/A	N/A	N/A	A13 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 S4-R	N/A	N/A
E12	N/A	N/A	N/A	N/A	A16 S1-R	N/A

6.4. 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 5.3.1.
7. All FSM States are described in section 5.3.1.
8. All FSM Events are described in section 5.3.1.
9. All FSM Actions are described in section 5.3.1.
10. The FSM Summary Tables are shown in section 5.3.1.

5.3.1. FSM Graphical View

The notations used are the following:

- SIE1, SIE2, SIE3 are the SSIE of the first, second and third flow.
- SIE1FragNb, SIE2FragNb, SIE3FragNb are the number of fragments of SIE1, SIE2, and SIE3, that is the number of negotiation cells necessary to transport the SSIE.
- SeqNb is the sequence number indicated in the acknowledgement cells.
- FragNb is the number of the next SSIE fragment to be received.

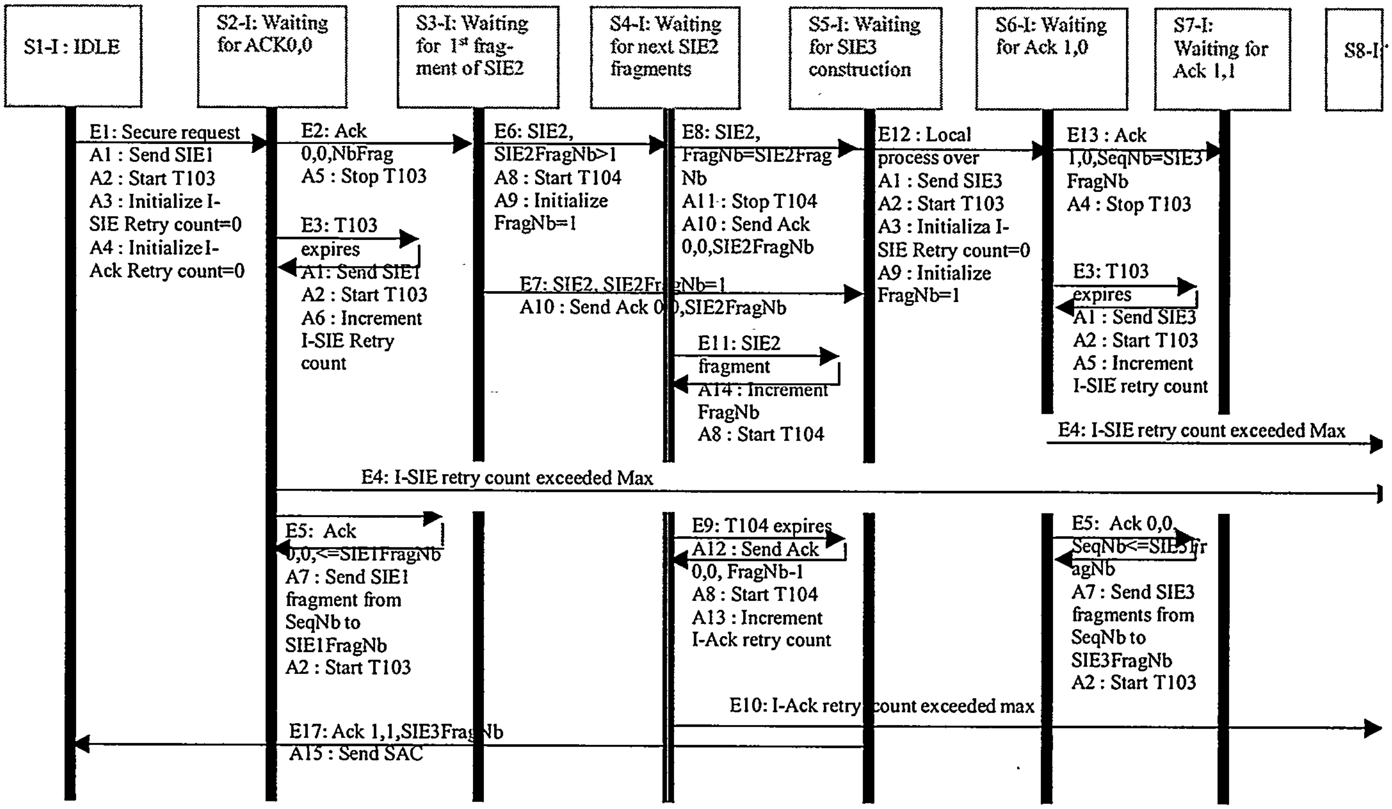

S1-I : IDLE
S2-I: Waiting for ACK0,0
S3-I: Waiting for 1st fragment of SIE2
S4-I: Waiting for next SIE2 fragments
S5-I: Waiting for SIE3 construction
S6-I: Waiting for Ack 1,0
S7-I: Waiting for Ack 1,1
S8-I
E1: Secure request
A1 : Send SIE1
A2 : Start T103
A3 : Initialize I-SIE Retry count=0
A4 : Initialize I-Ack Retry count=0
E2: Ack 0,0,NbFrag
A5 : Stop T103
E3: T103 expires
A1: Send SIE1
A2 : Start T103
A6 : Increment I-SIE Retry count
E6: SIE2, SIE2FragNb>1
A8 : Start T104
A9 : Initialize FragNb=1
E7: SIE2, SIE2FragNb=1
A10 : Send Ack 0,0,SIE2FragNb
E8: SIE2, FragNb=SIE2FragNb
A11 : Stop T104
A10 : Send Ack 0,0,SIE2FragNb
E11: SIE2 fragment
A14 : Increment FragNb
A8 : Start T104
E12 : Local process over
A1 : Send SIE3
A2 : Start T103
A3 : Initializa I-SIE Retry count=0
A9 : Initialize FragNb=1
E13 : Ack 1,0,SeqNb=SIE3FragNb
A4 : Stop T103
E3: T103 expires
A1 : Send SIE3
A2 : Start T103
A5 : Increment I-SIE retry count
E4: I-SIE retry count exceeded Max
E4: I-SIE retry count exceeded Max
E5: Ack 0,0,<=SIE1FragNb
A7 : Send SIE1 fragment from SeqNb to SIE1FragNb
A2 : Start T103
E9: T104 expires
A12 : Send Ack 0,0, FragNb-1
A8 : Start T104
A13 : Increment I-Ack retry count
E5: Ack 0,0, SeqNb<=SIE3FragNb
A7 : Send SIE3 fragments from SeqNb to SIE3FragNb
A2 : Start T103
E10: I-Ack retry count exceeded max
E17: Ack 1,1,SIE3FragNb
A15 : Send SAC

Figure 8. Initiator FSM.

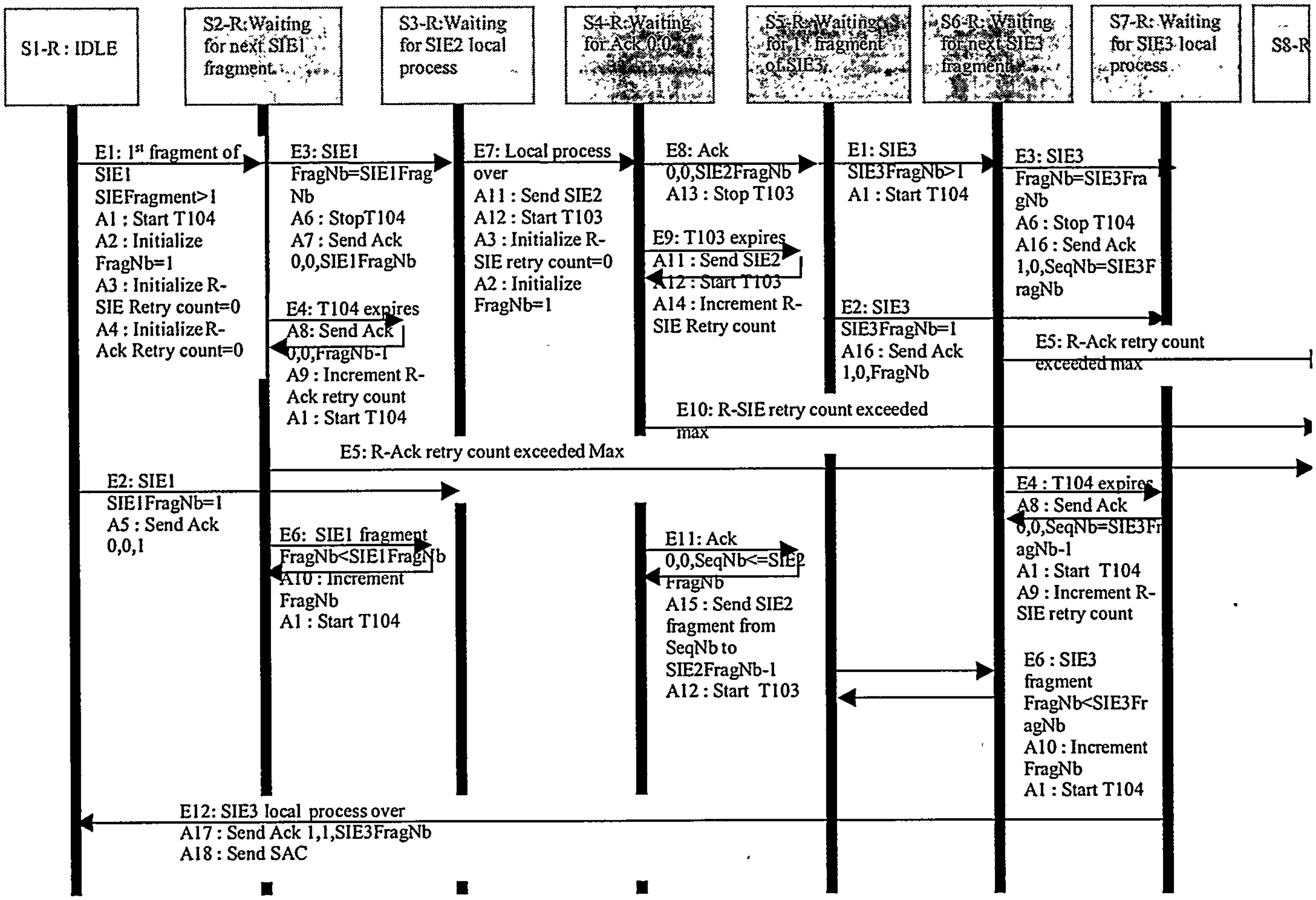

S1-R : IDLE
S2-R: Waiting for next SIE1 fragment
S3-R: Waiting for SIE2 local process
S4-R: Waiting for Ack 0,0,x
S5-R: Waiting for 1st fragment of SIE3
S6-R: Waiting for next SIE3 fragment
S7-R: Waiting for SIE3 local process
S8-R
E1: 1st fragment of SIE1 SIEFragment>1 A1 : Start T104 A2 : Initialize FragNb=1 A3 : Initialize R-SIE Retry count=0 A4 : Initialize R-Ack Retry count=0
E3: SIE1 FragNb=SIE1FragNb A6 : StopT104 A7 : Send Ack 0,0,SIE1FragNb
E7: Local process over A11 : Send SIE2 A12 : Start T103 A3 : Initialize R-SIE retry count=0 A2 : Initialize FragNb=1
E8: Ack 0,0,SIE2FragNb A13 : Stop T103
E1: SIE3 SIE3FragNb>1 A1 : Start T104
E3: SIE3 FragNb=SIE3FragNb A6 : Stop T104 A16 : Send Ack 1,0,SeqNb=SIE3FragNb
E9: T103 expires A11 : Send SIE2 A12 : Start T103 A14 : Increment R-SIE Retry count
E4: T104 expires A8: Send Ack 0,0,FragNb-1 A9 : Increment R-Ack retry count A1 : Start T104
E2: SIE3 SIE3FragNb=1 A16 : Send Ack 1,0,FragNb
E5: R-Ack retry count exceeded max
E10: R-SIE retry count exceeded max
E5: R-Ack retry count exceeded Max
E2: SIE1 SIE1FragNb=1 A5 : Send Ack 0,0,1
E6: SIE1 fragment FragNb<SIE1FragNb A10 : Increment FragNb A1 : Start T104
E11: Ack 0,0,SeqNb<=SIE2FragNb A15 : Send SIE2 fragment from SeqNb to SIE2FragNb-1 A12 : Start T103
E4 : T104 expires A8 : Send Ack 0,0,SeqNb=SIE3FragNb-1 A1 : Start T104 A9 : Increment R-SIE retry count
E6 : SIE3 fragment FragNb<SIE3FragNb A10 : Increment FragNb A1 : Start T104
E12: SIE3 local process over A17 : Send Ack 1,1,SIE3FragNb A18 : Send SAC

Figure 9. Responder FSM.

5.3.2. 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, RKC=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	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 constructio n	Acknowledge ment cells MEC=1, RKC=0	Initiator waits until the SIE3 is constructed and ready to be sent.
S6 -I	Waiting for Acknowled gement cells MEC=1 RKC=0	Full SIE3	Initiator waits for the acknowledgement cells which acknowledge the receipt of the SIE3
S7 -I	Waiting for Acknowled gement cells MEC=1 RKC=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 + Acknowledge ment cells MEC=1, RKC=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 constructio n	None	Responder waits until the SIE2 is locally constructed
S4-R	Waiting for Acknowled gement cells MEC=0 RKC=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	None	Initiator sends the first fragment of the SIE3

	fragment of 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, RKC=0	Responder waits until the SIE3 is processed locally
S8-R	Failed	None	Error occurred during the security association negotiation.

5.3.3. 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, RKC=0, SeqNb=SIEFragNb) Received	Acknowledgement cells have been received and indicate that the SSIE fragmented into SIEFragNb 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, RKC=0, SeqNb<=SIEFragNb) Received	Acknowledgement cells have been received but indicates that cells were lost • this event results in the new transmission of the SSIE fragments numbered from SeqNb
E6	**Valid SIE2 first fragment Received (SIE2FragNb>1)**	**The initiator receives the first fragment of the SIE2** • **this event results in new fragments being expected**
E7	**Valid SIE2 fully Received (SIE2FragNb=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=SIE2FragNb)**	**The initiator receives the full SIE2 in SIE2FragNb fragments** • **this event is expected if the SIE2 requires more than one fragment.**
E9	**T104 expires**	**Timer T104 has exceeded time shown in section 5.3.1.1** • **this event results in Acknowledgement cells MEC=0, RKC=0, SeqNb=FragNb-1**
E10	**I-Ack Retry Count Exceeded**	**Initiator has sent a group of acknowledgement cells (MEC=0, RKC=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, RKC=0, SeqNb=SIEFragNb) Received**	**Acknowledgement cells have been received** • **this event is expected**
E14	**Valid Ack (MEC=1, RKC=1, SeqNb=SIEFragNb) Received**	**Initiator receives Acknowledgement cells MEC=1, RKC=1** • **this event is expected**
	RESPONDER	
Number	Name	Description
E1	**Valid SSIE first fragment Received (SIEFragNb>1)**	**The responder receives the first fragment of the SSIE** • **this event results in new fragments being expected**

E2	Valid SSIE received (SIEFragNb=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=SIEFragNb)	The responder receives the full SSIE in SIEFragNb fragments • this event is expected if the SSIE requires more than one fragment.
E4	**T104 Expires**	**Timer T104 has exceeded time shown in section 5.3.1.1** • **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, RKC=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, RKC=0, SeqNb=SIE2FragNb) Received**	**Acknowledgement cells have been received** • **this event is expected**
E9	**T103 expires**	**Timer T103 has exceeded time shown in section 5.3.1.1** • **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, RKC=0, SeqNb<=SIE2FragNb) Received**	**Acknowledgement cells have been received but indicates that cells were lost** • **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**

5.3.4. FSM Actions

Table 10: Actions

	INITIATOR	
Number	Name	Description
A1	Send SSIE	Send a SSIE from initiator to responder using FragNbSIE 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 FragNbSIE from initiator to responder
A8	Start T104 Timer	Start timer T104
A9	Initialize FragNb counts	Set FragNb=1
A10	Send Ack MEC=0, RKC=0, SeqNb=FragNbSIE2	Send a group of acknowledgement cells with MEC=0, RKC=0, and SeqNb=FragNbSIE2 from initiator to responder
A11	Stop T104 Timer	Stop timer T104
A12	Send Ack MEC=0, RKC=0, SeqNb=FragNb-1	Send a group of acknowledgement cells with MEC=0, RKC=0, and SeqNb=FragNb from initiator to responder
A13	Increment I-Ack	Increment I-Ack counter
A14	Increment FragNb	Increment FragNb counter
A15	Send SAC	Send a group of SAC cells from initiator to responder
	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, RKC=0, SeqNb=1	Send a group of acknowledgement cells with MEC=0, RKC=0, and SeqNb=1 from responder to initiator
A6	Stop T104 Timer	Stop timer T104
A7	Send Ack MEC=0, RKC=0, SeqNb=FragNbSIE1	Send a group of acknowledgement cells with MEC=0, RKC=0, and SeqNb=FragNbSIE1 from responder to initiator
A8	Send Ack MEC=0, RKC=0,	Send a group of acknowledgement cells with MEC=0, RKC=0, and SeqNb=FragNb-1 from responder to initiator

	SeqNb=FragNb-1	
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 FragNbSIE2 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 FragNbSIE2 from responder to initiator
A16	Send Ack MEC=1, RKC=0, SeqNb=FragNbSIE	Send a group of acknowledgement cells with MEC=1, RKC=0, and SeqNb=FragNbSIE from responder to initiator
A17	Send Ack MEC=1, RKC=1, SeqNb=FragNbSIE	Send a group of acknowledgement cells with MEC=1, RKC=1, and SeqNb=FragNbSIE from responder to initiator
A18	**Send SAC**	**Send a group of SAC cells from responder to initiator**

5.3.5. 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 S2-I	N/A	N/A	N/A	N/A	N/A	N/A	N/A
E2	N/A	A5 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, A5 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 S2-I	N/A	N/A	N/A	A7, A2 S6-I	N/A	N/A
E6	N/A	N/A	A8, A9 S4-I	N/A	N/A	N/A	N/A	N/A
E7	N/A	N/A	A10 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	N/A	A12, A8, A13 S4-I	N/A	N/A	N/A	N/A
E10	N/A	N/A	N/A	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 S6-I	N/A	N/A	N/A
E13	N/A	N/A	N/A	N/A	N/A	A4 S7-I	N/A	N/A
E14	N/A	N/A	N/A	N/A	N/A	N/A	A15 S1-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, A4 S2-R	N/A	N/A	N/A	A1 S6-R	N/A	N/A	N/A
E2	A5 S3-R	N/A	N/A	N/A	A16 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	N/A	A8, A9, A1 S2-R	N/A	N/A	N/A	A8, A9, A1 S6-R	N/A	N/A
E5	N/A	S8-R	N/A	N/A	N/A	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 S4-R	N/A	N/A	N/A	N/A	N/A
E8	N/A	N/A	N/A	A13 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 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 S1-R	N/A

7. Acknowledgment

Ideas of negotiation OAM cells come from the project SCAN (Secure Communications in ATM Networks) which is funded by the European Commission, Directorate General XIII, ACTS Programme 3rd call, project number AC330.

8. References

ATM Forum, "ATM Security Specification Version 1.0", February 1999.

M. Laurent, C. Delahaye, M. Achemlal, "Security services negotiation through OAM cells", ATM Forum/99-0335, July 1999, New Orleans, Louisiana.

H. Leitold, R. Posch, E. Areizaga, A Bouabdallah, M. Laurent, J.M. Mateos, O. Molino, “Security services in ATM networks”, ICON Journal, 1999.