

SECURE COMMUNICATION INTEROPERABILITY PROTOCOL (SCIP)

The Secure Communication Interoperability Protocol (SCIP) is a communications standard developed by the National Security Agency (NSA) to enable interoperable secure communications among allies and partners around the globe.

The SCIP-210 Signaling Plan is the specification that defines the application layer signaling used to negotiate a secure end-to-end session between two communication devices, independent of network transport. SCIP negotiates the operational mode (e.g., voice, data, etc.), the cryptographic algorithm suite (e.g., Suite A, Suite B, etc), and the traffic encryption key used for each secure session. It also provides capabilities for cryptographic synchronization and operational mode control between communicating end-point devices. SCIP is designed to operate over any network and is currently utilized in devices operating on a wide variety of networks including PSTN, ISDN, CDMA, GSM, IP, and satellite.

Potential developers of SCIP devices may contact the NSA SCIP Program Office at SCIP_POC@missi.ncsc.mil for further information. The SCIP-210 Signaling Plan is available without restrictions on its use for the development, manufacture, and sale of SCIP products. Compliance and interoperability testing will be necessary to ensure secure interoperability between the wide variety of current and future SCIP products.

SCIP Signaling Plan

Revision 3.2

December 19, 2007

Prepared for:

**National Security Agency
9800 Savage Road
Fort George G. Meade, MD 20755**

Prepared By:

GENERAL DYNAMICS
C4 Systems
77 "A" Street
Needham, MA 02494

TABLE OF CONTENTS

1.0 INTRODUCTION	1
1.1 Purpose.....	2
1.2 Scope.....	3
1.3 Definitions	4
1.4 Acronyms and Abbreviations	5
1.5 Applicable Documents.....	7
1.5.1 NSA Documents	7
1.5.2 Industry Standards	8
1.5.3 International and National Standards.....	8
1.5.4 Federal and DoD Standards	9
1.5.5 NATO Standards.....	9
1.5.6 Other Relevant Technical Papers.....	9
1.6 Signaling Plan Overview	10
1.6.1 SCIP Application State Diagram	10
1.6.2 SCIP Protocol Layer Diagram	12
1.7 Document Conventions.....	13
2.0 SCIP SIGNALING – Point-to-Point Operation.....	15
2.1 SCIP Message Transport	15
2.1.1 The MER-OC Message Transport Option and the Branch Point Mechanism.....	15
2.1.2 Message Transport Timelines.....	16
2.1.3 Transport Framing	19
2.1.3.1 Start of Message.....	21
2.1.3.2 Frame Count.....	21
2.1.3.3 Cyclic Redundancy Check.....	21
2.1.3.4 Forward Error Control	22
2.1.3.5 End of Message.....	23
2.1.4 Escape	23
2.1.5 Transport Layer Control Messages.....	25
2.1.5.1 REPORT Message	25
2.1.5.1.1 REPORT Message Format.....	26
2.1.5.1.2 Conditions for REPORT Message Transmission	27
2.1.5.1.3 Processing for REPORT Message Reception.....	28
2.1.6 Message Transmission	29
2.1.6.1 Transmit Request	29
2.1.6.2 Transmitter Actions on Receipt of a REPORT	33
2.1.6.3 Retransmit Timeout	33
2.1.7 Message Reception	34
2.1.8 Octet Alignment.....	37
2.2 SCIP Call Setup Signaling.....	38
2.2.1 Introduction and Overview	38
2.2.1.1 Secure Call Setup Signaling Time Line.....	38
2.2.1.1.1 FIREFLY Example	38
2.2.1.1.2 PPK Example	40

TABLE OF CONTENTS (Cont.)

2.2.1.2	First Message Time-Out	42
2.2.1.3	Unrecognized Messages	43
2.2.1.4	Message Limitations	43
2.2.2	Capabilities Message	43
2.2.2.1	Capabilities Message Definition.....	43
2.2.2.2	Capabilities Message Transmission.....	56
2.2.2.3	Capabilities Message Reception	58
2.2.2.3.1	Capabilities Message Reception Unique Processing	59
2.2.2.3.2	Common Capabilities Message Processing	60
2.2.2.4	Extended Keysets List Message Definition	64
2.2.3	Parameters/Certificate Message.....	66
2.2.3.1	Parameters/Certificate Message Definition	66
2.2.3.2	Parameters/Certificate Message Transmission	69
2.2.3.3	Parameters/Certificate Message Reception	71
2.2.4	F(R) Message.....	76
2.2.4.1	F(R) Message Definition	76
2.2.4.2	F(R) Message Transmission	78
2.2.4.3	F(R) Message Reception.....	80
2.2.4.3.1	F(R) Message Received.....	80
2.2.4.3.2	Parameters/Certificate Message Received.....	80
2.2.5	Cryptosync Exchange	81
2.2.5.1	Cryptosync Message Definition.....	82
2.2.5.2	Cryptosync Message Transmission	83
2.2.5.3	Cryptosync Message Reception.....	85
2.2.5.3.1	Cryptosync Message Received	85
2.2.5.3.2	Parameters/Certificate Message Received.....	87
2.2.6	Operational Mode and Keypset Type Specific Instantiations	87
2.2.6.1	Key Agreement Specifics	87
2.2.6.1.1	Capabilities and Extended Keysets List Messages.....	87
2.2.6.1.1.1	Type 1 FIREFLY Without CSE.....	87
2.2.6.1.1.2	Type 1 FIREFLY With CSE.....	88
2.2.6.1.1.3	Type 1 U.S. Generic PPK Without CSE.....	90
2.2.6.1.1.4	ECMQV/AES Without CSE – Phase 1.....	90
2.2.6.1.1.5	ECMQV/AES With CSE – Phase 1.....	91
2.2.6.1.1.6	NATO ECMQV/AES Without CSE.....	91
2.2.6.1.1.7	NATO ECMQV/AES With CSE	92
2.2.6.1.1.8	NATO PPK/AES Without CSE.....	94
2.2.6.1.1.9	Extended Keysets List Support.....	94
2.2.6.1.2	Parameters/Certificate Message.....	95
2.2.6.1.2.1	Type 1 FIREFLY	95
2.2.6.1.2.2	Type 1 U.S. Generic PPK	95
2.2.6.1.2.3	ECMQV/AES – Phase 1	95
2.2.6.1.2.4	NATO ECMQV/AES	96
2.2.6.1.2.5	NATO PPK/AES	96

TABLE OF CONTENTS (Cont.)

2.2.6.1.3 F(R) Message	96
2.2.6.1.3.1 Type 1 FIREFLY	97
2.2.6.1.3.2 Type 1 U.S. Generic PPK	97
2.2.6.1.3.3 ECMQV/AES – Phase 1	97
2.2.6.1.3.4 NATO ECMQV/AES	97
2.2.6.1.3.5 NATO PPK/AES	97
2.2.6.2 Secure Voice Specifics	98
2.2.6.2.1 Secure MELP and Secure G.729D Voice Options	100
2.2.6.2.2 Secure AMBE Voice Specific Option	100
2.2.6.3 Secure Data Specifics	100
2.2.6.3.1 Secure Data Operational Mode Parameters	101
2.2.6.3.2 Enhanced Secure Data Operational Mode Parameters	103
2.2.6.4 Secure Electronic Rekey Specifics	104
2.2.6.5 Clear MELP Voice Specifics	105
2.3 SCIP Call Control Signaling.....	106
2.3.1 Call Control Timelines.....	106
2.3.2 Notification Message Processing.....	109
2.3.2.1 Notification Message Definition.....	110
2.3.2.2 Notification (Connection Terminate).....	114
2.3.2.3 Notification (Native Clear Voice/Connection Idle).....	116
2.3.2.3.1 Failed Call.....	119
2.3.2.3.2 Nonsecure Selected.....	120
2.3.2.3.3 Secure Selected	120
2.3.2.3.4 Secure Restart	120
2.3.2.3.5 Notification (Native Clear Voice/Connection Idle) Receive Processing	121
2.3.2.4 Notification (CKL Transfer).....	121
2.3.2.5 Notification (Secure Dial).....	124
2.3.2.5.1 Encryption of Secure Dial Characters	127
2.3.2.5.2 Data Transmission and Reception	127
2.3.2.6 Notification (Attention)	129
2.3.2.7 Notification (Secure Update).....	131
2.3.3 Mode Change Processing.....	134
2.3.3.1 Mode Change Request Message	134
2.3.3.2 Mode Change Response Message.....	137
2.3.4 Two-Way Resync Processing	139
2.4 SCIP Signaling Timeouts.....	142
2.5 SCIP Signaling Constants.....	144
2.5.1 Source Definitions	144
2.5.2 MIDs	144
2.5.3 Miscellaneous SCIP Signaling Constants.....	145
2.5.3.1 ESCAPE.....	145
2.5.3.2 Start of Message (SOM) and End of Message (EOM)	145
2.5.3.3 START and FILLER.....	145

TABLE OF CONTENTS (Cont.)

2.5.3.4 Headers	146
3.0 SCIP USER APPLICATION SIGNALING – Point-to-Point Operation	149
3.1 SCIP User Applications	149
3.2 Application Start-up/Restart Signaling.....	149
3.2.1 Full Bandwidth Applications	149
3.2.1.1 Application Timeout	150
3.2.2 Reliable Transport Applications	152
3.3 Secure Voice Applications.....	153
3.3.1 Secure MELP Voice	153
3.3.1.1 Secure 2400 bps MELP Voice – Blank and Burst.....	153
3.3.1.1.1 Sync Management Frame	154
3.3.1.1.2 Encryption and Transmission Ordering.....	156
3.3.1.2 Secure MELP Voice – Burst w/o Blank	158
3.3.1.2.1 Sync Management Frame	159
3.3.1.2.2 Encryption and Transmission Ordering.....	160
3.3.1.3 Clear MELP Voice – Blank and Burst.....	162
3.3.1.3.1 Sync Management Frame	163
3.3.1.3.2 Transmission Ordering	164
3.3.1.4 Voice Activity Factor Processing	165
3.3.1.4.1 Discontinuous Voice Transmission	165
3.3.1.4.2 Force Continuous Transmission	165
3.3.2 Secure G.729D Voice	166
3.3.2.1 Secure G.729D Voice Frame	168
3.3.2.1.1 Secure G.729D Voice Transmission Format	168
3.3.2.2 Sync Management Frame	169
3.3.2.3 Encrypted Speech Frame Header.....	170
3.3.2.4 Encryption and Transmission Ordering	171
3.3.2.5 Discontinuous Voice Transmission	173
3.3.2.6 Force Continuous Transmission	173
3.4 Secure Data Applications.....	173
3.4.1 Secure Reliable Transport (RT) Asynchronous Data	173
3.4.1.1 Secure RT Asynchronous Data Transmission	174
3.4.1.1.1 Encryption and Transmission Ordering.....	175
3.4.1.1.2 Message Transmission.....	177
3.4.1.2 Secure RT Asynchronous Data Message Reception.....	177
3.4.2 Secure Best Effort Transport (BET) Asynchronous Data	177
3.4.2.1 Sync Management Frame	180
3.4.2.2 Encryption and Transmission Ordering.....	180
4.0 SCIP ELECTRONIC REKEY SIGNALING	183
4.1 Electronic Rekey Protocol Architecture and Communication Paths	184
4.2 SCIP Electronic Rekey Message Transport.....	186
4.2.1 Encryption and Transmission Ordering.....	187
4.2.2 SCIP Rekey Message Transmission	188
4.2.3 SCIP Rekey Message Reception.....	188

TABLE OF CONTENTS (Cont.)

4.3	Adaptation Layer	189
4.4	Generic Rekey Application Layer	190
5.0	SCIP SIGNALING – Multipoint Operation	191
5.1	Multipoint Message Transport.....	191
5.1.1	Multipoint Transport Framing	191
5.1.1.1	Multipoint Message Transmission.....	192
5.1.1.2	Multipoint Message Reception	192
5.1.2	Multipoint Cryptosync Message.....	193
5.1.2.1	Multipoint Cryptosync Message Definition	193
5.1.2.2	Multipoint Sync Parameters.....	195
5.1.2.2.1	Sync Verification Pattern.....	196
5.1.3	FILLER – Multipoint Operation.....	196
5.1.4	START – Multipoint Operation.....	196
5.1.5	End of Transmission – Multipoint Operation	196
5.2	Multipoint Session	197
5.2.1	Multipoint Transmission.....	198
5.2.1.1	Multipoint Cryptosync Message Transmission	199
5.2.1.2	Multipoint Secure Traffic Transmission.....	201
5.2.1.2.1	Multipoint Secure MELP Voice Transmission.....	201
5.2.1.2.2	Multipoint Secure G.729D Voice Transmission.....	202
5.2.1.2.3	Multipoint Secure Data Transmission	202
5.2.1.3	End of Multipoint Secure Traffic Transmission	203
5.2.2	Multipoint Reception	204
5.2.2.1	Multipoint Cryptosync Message Reception.....	204
5.2.2.2	Multipoint Secure Traffic Reception	206
5.2.2.2.1	Multipoint Secure MELP Voice Reception	207
5.2.2.2.2	Multipoint Secure G.729D Voice Reception.....	207
5.2.2.2.3	Multipoint Secure Data Reception.....	207
5.2.2.3	Late Entry (Including Re-Entry).....	207
5.2.2.4	End of Multipoint Secure Traffic Reception	209
A.0	SCIP MESSAGE TRANSPORT PROTOCOL EXAMPLES	A-1
A.1	Normal Capabilities Message Transfer.....	A-2
A.2	Parameters/Certificate Message Transfer with Corrupted and Missing Frames	A-4
A.3	F(R) Message Transfer with Corrupted SOM and EOM Sequences	A-7
A.4	CAPABILITIES Message Transfer with Corrupted REPORT Responses.....	A-10
A.5	Normal Transition from Signaling to Full Bandwidth Application.....	A-13
A.6	Transition from Signaling to Full Bandwidth Application with Final REPORT Lost	A-16
A.7	Transition from Signaling to Full Bandwidth Application with START Lost	A-20
A.8	Two Way Resync from Full Bandwidth Application, Terminal A is Leader	A-24
A.9	Two Way Resync from Full Bandwidth Application with Corrupted ESC Sequence, Terminal A is Leader.....	A-27
A.10	Normal Termination from Full Bandwidth Application, Terminal A is Leader.	A-30
A.11	Terminal A Sends Notify(Attention) from Full Bandwidth Application.....	A-32

TABLE OF CONTENTS (Cont.)

B.0 DISCONTINUOUS VOICE (DTX)	B-1
B.1 Voice Activity Detection (VAD)	B-1
B.2 Default Voice Activity Detection (VAD) Algorithm.....	B-2
B.3 Grace Period.....	B-3
B.4 Blank Period.....	B-4
B.5 Comfort Noise.....	B-5
B.6 Re-Start	B-5
C.0 PERFORMANCE	C-1
C.1 DTX Voice	C-1
C.1.1 MELP Blank and Burst	C-1

LIST OF FIGURES

Figure 1.6-1 SCIP Application State Diagram - Point-to-Point	11
Figure 1.6-2 SCIP Protocol Layer Diagram - Point-to-Point	12
Figure 1.7-1 Process Diagram Symbols.....	13
Figure 2.1-1(a) Transport Layer Signaling Time Line (Framed)	17
Figure 2.1-1(b) Transport Layer Signaling Time Line (Full bandwidth-to-Framed).....	18
Figure 2.1-1(c) Transport Layer Signaling Time Line (Full bandwidth-to-Full bandwidth)	18
Figure 2.1-2 Transmission Frame Group.....	19
Figure 2.1-3 ESCAPE Processing	24
Figure 2.1-4(a) Message Transmission.....	31
Figure 2.1-4(b) Message Transmission (Cont.).....	32
Figure 2.1-5(a) Message Reception	35
Figure 2.1-5(b) Message Reception (Cont.)	36
Figure 2.2-1(a) FIREFLY Secure Call Setup Signaling Time Line	39
Figure 2.2-1(b) PPK Secure Call Setup Signaling Time Line.....	41
Figure 2.2-2 Capabilities Message Transmission.....	57
Figure 2.2-3 Capabilities Message Reception Unique Processing	60
Figure 2.2-4 Common Capabilities Message Processing	62
Figure 2.2-5 Parameters/Certificate Message Transmission	70
Figure 2.2-6(a) Parameters/Certificate Message Reception	72
Figure 2.2-6(b) Parameters/Certificate Message Reception (Cont.)	73
Figure 2.2-6(c) Parameters/Certificate Message Reception (Cont.).....	74
Figure 2.2-7 F(R) Message Transmission.....	79
Figure 2.2-8 F(R) Message Reception.....	81
Figure 2.2-9 Cryptosync Message Transmission.....	84
Figure 2.2-10 Cryptosync Message Reception.....	86
Figure 2.3-1(a) Notification Message Signaling Time Line (Full Bandwidth to Framed).....	107
Figure 2.3-1(b) Notification Message Signaling Time Line (Framed to Framed)	107
Figure 2.3-1(c) Notification Message Signaling Time Line (Full Bandwidth to Full Bandwidth).....	107
Figure 2.3-1(d) Mode Change Signaling Time Line	108
Figure 2.3-1(e) Two-Way Resync Signaling Time Line	108
Figure 2.3-2 Notification Message Processing (Connection Terminate)	115
Figure 2.3-3(a) Notification Message Processing (Native Clear Voice/Connection Idle)	117
Figure 2.3-3(b) Notification Message Processing (Native Clear Voice/Connection Idle) (Cont.)	118
Figure 2.3-4 Notification Message Receive Processing (CKL Transfer).....	124
Figure 2.3-5 Notification Message Processing (Secure Dial)	126
Figure 2.3-6 Notification Message Processing (Attention)	130
Figure 2.3-7 Notification Message Processing (Secure Update).....	133
Figure 2.3-8 Mode Change Processing.....	135
Figure 2.3-9 Two-Way Resync Processing	140
Figure 3.2-1 Application Timeout Processing.....	151
Figure 3.3-1 Secure MELP Voice Transmission Format – Blank and Burst	154
Figure 3.3-2 Sync Management Frame Format – Blank and Burst	154

LIST OF FIGURES (Cont.)

Figure 3.3-3 Secure MELP Voice Transmission Format – Burst w/o Blank	159
Figure 3.3-4 Sync Management Frame Format – Burst w/o Blank.....	159
Figure 3.3-5 Clear MELP Voice Transmission Format.....	163
Figure 3.3-6 Clear MELP Voice Sync Management Frame Format	163
Figure 3.3-7 Secure G.729D Voice Transmission.....	166
Figure 3.3-8 Secure G.729D Voice Superframe Details	167
Figure 3.3-9 Secure G.729D Voice Escape and Return Example (No Cryptosync)	167
Figure 3.3-10 Secure G.729D Voice Sync Management Frame Format.....	169
Figure 3.3-11 Secure G.729D Voice Encrypted Speech Frame Header.....	170
Figure 3.4-1 Secure RT Asynchronous Data Message Preparation.....	174
Figure 3.4-2 V.14 Asynchronous Data Input Ordering	175
Figure 3.4-3 Secure BET Asynchronous Data Transmission Format	178
Figure 3.4-4 Secure BET Asynchronous Data Superframe Structure	179
Figure 3.4-5 Sync Management Frame Format	180
Figure 3.4-6 V.14 Asynchronous Data Input Ordering	180
Figure 4.1-1 Rekey Protocol Conversion Using the GRFE.....	184
Figure 4.1-2 Electronic Rekey System Infrastructure	185
Figure 4.2-1 SCIP Rekey Message Preparation.....	186
Figure 5.1-1 Multipoint Transport Signaling Timeline	191
Figure 5.1-2 Multiple Multipoint Cryptosync Message Transmissions	192
Figure 5.2-1 SCIP Multipoint State Diagram	197
Figure 5.2-2 Multipoint Secure Voice Transmit Signaling Time Line.....	198
Figure 5.2-3 Multipoint Cryptosync Message Transmission.....	200
Figure 5.2-4 Multipoint MELP Voice Transmission Format – Blank and Burst	202
Figure 5.2-5 End of Multipoint Secure Traffic Transmission	203
Figure 5.2-6 Multipoint Cryptosync Message Reception.....	205
Figure 5.2-7 Multipoint Secure Voice Traffic Reception.....	206
Figure 5.2-8 Multipoint Late Entry Cryptographic Synchronization	208
Figure 5.2-9 End of Multipoint Secure Traffic Reception.....	209
Figure B-1 DTX Voice	B-1

LIST OF TABLES

Table 2.1-1	Frame Group Format	20
Table 2.1-2	REPORT Message Data Format.....	26
Table 2.2-1(a)	Capabilities Message Format.....	44
Table 2.2-1(b)	Capabilities Message Format (Cont.)	45
Table 2.2-1(c)	Capabilities Message Format – Version 1 or Higher (Cont.)	45
Table 2.2-2	SCIP Standard Operational Modes.....	47
Table 2.2-3(a)	Keysets List Entry - General Format	48
Table 2.2-3(b)	SCIP Standard Keyset Types.....	49
Table 2.2-3(c)	Terminal Priority COI Values.....	50
Table 2.2-3(d)	Terminal Priority Values	50
Table 2.2-3(e)	Example of Capabilities Message Contents – Enhanced FF Capable	52
Table 2.2-3(f)	Extended Keysets List Message Format	65
Table 2.2-4	Parameters/Certificate Message Format.....	67
Table 2.2-5	F(R) Message - General Format.....	77
Table 2.2-6	Cryptosync Message - General Format.....	82
Table 2.2-7(a)	Keyset Parameters Entry – Type 1 Basic and Enhanced FF w/o CSE Format....	88
Table 2.2-7(b)	Keyset Parameters Entry – Type 1 Basic and Enhanced FF w/CSE Format.....	89
Table 2.2-7(c)	Keyset Parameters Entry – Type 1 U.S. Generic PPK w/o CSE Format.....	90
Table 2.2-7(d)	Keyset Parameters Entry – ECMQV/AES w/o CSE – Phase 1 Format	90
Table 2.2-7(e)	Keyset Parameters Entry – ECMQV/AES w/CSE – Phase 1 Format.....	91
Table 2.2-7(f)	Keyset Parameters Entry – NATO ECMQV/AES w/o CSE Format	92
Table 2.2-7(g)	Keyset Parameters Entry – NATO ECMQV/AES w/CSE Format.....	93
Table 2.2-7(h)	Keyset Parameters Entry – NATO PPK/AES w/o CSE Format.....	94
Table 2.2-7(i)	Keyset Parameters Entry – Extended Keysets List Support Format	94
Table 2.2-8	Certificate Field Format	95
Table 2.2-9	F(R) Field Format.....	96
Table 2.2-10	Operational Mode Parameters – Secure Voice.....	98
Table 2.2-11	Interoperable Security Levels.....	99
Table 2.2-12	Secure Voice Options.....	100
Table 2.2-13	Secure Data/Enhanced Secure Data Options.....	101
Table 2.2-14	Operational Mode Parameters – Secure Data.....	102
Table 2.2-15(a)	Operational Mode Parameters – Enhanced Secure Data	103
Table 2.2-15(b)	Enhanced Secure Data Option Entry	103
Table 2.2-16	Operational Mode Parameters – Secure Electronic Rekey	104
Table 2.2-17	Electronic Rekey Options.....	105
Table 2.3-1	Notification Message Format	110
Table 2.3-2	SCIP Standard Action Field Values	111
Table 2.3-3	Information Field Entry Format	112
Table 2.3-4	SCIP Standard Information Code Definitions.....	113
Table 2.3-5	CKL Transfer - Information Text.....	122
Table 2.3-6	Secure Dial Characters	125
Table 2.3-7	Secure Dial - Information Text	128
Table 2.3-8	Secure Update - Information Text.....	131
Table 2.3-9	Mode Change Request Message Format	136

LIST OF TABLES (Cont.)

Table 2.3-10 Mode Change Response Message Format.....	138
Table 2.4-1 SCIP Signaling Timeouts	142
Table 2.5-1 Source Definitions	144
Table 2.5-2 MIDs.....	144
Table 2.5-3 Miscellaneous SCIP Signaling Constants	147
Table 3.3-1 Sync Management Frame Contents – Blank and Burst.....	155
Table 3.3-2 Secure MELP Transmission Bit Ordering – Blank and Burst.....	157
Table 3.3-3 Sync Management Frame Contents – Burst w/o Blank	160
Table 3.3-4 Secure MELP Transmission Bit Ordering – Burst w/o Blank	161
Table 3.3-5 Clear MELP Voice Sync Management Frame Contents.....	164
Table 3.3-6 Clear MELP Voice Transmission Bit Ordering – Blank and Burst	164
Table 3.3-7 Secure G.729D Voice Frame Parameters	168
Table 3.3-8 G.729D Vocoder Frame Bit Transmission Order.....	169
Table 3.3-9 Secure G.729D Voice Sync Management Frame Contents.....	170
Table 3.3-10 Secure G.729D Voice Encrypted Speech Frame Header Contents.....	170
Table 3.3-11(a) Secure G.729D Voice Transmission Bit Ordering (Octets 1 - 8)	171
Table 3.3-11(b) Secure G.729D Voice Transmission Bit Ordering (Octets 9 - 288).....	172
Table 3.4-1 Secure RT Asynchronous Data Message Format.....	176
Table 3.4-2 Validity Count Field Values.....	179
Table 3.4-3 Sync Management Frame Contents.....	180
Table 3.4-4 Secure BET Asynchronous Data Transmission Bit Ordering	181
Table 4.2-1 SCIP Rekey Message Format.....	187
Table 4.4-1 Generic Rekey Protocol Data Units (GRPDU).....	190
Table 5.1-1 Multipoint Cryptosync Message – General Format	193
Table 5.1-2 Sync Parameters	195
Table B.1-1 DTX VAF Values	B-2
Table B.3-1 MELP Comfort Noise Parameter Values.....	B-4
Table B.4-1 Blank Period Values	B-4

Signaling Plan Notice

Revision 3.2 of the SCIP Signaling Plan, designated as SCIP-210, is an update of Revision 3.1. It incorporates changes from ECPs 26 and 27. The more significant changes are listed below.

- Applicable documents were updated and acronyms were added.
- A Message Limitations section was added to ensure interoperability with SCIP devices.
- Signaling changes for Extended Keypsets Lists were added.
 - An Extended Keypsets List Message and an Extended Keypsets List Support Keypset were added to extend the keyset list in the Capabilities Message.
 - The Common Capabilities Message Processing and Secure Call Setup Signaling Time Lines were modified to show optional Extended Keypsets List Message exchanges.
- Signaling changes for Enhanced Secure Data were added.
 - An Enhanced Secure Data Operational mode was added along with an Enhanced Secure Data Operational Mode Parameters format.
 - Data options may be listed in the Operational Mode Parameters associated with Secure Data, Enhanced Secure Data, or both Operational Mode(s).
- Clarified that SCIP signaling can be used to negotiate specific data application uses (e.g., fax, chat) of data Options (e.g., Secure RT Data) by assigning them a different Option ID.
- Guaranteed Throughput (GT) Data was renamed to Best Effort Transport (BET) Data.
 - References to 2400 bps were removed; this data mode scales to any data rate.
- References to the order in which bits are encrypted were removed since the requirements are specified in the cryptographic specifications.
- Bit ordering at the application layer was separated from bit ordering at the lower layers.
 - SCIP terminal transmission bit ordering over various network interfaces will be provided in SCIP-214 and SCIP-215.

All changes are indicated by change bars. If changes were made to a figure or a table, a change bar appears at the end of the title.

THIS PAGE INTENTIONALLY LEFT BLANK.

1
2 **1.0 INTRODUCTION**
3

4 This document specifies the signaling requirements for the Secure Communication
5 Interoperability Protocol (SCIP) operational modes. The requirements represent the efforts of a
6 working group established for the development, analysis, selection, definition and refinement of
7 signaling for the operational modes of a new class of secure voice and data terminals intended
8 for use on the emerging digital narrowband channels. These channels include digital cellular
9 systems such as GSM and CDMA, digital mobile satellite systems, and a variety of other
10 narrowband digital systems that are also within the scope of interest for the working group. The
11 SCIP signaling is designed to be sufficiently flexible so that subsequent updates and revisions
12 may include various future networks of interest.

13
14 The main body of the SCIP Signaling Plan contains requirements common to all SCIP
15 implementations. It specifies a secure overlay capable of interoperation with SCIP compatible
16 equipment on various similar or disparate networks. Since the various networks will often have
17 different lower-layer communications protocols, the SCIP secure overlay specification specifies
18 the higher-layer end-to-end protocols only. The implementation-specific details for a terminal
19 connected to a particular network are defined in an appendix specific to that network. The
20 appendices specify modes, service options, and other network-specific issues that do not affect
21 terminals on another network. A full terminal design requires the secure overlay specification
22 and the appendix with the requirements for use of the lower-layer communications interface.
23 The secure overlay description and the appendices may be published as a single document or
24 separately as desired.

25
26 The goal of separating the secure overlay from the network-specific appendices is to ensure that
27 there is a stable specification for interoperability and to avoid confusion caused by the differing
28 requirements for the various networks. A specific product development will involve generation
29 of a network-specific appendix which is independent of the application overlay requirements.
30 Each terminal development program (e.g., CDMA cellular, etc.) can proceed independently by
31 generating and/or modifying the implementation-specific appendix for that network. By
32 avoiding modifications to the secure overlay description, configuration management will be
33 simplified. Also, developers of a terminal for one network need not be concerned with the
34 lower-layer requirements for another network.
35
36

37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

1.1 Purpose

The purpose of this document is to define the signaling for point-to-point and multipoint secure communication among terminals operating over narrowband digital networks. The Signaling Plan defines:

- (1) The exchange of keys, certificates or other information between point-to-point terminals preparatory to the exchange of secure voice or data traffic,
- (2) The transmission of secure voice traffic among the user terminals for point-to-point and multipoint operation using the DoD standard MELP or NATO standard MELPe vocoder at 2400 bps, and the ITU-T Recommendation G.729 Annex D CS-ACELP vocoder at 6400 bps,
- (3) The transmission of secure data traffic between the user terminals for point-to-point secure data communication,
- (4) The security control signaling necessary to establish, maintain, and terminate the secure mode of operation,
- (5) The signaling to support point-to-point electronic or over-the-air rekey of the keys or keying material used by the terminals,
- (6) The signaling point of departure to allow vendors to add proprietary signaling and modes of operation to the interoperable standard modes defined by the remainder of the signaling plan.

The purpose of this Signaling Plan is to support communication between SCIP terminals independent of the transport network being used (e.g., digital wireless networks, IP networks, and PSTN/ISDN networks). The signaling is intended to operate using commercially available standards based data services, and standard Interworking Functions (IWFs) with no need for additional specialized interworking functions or operations.

Within the class of commercially operated digital wireless networks, the purpose of this Signaling Plan is to define the signaling required for secure voice operation over the CDMA and GSM digital cellular systems, mobile satellite systems, and other narrowband digital systems.

73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

1.2 Scope

This Signaling Plan is intended to specify the end-to-end signaling used by the secure voice and data elements. Nothing will be contained in the Signaling Plan about the additional signaling within the communication links that might be used to convey the signaling between the terminal elements.

It is within the scope of this Signaling Plan to provide flexibility for the extension to subsequent versions so that if changes are required to incorporate additional networks and objectives, the changes can be incorporated.

It is not within the scope of the Signaling Plan to dictate or otherwise specify any particular method of implementation. Where implementation methods may be implied by the signaling, this is only for illustrative purposes. The potential for new features after the first equipment models, however, suggests that implementers may want to perform the implementation with some flexibility and expansion potential for subsequent models of equipment designed to operate over additional networks.

The Signaling Plan is intended to define the SCIP overlay signaling for the clear digital voice and secure voice/data applications using a standard data bearer service. The SCIP clear digital voice mode signaling is based on the possibility that a voice-followed-by-data communications service for the clear to secure mode transition may not exist. Note that the SCIP clear digital voice mode utilizes SCIP specific signaling and is compatible with SCIP devices only.

Signaling aspects that are specifically outside the scope of this signaling plan are:

- (1) Signaling for the creation of the network connection between terminals as required to establish a path for the “native” (non-SCIP) clear or non-secure mode of operation.
- (2) Signaling for establishing the bearer service or service option preparatory to the initiation of the secure mode of operation.

106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146

1.3 Definitions

The following terms are used throughout this document:

Initiator - The terminal that initiates the secure call setup.

Responder - The terminal that responds to the signaling sequence started by the Initiator.

Leader - The terminal that begins a signaling sequence as a result of some user/machine determined condition, e.g., out of sync detection, voice/data transition, activating the non-secure control, or an error (failed call) condition.

Follower – The terminal that responds to the signaling sequence started by the Leader.

Local – The terminal where operation is currently being described.

Remote – The far-end terminal.

Clear – Not encrypted (does not refer to a user action).

Protected – A level of security used for Sensitive, but Unclassified information. Note that “protected” with a lower case “p” refers to the standard English definition.

Credentials – Certificate and F(R).

MER-OC – If this capability is implemented, it must be as specified herein.

Type 1 – NSA approved encryption for protection of Classified information.

Non-Type 1 – NSA or NIST approved encryption for protection of Sensitive, but Unclassified information.

ECMQV/AES – Non-Type 1 cryptographic suite that is specified in SCIP-231.

NATO ECMQV/AES – NATO interim cryptographic suite, specified in SCIP-232, for protection of Classified information.

SCIP-23x – The Cryptography Specifications listed in Section 1.5.1 (e.g., SCIP-230, SCIP-231, or SCIP-232).

147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192

1.4 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this document.

- ACL - Access Control List
- AES - Advanced Encryption Standard
- AMBE - Advanced Multi-Band Excitation
- APDU - Application Protocol Data Unit
- ASN.1 - Abstract Syntax Notation One
- BCH - Bose-Chaudhuri, Hocquenghem (Error Correcting Code)
- BER - Bit Error Rate
- BET - Best Effort Transport
- bps - bits per second
- CCITT - International Consultative Committee on Telegraphy and Telephony
- CDMA - Code Division Multiple Access
- CELP - Codebook Excited Linear Prediction
- CIK - Crypto Ignition Key
- CF - Central Facility
- CKL - Compromised Key List
- COI - Community of Interest
- CRC - Cyclic Redundancy Check
- CSE - Call Setup Encryption
- CTS - Clear to Send
- DCD - Data Carrier Detect
- DER - Distinguished Encoding Rules
- DSR - Data Set Ready
- DTE - Data Terminal Equipment
- DTMF - Dual Tone Multi-frequency
- DTR - Data Terminal Ready
- DTX - Discontinuous (Voice) Transmission
- ECMQV - Elliptic Curve Menezes-Qu-Vanstone
- ECU - End Cryptographic Unit (e.g., STE)
- EIA - Electronic Industries Association
- EOM - End of Message
- EOT - End of Transmission
- EKMS - Electronic Key Management System
- ESC - Escape
- FC - Frame Count
- FCT - Force Continuous Transmission
- FDX - Full Duplex
- FEC - Forward Error Control/Forward Error Correction
- FF - FIREFLY
- FIPS - Federal Information Processing Standard
- FNBDT - Future Narrowband Digital Terminal
- FSVS - Future Secure Voice System

193	GRFE	-	Generic Rekey Front End
194	GRPDU	-	Generic Rekey PDU
195	HDX	-	Half Duplex
196	Hz	-	Hertz
197	IP	-	Internet Protocol
198	ISDN	-	Integrated Services Digital Network
199	ISO	-	International Standards Organization
200	ITU-T	-	International Telecommunication Union - Telecommunication
201			Standardization Sector
202	IV	-	Initialization Vector
203	IWF	-	Interworking Function
204	kbps	-	kilobits per second
205	KG	-	Key Generator
206	KMC	-	(STU-III) Key Management Center
207	KMF	-	Key Management Facility - Synonymous with CF
208	KMID	-	Key Material Identifier
209	KP	-	Key Processor
210	KPF	-	Key Processing Facility
211	LIT		Line Interface Terminal
212	LMD	-	Local Management Device
213	lsb	-	Least Significant Bit
214	MCS	-	Multipoint Cryptosync message
215	MELP	-	Mixed Excitation Linear Prediction
216	MELPe	-	Mixed Excitation Linear Prediction - Enhanced
217	MER	-	Minimum Essential Requirement
218	MID	-	Message Identifier
219	ms	-	millisecond
220	msb	-	Most Significant Bit
221	NATO	-	North Atlantic Treaty Organization
222	NIST	-	National Institute of Standards & Technology
223	PCM	-	Pulse Code Modulation
224	PDU	-	Protocol Data Unit
225	PLC	-	Partial Long Component
226	PN	-	Pseudo-Noise
227	POTS	-	Plain Old Telephone Service
228	PPK	-	Pre-Placed Key
229	PSTN	-	Public Switched Telephone Network
230	RT	-	Reliable Transport
231	RTS	-	Request to Send
232	SCIP	-	Secure Communication Interoperability Protocol
233	SCN	-	Specification Change Notice
234	sec	-	second
235	SM	-	Sync Management frame
236	SOM	-	Start of Message
237	SPI	-	Security Parameters Index
238	STE	-	Secure Terminal Equipment

239	STU	-	Secure Telephone Unit
240	TBD	-	To Be Defined
241	TBSL	-	To Be Supplied Later
242	TEK	-	Traffic Encryption Key
243	TIA	-	Telecommunications Industry Association
244	VAD	-	Voice Activity Detection
245	VAF	-	Voice Activity Factor
246	w/o	-	Without

247
248

1.5 Applicable Documents

249
250

The following documents are applicable to the extent specified in the remainder of the Signaling Plan. Where conflicts may exist, the order of precedence shall be to this specification, then to other SCIP-related specifications, then to NSA specifications, Industry standards, Federal and DoD standards, and National and International standards, in that order.

251
252
253
254
255

The documents controlled by the NSA are identified as the latest known issue in existence at the time of the issue date of this Signaling Plan. These documents may be changed through Specification Change Notices through a configuration controlled process. Industry, National, and International standards listed shall be considered the binding version unless this list of applicable specifications is changed through a Specification Change Notice (SCN) issued through the accompanying configuration control procedures.

256
257
258
259
260
261
262

This Signaling Plan references the Cryptography Specifications, listed in Section 1.5.1, throughout the document. When a Cryptography Specification is referenced, the signaling requirement is supported by the cryptographic suite specified in that Cryptography Specification. When a Cryptography Specification is not referenced, the signaling requirement is not applicable to the cryptographic suite specified in that Cryptography Specification.

263
264
265
266
267
268
269

1.5.1 NSA Documents

270
271

SCIP-215, Revision 2.0
U.S. Secure Communication Interoperability Protocol (SCIP) over IP
Implementation Standard and Minimum Essential Requirements (MER) Publication
3 October 2007

272
273
274
275
276

SCIP-216, Revision 2.0
Minimum Essential Requirements (MER) for V.150.1 Gateways Publication
2 November 2007

277
278
279
280

SCIP-230, Revision 3.1
Secure Communication Interoperability Protocol
Cryptography Specification
7 February 2007

281
282
283
284

285
286 SCIP-231, Revision 1.2
287 Secure Communication Interoperability Protocol
288 ECMQV/AES Cryptography Specification
289 19 December 2007
290
291 SCIP-232, Revision 1.0
292 Secure Communication Interoperability Protocol
293 ECMQV/AES – NATO Cryptography Specification
294 31 May 2007

295
296 EKMS 218
297 Generic Rekey Front End System Requirements Specification
298 Baseline Version (RFC-2001-010R2)
299 13 December 2001
300

301 **1.5.2 Industry Standards**

302
303 EIA/TIA-232-E
304 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment
305 Employing Serial Binary Data Interchange
306 July, 1991
307
308

309 **1.5.3 International and National Standards**

310
311 CCITT Recommendation Z.100
312 Functional Specification and Description Language (SDL)
313 (Melbourne 1988)
314 Fascicles X.1 - X.5
315

316
317 ITU-T Recommendation G.729
318 Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-
319 Prediction (CS-ACELP)
320 03/96
321

322
323 ITU-T Recommendation G.729 Annex D
324 6.4 kbit/s CS-ACELP Speech Coding Algorithm
325 09/98

326
327 ITU-T Recommendation G.729 Annex F
328 Reference Implementation of G.729 Annex B DTX Functionality for Annex D
02/00

329
330 ISO/IEC 8824
331 Information Processing Systems - Open Systems Interconnect - Abstract Syntax
332 Notation One (ASN.1),
333 Second Edition, International Standards Organization,
334 1990
335
336 ISO DIS 8825
337 Information Processing Systems - Open Systems Interconnect - Specification of Basic
338 Encoding Rules for Abstract Syntax Notation One (ASN.1),
339 Second Edition, International Standards Organization,
340 1990

341

342

343 **1.5.4 Federal and DoD Standards**

344

345 MIL-STD-3005
346 Analog-to-Digital Conversion of Voice by 2400 Bit/Second Mixed Excitation Linear
347 Prediction (MELP)
348 20 December 1999

349

350

351 **1.5.5 NATO Standards**

352

353 NATO STANAG 4591
354 NATO Interoperable Narrow Band Voice Coder [MELPe]
355 In ratification – date TBD

356

357

358 **1.5.6 Other Relevant Technical Papers**

359

360 Discontinuous Transmission for MELP in FNBDT
361 Richard A. Dean and Lynn M. Supplee
362 October 22, 1998

363

364 Description of a Decoder for the (160, 128) $t = 4$ Binary BCH Code
365 [ITB:98-027]
366 Arnold M. Michelson

367

368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412

1.6 Signaling Plan Overview

The SCIP signaling provides the capability for the user to communicate with other compatible instruments using a secure overlay on a variety of digital networks. It includes the capability for both clear and secure communications, defined respectively as clear traffic and secure traffic. When the far-end terminal is a standard commercially available telephone, communication proceeds using the techniques and procedures of the underlying network. When the far-end terminal is another SCIP-compatible device, secure communication may proceed using the security capabilities specified herein. The secure modes of operation addressed in this Signaling Plan include both secure voice and secure data. In addition to the signaling for the operational traffic, the Signaling Plan also includes control signaling to establish and coordinate the clear and secure traffic modes of operation and signaling to perform electronic rekey when a call is established to the Electronic Key Management System Central Facility. The abilities to transmit and receive alerting and display information in the clear and secure dial digits are also included.

The Signaling Plan defines several modes of operation. For each mode of operation the minimal signaling that must be used by terminals, that are advertised as SCIP capable, is specified herein. This includes signaling for the “core SCIP functions,” such as secure call setup, that is specified in the main body of this Signaling Plan. However, not all SCIP capable terminals will implement all modes of operation (e.g., there will be data only and voice only terminals), and the MERs for a specific terminal will be defined elsewhere.

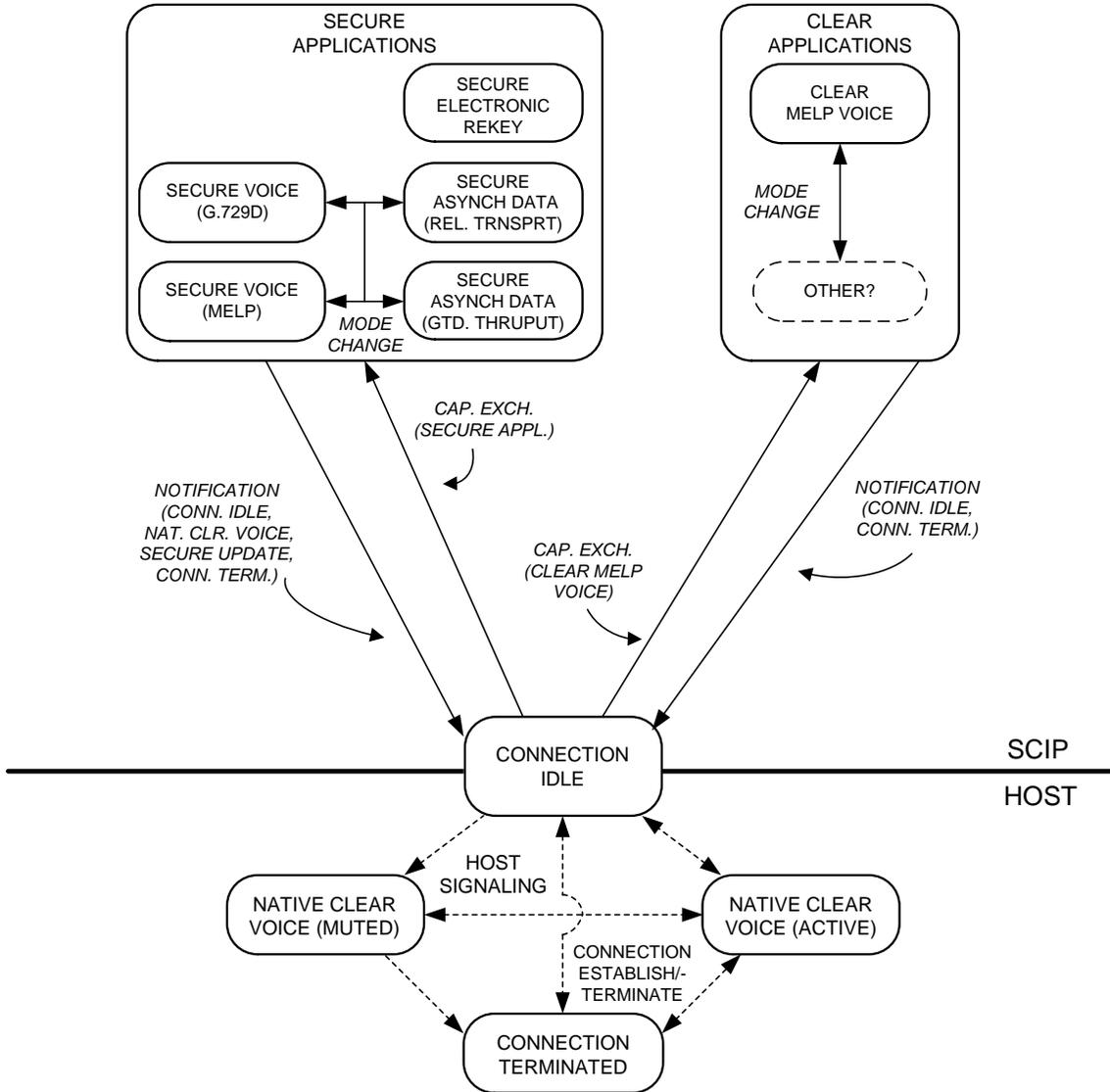
The Signaling Plan is intended to be “network independent,” that is, the signaling is designed to operate over a variety of narrowband, wideband, and protected digital networks. Requirements that are dependent on the network to which the terminal is connected, i.e., call establishment procedures and characteristics of the physical interface to the network, are specified in appendices to the Signaling Plan.

1.6.1 SCIP Application State Diagram

Figure 1.6-1 provides a high level conceptual application state diagram of a terminal that incorporates SCIP signaling.

The terminal starts in a Connection Terminated state in which there is no communication path to the far end. Before the signaling defined in this Signaling Plan may be executed, a clear data path, which will be used to carry the SCIP messages, must be established between the two ends. The state in which such a clear data path exists, but over which no SCIP application signaling is in process, is known as Connection Idle. (Note that while the term “Connection Idle” is used to name this state in this Signaling Plan, it is likely that a different name will be used for it in documents that define the native signaling of the terminal.) Of course the native signaling in the terminal may be used to invoke other underlying “native” functions (e.g., Native Clear Voice) as well. When a terminal transitions from a secure application to Native Clear Voice, the user must acknowledge the transition. Therefore, the terminal remains in the Native Clear Voice (Muted)

413 state until the user acknowledges the transition, and it then switches to the Native Clear Voice
414 (Active) state.
415
416



417
418
419 **Figure 1.6-1 SCIP Application State Diagram - Point-to-Point**
420

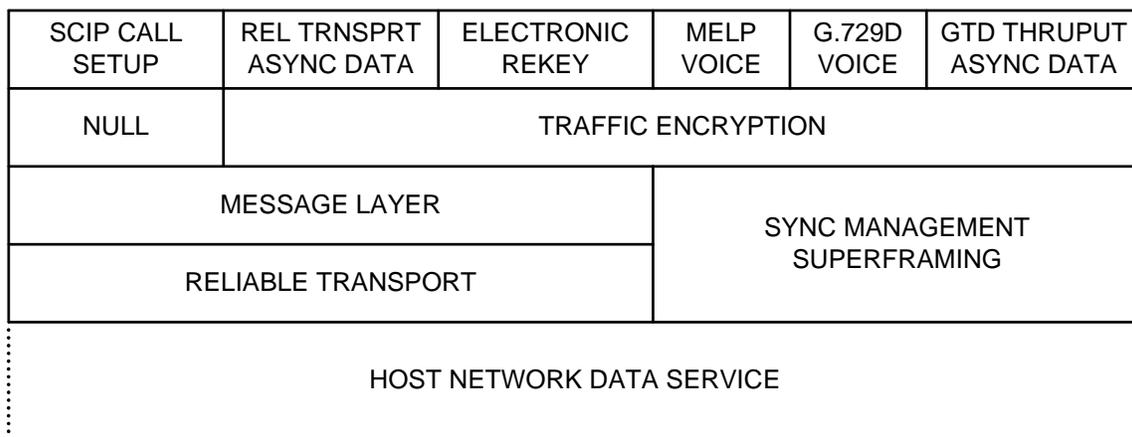
421 SCIP applications can be accessed from Connection Idle. Standard SCIP clear voice
422 applications (of which only Clear MELP Voice is currently defined) are chosen using the first
423 SCIP call setup exchange, the Capabilities Exchange. In addition to the Capabilities Exchange,
424 further exchanges are required to negotiate the parameters for standard SCIP secure applications.
425 The choice of vendor unique SCIP applications also starts with a Capabilities Exchange, after
426 which either the standard SCIP call setup signaling or vendor defined signaling may be used.
427 Native functions may be executed directly from this state using native host signaling, or may be
428 chosen using the Capabilities Exchange (in which case control passes back to Connection Idle
429 and through Connection Idle to the chosen native function).

430
431 For changing between secure applications that use the same traffic key or between SCIP clear
432 applications, a Mode Change function is provided. Transitions to other applications are made by
433 returning to Connection Idle. If a transition from a SCIP application to a common native
434 function is desired, this is indicated in the Notification Message. If a transition to a SCIP mode
435 is desired, an ensuing Capabilities Exchange is executed. For vendor unique mode transitions,
436 the terminals may use the standard mechanisms defined in this Signaling Plan or they may use
437 vendor unique methods for executing the transitions.

438
439 To terminate the call from a standard SCIP application, a Notification Message is used to return
440 to Connection Idle with an indication that the underlying native mechanism be used to close the
441 underlying clear data path and return to the Connection Terminated state.

442
443
444 **1.6.2 SCIP Protocol Layer Diagram**

445
446 Figure 1.6-2 shows a protocol layer diagram for the SCIP secure applications and secure call
447 setup. The Clear MELP Voice application is not shown in the diagram; however, it is exactly
448 like the Secure MELP Voice application, but without the encryption layer.

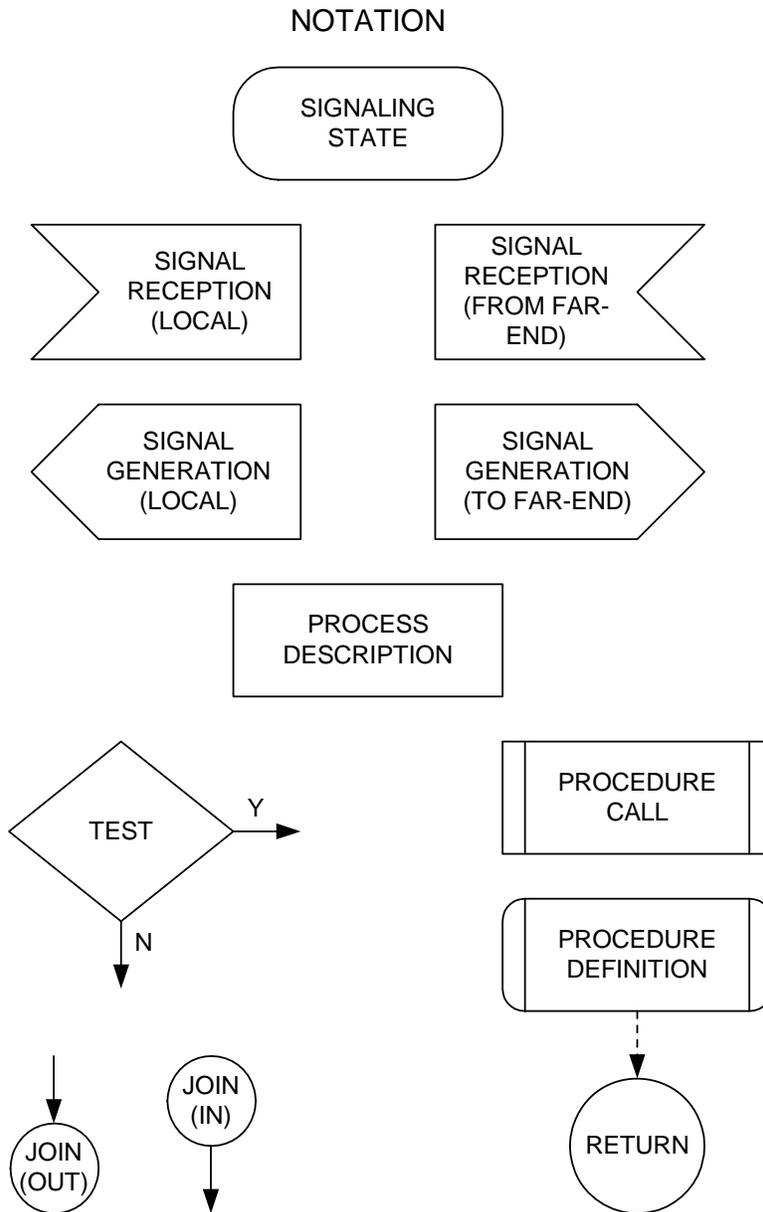


451
452
453 **Figure 1.6-2 SCIP Protocol Layer Diagram - Point-to-Point**

454
455
456
457
458
459
460

1.7 Document Conventions

The process diagram symbols used in the figures in this Signaling Plan are based on the process diagram symbols defined in ITU Z.100 and are shown in Figure 1.7-1.



461
462
463
464

Figure 1.7-1 Process Diagram Symbols

465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488

THIS PAGE INTENTIONALLY LEFT BLANK.

2.0 SCIP SIGNALING – Point-to-Point Operation

This section defines the SCIP call setup and control signaling for point-to-point operation. Section 2.1 specifies SCIP Transport Layer signaling, message framing, Transport Layer messages, and the Transport Layer protocol rules. Section 2.2 specifies call setup signaling including the Capabilities Exchange, which is always required, and the Parameters/Certificate Exchange, F(R) Exchange, and Cryptosync Exchange, which are used to invoke a SCIP secure application. Section 2.3 specifies the SCIP call control signaling including the Notification Message, the Mode Change exchange, and the Two-Way Resync exchange. Section 2.4 specifies SCIP signaling timeouts, and Section 2.5 specifies signaling constants.

2.1 SCIP Message Transport

The SCIP MER message transport incorporates a number of error control mechanisms to facilitate reliable delivery of signaling messages to the far-end terminal. Signaling transmissions start with a Start of Message (SOM) and end with an End of Message (EOM) pattern and will be referred to herein as “frame groups”. A frame group is composed of frames, each of which is protected by a binary BCH code used for forward error correction (FEC) and a cyclic redundancy check (CRC) code. Recovery from transmission errors that cannot be corrected by the FEC is provided through the use of a combination of positive acknowledgment and selective reject on a frame-by-frame basis. A Retransmission Timer provides protection for the cases where an entire frame group is lost or does not arrive at the far-end terminal in a recognizable form. Finally, a sliding window function, 127 frames in length, is used to control transmissions.

2.1.1 The MER-OC Message Transport Option and the Branch Point Mechanism

Sections 2.1.2 through 2.1.8 specify a MER message transport that all SCIP terminals must implement. Additionally, alternate MER-OC message transports may be defined and implemented.

If a developer chooses to implement a MER-OC message transport, a timeout based branch transport mechanism must also be implemented. The timer shall be started after an end-to-end connection has been established. Through the branch transport mechanism, the MER-OC terminal shall fall back to the MER message transport unless it can determine, prior to the expiration of the timeout, that the far-end terminal will successfully establish a compatible MER-OC mode. For human factors reasons, the MER-OC timeout should be kept as short as possible but shall be long enough to be compatible with establishing the fallback MER message transport prior to the expiration of the First Message Timer (see Section 2.4).

Note that, except for the extra delay, a MER-only terminal should be unaware of the far end's attempt to establish a MER-OC transport. Note also that two terminals have a second chance to establish a compatible MER-OC transport by offering such developer defined Operational

534 Modes as part of the Capabilities Exchange. MER-OC message transports are not further
535 defined in this document.

538 **2.1.2 Message Transport Timelines**

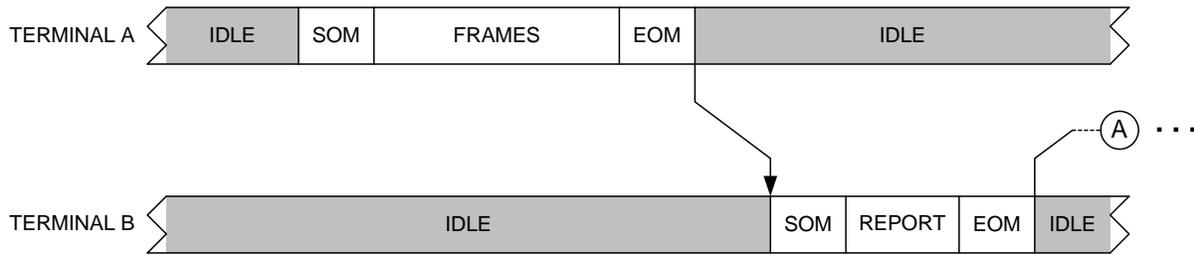
539
540 Throughout this document, references are made to “framed” and “full bandwidth” traffic. In the
541 context of SCIP-210, “framed” traffic refers to traffic that is formatted with the framing
542 information shown in Figure 2.1-2, starting with a SOM and ending with an EOM. In contrast,
543 “full bandwidth” traffic refers to application traffic that is transmitted with only sync
544 management information added as specified in Sections 3.3 and 3.4.2. It does not include a
545 leading SOM and a trailing EOM, although it should be noted that there may be other layers of
546 framing provided by the underlying network. Full bandwidth traffic is always preceded by the
547 START pattern.

548
549 It should also be noted that the transmit and receive channels of a terminal operate
550 independently. This means that if a terminal receives a START, its receive channel will be in
551 full bandwidth traffic, but its transmit channel will not be in full bandwidth traffic until it
552 transmits a START. The result is that during transition periods of entering or exiting full
553 bandwidth traffic, a terminal may in fact be operating with both framed and full bandwidth
554 traffic.

555
556 An example transport signaling timeline for transmitting a frame group using SCIP point-to-
557 point signaling when in framed traffic is shown in Figure 2.1-1(a). This figure shows
558 transmission of a frame group for which some of the frames are received with uncorrectable
559 errors. The frames received with uncorrectable errors are retransmitted and received correctly on
560 the second attempt.

561
562 The Transport Layer at Terminal A receives a message from the Message Layer, formats it into
563 frames, adds an SOM and an EOM, and transmits the frame group. Terminal B receives the
564 frame group and executes error detection and correction. In the case shown, some of the frames
565 are received with uncorrectable errors; therefore, Terminal B formats a REPORT message
566 identifying the frames that contained uncorrectable errors and transmits it. Upon receiving the
567 REPORT message, Terminal A formats the frames that were not received correctly into a new
568 frame group by adding an SOM and an EOM and transmits it. Terminal B receives this frame
569 group, decodes the frames, and finds no uncorrectable errors. Therefore, Terminal B sends a
570 REPORT message indicating that all of the frames contained in the original frame group have
571 been received correctly. The intervals between transmissions are shown as IDLE in Figure 2.1-
572 1(a). This means there is no transmission of data by the SCIP application; however,
573 transmissions may occur on individual links related to handshaking performed by the underlying
574 channel protocols.

576



577

578

579

580

581

Figure 2.1-1(a) Transport Layer Signaling Time Line (Framed)

582

583

584

585

586

An example transport signaling timeline for transmitting a frame group using SCIP point-to-point signaling when in full bandwidth traffic is shown in Figure 2.1-1(b). This figure shows transmission of a frame group for which all of the frames are received successfully on the first attempt. Also, following the frame group transmission and acknowledgment, the terminals remain in framed operation.

587

588

589

590

591

592

593

594

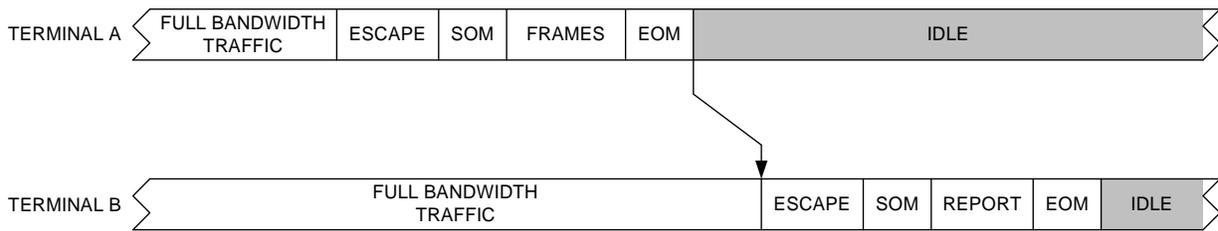
595

596

597

The Transport Layer at Terminal A receives a message from the Message Layer, formats it into frames, adds an SOM and an EOM, and prepares it for transmission. Since the terminals are in full bandwidth traffic, an ESCAPE is transmitted followed immediately by the frame group. Terminal B detects the ESCAPE, switches to framed receiver operation, receives the frame group, and checks it for errors. In the case shown, none of the frames are received with uncorrectable errors. Since Terminal B is still in full bandwidth transmitter operation, it transmits an ESCAPE followed immediately by a REPORT message indicating that all of the frames in the frame group were received successfully. Both terminals are now in framed operation, so the intervals following the transmissions are shown as IDLE.

598



599

600

601

Figure 2.1-1(b) Transport Layer Signaling Time Line (Full bandwidth-to-Framed)

602

603

604

605

606

607

608

609

Another example transport signaling timeline for transmitting a frame group using SCIP point-to-point signaling when in full bandwidth traffic is shown in Figure 2.1-1(c). This figure shows transmission of a frame group for which all of the frames are received successfully on the first attempt. In this example, following the frame group transmission and acknowledgment, the terminals reenter full bandwidth operation.

610

611

612

613

614

615

616

617

618

619

620

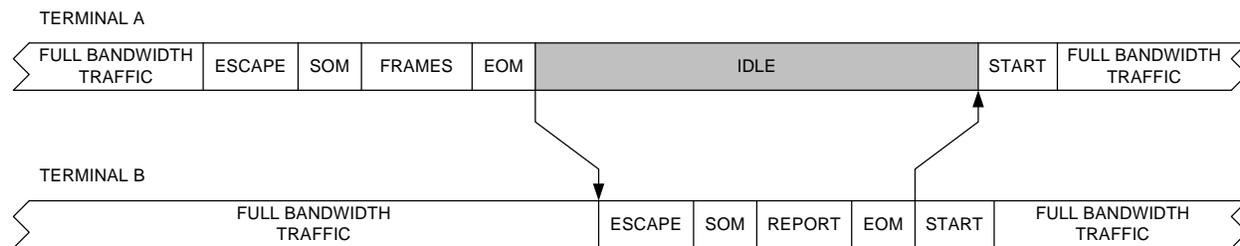
621

622

The Transport Layer at Terminal A receives a message from the Message Layer, formats it into frames, adds an SOM and an EOM, and prepares it for transmission. Like the previous example, the terminals are in full bandwidth traffic, so an ESCAPE is transmitted followed immediately by the frame group. Terminal B detects the ESCAPE, switches to framed receiver operation, receives the frame group, and checks it for errors. Again, none of the frames are received with uncorrectable errors. Since Terminal B is still in full bandwidth transmitter operation, it transmits an ESCAPE followed immediately by a REPORT message indicating that all of the frames in the frame group were received successfully. Following transmission of the REPORT message, Terminal B transmits the START pattern followed by full bandwidth traffic. When Terminal A has received the REPORT message, it transmits the START pattern followed by full bandwidth traffic.

621

622



623

624

625

Figure 2.1-1(c) Transport Layer Signaling Time Line (Full bandwidth-to-Full bandwidth)

626

660
661
662
663
664
665
666
667
668
669
670
671
672

Each message shall be partitioned into 13-octet data segments that are transmitted in order. Octets 1 through 13 shall be placed in the first frame to be transmitted, octets 14 through 26 in the second frame, etc. Any octets left over shall be transmitted in the Message Data field of the final frame, which shall be padded out to 13 octets with padding octets having a value of 0x00. Octets 1 - 13 of the message are placed in octets 10 - 22 of the frame group, octets 14 - 26 of the message are placed in octets 30 - 42 of the frame group, etc. Bits within an octet of the message are placed in the corresponding bit position of the frame, i.e., bit 1 of a message octet is placed in bit 1 of the corresponding octet of the frame, etc.

Table 2.1-1 Frame Group Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
SOM								
0	1	1	1	1	1	1	0	1
•••								•
•••								•
0	1	0	1	0	1	1	0	8
First Frame Frame Count								
b7	b6	b5	b4	b3	b2	b1	b0	9
Message Data								
X	X	X	X	X	X	X	X	10
•••								•
•••								•
X	X	X	X	X	X	X	X	22
CRC								
b8	b9	b10	b11	b12	b13	b14	b15-msb	23
b0-lsb	b1	b2	b3	b4	b5	b6	b7	24
FEC								
b24	b25	b26	b27	b28	b29	b30	b31-msb	25
•••								•
b0-lsb	b1	b2	b3	b4	b5	b6	b7	28
•								•
•								•
•								•
Mth Frame Frame Count								
b7	b6	b5	b4	b3	b2	b1	b0	9+20(M-1)
•••								•
•••								•
X	X	X	X	X	X	X	X	28+20(M-1)
EOM								
1	0	0	0	0	0	0	1	9+20M
•••								•
•••								•
1	0	1	0	1	0	0	1	16+20M

673 M = Number of frames in the frame group

674
675 **2.1.3.1 Start of Message**
676

677 The Start of Message is a 64-bit pseudorandom sequence that begins each transmit frame group.
678 It is designed to allow acceptable detection performance in the anticipated error environments,
679 and to allow the receiver to determine the first bit of the first octet of a frame group. The SOM
680 value is specified in Table 2.5-3. The first frame shall immediately follow SOM transmission.
681

682
683 **2.1.3.2 Frame Count**
684

685 As shown in Figure 2.1-2, the first octet of each frame of a transmitted frame group shall contain
686 the Frame Count. The first frame of the first message transmitted, after initial entry or upon re-
687 entry from a native mode, shall have Frame Count = 0x01. The Frame Count shall be
688 incremented for each subsequent frame transmitted (modulo 256 - with return to 0x01 after
689 0xFF) without regard to frame group boundaries. The Frame Count is not reset upon entry to or
690 exit from a SCIP application. In particular, it shall continue with the next value in sequence
691 following a transition from call setup signaling to a reliable transport application. For Transport
692 Layer control messages (REPORT), the Frame Count shall be set to 0x00 for all frames. This
693 identifies these messages as Transport Layer control messages.
694

Editor's Note: Note that with a k -bit Frame Count, which provides a Frame Count range of 2^k ,
the maximum window size is limited to 2^{k-1} outstanding frames in order to prevent ambiguities.
For this Signaling Plan, a window size of 128 frames outstanding would have been the result of
a one-octet Frame Count field. However, Frame Count = 0x00 is reserved for Transport Layer
control messages, thus a window size of 127 frames outstanding results.

695
696
697 **2.1.3.3 Cyclic Redundancy Check**
698

699 A Cyclic Redundancy Check shall be calculated over the Frame Count and Message Data fields
700 of each frame. The CRC shall be the North American standard CRC-16. Its generator
701 polynomial is $P(x) = x^{16} + x^{15} + x^2 + 1$. The CRC shall be computed as follows. Let $S(x)$ be the
702 polynomial representing the 112 bits (14 octets) of the transport frame beginning with the least
703 significant bit of the Frame Count and extending, in the order that the bits are transmitted,
704 through the most significant bit of the 13th octet of the Message Data field. The least significant
705 bit of the Frame Count is the coefficient of the highest degree term of $S(x)$. The transmitted
706 CRC checksum, $F(x)$, shall be the ones complement of the remainder of $(x^{16}S(x) + x^{112}(x^{15} + x^{14}$
707 $+ x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1))/P(x)$. Note that multiplying
708 $S(x)$ by x^{16} is equivalent to shifting $S(x)$ 16 places to provide the space for the 16-bit CRC parity
709 bits, and adding $x^{112}(x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$
710 to $x^{16}S(x)$ is equivalent to inverting the first 16 bits of $S(x)$. $F(x)$ is then added to $x^{16}S(x)$ forming
711 the 128-bit transport frame, exclusive of the FEC field. The coefficient of the x^{15} term of $F(x)$
712 shall be transmitted immediately following the most significant bit of the 13th octet of the
713 Message Data field (see Table 2.1-1).
714

Editor's Note: Inverting the first 16 bits of $S(x)$ can also be accomplished in a shift register implementation by setting the register to all “ones” initially. This permits the receiver to detect erroneous addition or deletion of zero bits at the leading end of $S(x)$. Complementing the remainder permits the receiver to detect addition or deletion of trailing zeros that may appear as a result of errors. At the receiver, the shift register is again set to all “ones” initially, and the CRC is computed over the received $S(x)$. If the computed and received CRC are the same value, there are no errors.

715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741

2.1.3.4 Forward Error Control

Forward error control shall be implemented with a four error correcting binary BCH code shortened from a natural block length of 255. The block length of the code is 160; there are 128 information bits and 32 check bits per code block. The check bits are computed over the Frame Count, Message Data, and CRC fields, that is, over 128 information bits or 16 octets. The generator polynomial is

$$g(x) = x^{32} + x^{31} + x^{30} + x^{29} + x^{27} + x^{26} + x^{25} + x^{22} + x^{20} + x^{19} + x^{17} + x^{16} + x^{14} + x^9 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1.$$

The check bits for the code shall be computed as follows. Let $I(x)$ be the polynomial representing the 128 bits to be encoded beginning with the least significant bit of the Frame Count and extending, in the order that the bits are transmitted, through the most significant bit of the second octet of the CRC field. The least significant bit of the Frame Count is the coefficient of highest degree in $I(x)$. The transmitted check bits, $R(x)$, shall be calculated as

$$R(x) = (x^{32} I(x)) \bmod g(x).$$

Note that multiplying $I(x)$ by x^{32} is equivalent to shifting $I(x)$ 32 places to provide space for the 32 check bits. $R(x)$ is then added to $x^{32} I(x)$ to form the 160-bit BCH code word. The coefficient of the x^{31} term of $R(x)$ shall be transmitted immediately following the most significant bit of the second octet of the CRC field (which contains the least significant bit of the CRC), and the least significant bit of $R(x)$ shall be transmitted last (see Table 2.1-1).

742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776

2.1.3.5 End of Message

The End of Message is a 64-bit pseudorandom sequence that immediately follows the final frame of each transmitted frame group. It allows the receiving terminal to reliably detect the end of a received frame group in the anticipated error environments. The EOM value is specified in Table 2.5-3. Note that it is the bit-by-bit complement of the SOM. EOM shall be transmitted following the final octet of the final frame of a frame group.

2.1.4 Escape

The ESCAPE sequence is a 256-bit pseudorandom sequence that allows reliable detection in the background of full bandwidth traffic under expected channel conditions. The ESCAPE sequence is used to permit the detection of transmitted frame groups that interrupt full bandwidth traffic. The value of the ESCAPE sequence is specified in Table 2.5-3.

Transmit and receive processing for the ESCAPE are shown in Figure 2.1-3. When a terminal is transmitting full bandwidth traffic (entry into full bandwidth traffic is described in Section 3), it shall precede frame group transmissions with an ESCAPE. (Whether or not the far-end terminal receiver has entered full bandwidth traffic is irrelevant. If it has entered full bandwidth traffic, the ESCAPE is necessary. If it has not yet done so, the ESCAPE will be ignored, and the SOM will be detected.)

When transmission of a frame group, which can be either a Call Control or a REPORT message, is invoked during full bandwidth transmission, the terminal shall stop transmitting full bandwidth traffic, transmit the ESCAPE sequence, and enable framing. The terminal shall then format and transmit the requested frame group as specified in Section 2.1.6. (Note that the state of the terminal's receiver remains unchanged.)

A terminal that receives an ESCAPE sequence during full bandwidth reception shall enable framed reception and process the incoming frame group as specified in Section 2.1.7. (Note that the state of the terminal's transmitter remains unchanged.)

777

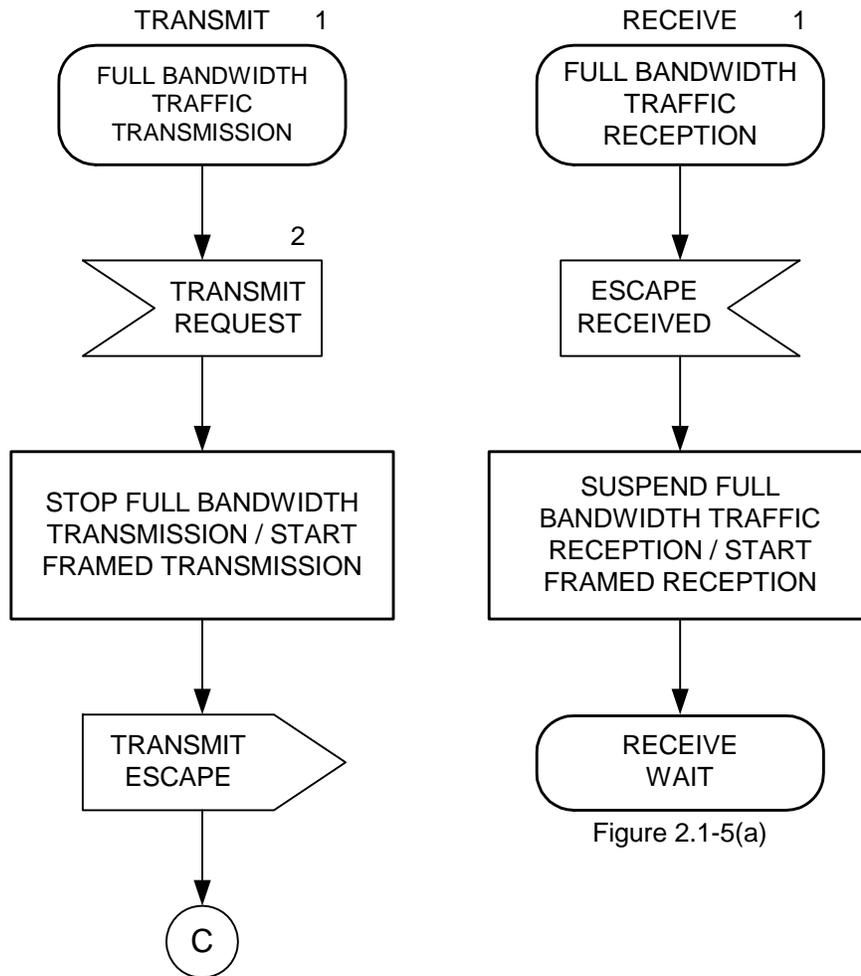


Figure 2.1-5(a)

Figure 2.1-4(a)

NOTES:

1. A transmitting terminal is considered to be in full bandwidth traffic if it has transmitted a START. A receiving terminal is considered to be in full bandwidth traffic if it has received a START.
2. Can be either call control messages or REPORT.

778

779

780

781

782

Figure 2.1-3 ESCAPE Processing

783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802

2.1.5 Transport Layer Control Messages

Transport Layer control messages are messages that are exchanged between peer Transport Layers and are not passed up to higher layers. They shall be transmitted with the Frame Count field set to 0x00 to distinguish them from messages intended for higher layers. The REPORT message is the only Transport Layer control message currently defined for SCIP signaling. The REPORT message shall have a length of one frame.

2.1.5.1 REPORT Message

The REPORT message provides both the capability to acknowledge successful reception of contiguous frames of received messages and the capability to selectively reject individual frames of received messages. It contains an ACK'ed Frame Count field that corresponds to the last consecutive message frame that was received successfully (either with no errors or with correctable errors) and NAK'ed Frame fields corresponding to a maximum of seven message frames that were lost (either not received or received with uncorrectable errors). Conditions under which the REPORT message may be transmitted are specified in Section 2.1.5.1.2.

803
804
805
806
807
808
809
810

2.1.5.1.1 REPORT Message Format

The format of the REPORT message is shown in Table 2.1-2.

Table 2.1-2 REPORT Message Data Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0-lsb	2
Message Length								
0-msb	0	0	0	0	0	0	0	3
0	0	0	0	1	0	1	1-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
ACK'ed Frame Count								
X	X	X	X	X	X	X	X	6
NAK'ed Frame #1								
X	X	X	X	X	X	X	X	7
NAK'ed Frame #2								
X	X	X	X	X	X	X	X	8
NAK'ed Frame #3								
X	X	X	X	X	X	X	X	9
NAK'ed Frame #4								
X	X	X	X	X	X	X	X	10
NAK'ed Frame #5								
X	X	X	X	X	X	X	X	11
NAK'ed Frame #6								
X	X	X	X	X	X	X	X	12
NAK'ed Frame #7								
X	X	X	X	X	X	X	X	13

811
812

- For the REPORT message, the value of the MID is 0x0020.

- 813 • The Message Length contains the actual length of the message body (including the
814 length of the Message Length field itself but not including the length of the MID
815 field) in octets. The value of the field is an unsigned binary integer with the high
816 order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4. It is always
817 set to 0x000B, since this is a fixed length message
- 818 • For the version of the REPORT message defined in this version of the Signaling Plan,
819 the value of the Message Version field is 0x00.
- 820 • The ACK'ed Frame Count field contains the Frame Count corresponding to the most
821 recent frame being acknowledged.
- 822 • The NAK'ed Frame fields contain the Frame Counts corresponding to up to seven
823 frames being negatively acknowledged (i.e., indicating that the frame was not
824 received successfully). If fewer than seven frames are to be NAK'ed, the remaining
825 (unused) NAK'ed Frame fields shall be set to 0x00. Also, if more than seven frames
826 are to be NAK'ed, multiple REPORT messages shall be transmitted.

827 828 829 **2.1.5.1.2 Conditions for REPORT Message Transmission**

830
831 The REPORT message shall be transmitted to indicate both successful (either error-free or with
832 correctable errors) reception of contiguous message frames and lost message frames (either not
833 received or received with uncorrectable errors). Except after a frame group containing Transport
834 Layer frames (i.e., frames with Frame Count 0x00) where it should not be sent, the REPORT
835 message shall be queued for transmission whenever an EOM or an unexpected SOM (indicating
836 that a previous EOM was lost) is received. In the transmit queue, REPORT messages have
837 priority over messages received from the higher layers and shall be transmitted first.

838
839 The ACK'ed Frame Count field shall be set to the Frame Count of the last frame of contiguous
840 frames received successfully. That is, the ACK'ed Frame Count indicates that all frames in the
841 window up to and including the indicated frame have been received successfully. In the case
842 where the first frame upon entry from Native Mode is received in error, the ACK'ed Frame
843 Count shall be set to 255 (i.e., to 0xFF).

844
845 If there are uncorrectable errors in the received frames (as detected by failure of FEC decoding
846 and/or CRC verification), the NAK'ed Frame fields shall be populated with their Frame Counts.
847 The NAK'ed Frame fields serve as a request to the far-end terminal for retransmission of the
848 indicated frames. The NAK'ed Frame fields shall be set to the Frame Counts of up to seven
849 frames either not received or received with uncorrectable errors. NAK'ed Frames included
850 within a REPORT message shall be in Frame Count sequence (e.g., frame 10 before frame 11,
851 frame 255 before the next frame 1) with the first Frame Count appearing in octet 7. If more than
852 seven frames are to be NAK'ed, an alternative to sending multiple REPORT messages that
853 provide the capability for NAK'ing the frames individually is to request that all frames,
854 beginning with the first frame to be NAK'ed, be retransmitted (i.e., request that the transmitter
855 go back to this frame and restart the transmission). This shall be accomplished by setting all
856 seven NAK'ed Frame fields in the REPORT message to the Frame Count of the frame at which
857 the retransmission is to start.

859 If more than seven frames are not received or are received with uncorrectable errors and multiple
860 REPORT messages are created and transmitted, the first REPORT message shall contain the
861 seven lowest (in Frame Count sequence) NAK'ed Frame Counts, the next REPORT message
862 shall contain the next seven lowest NAK'ed Frame Counts and so on until no NAK'ed Frames
863 remain. As in the case of a single REPORT message, the NAK'ed Frame Counts included
864 within each REPORT message shall be in Frame Count sequence.

865
866 If multiple REPORT messages are waiting in the transmit queue due to a busy transmitter, the
867 information may be consolidated and transmitted as a single REPORT message. Also, if frames
868 are received that have been previously acknowledged (indicating loss of a previous REPORT
869 message) and these frames are subsequently received with uncorrectable errors, they will not be
870 negatively acknowledged, but instead shall be acknowledged again and then discarded, as they
871 have previously been received successfully.

872

Editor's Note: Note that while the specifications dictate when the REPORT must be sent, there are no restrictions concerning the transmission of additional REPORTs. Additional REPORTs may be sent at the discretion of the implementer. For example, a terminal may transmit a REPORT message prior to a timeout during which no frames are received. This allows frames received successfully to be acknowledged in the case where the final portion of a message, including the EOM, is lost during transmission and no subsequent transmissions occur during the timeout interval. The "retransmit starting at frame N" capability may be used to avoid a timeout, e.g., where a transmission disturbance has caused frame alignment to be lost, i.e., all received frames following the disturbance are failing FEC/CRC processing.

873

874

875 **2.1.5.1.3 Processing for REPORT Message Reception**

876

877 The NAK'ed Frame fields of the REPORT message indicate that specific frames have not been
878 received successfully. Note that the NAK'ed Frame fields may be empty (i.e., filled with all
879 0's), in which case processing in addition to that specified below for the ACK'ed Frame Count is
880 not necessary. Upon or after receipt of one or more REPORT messages containing NAK'ed
881 Frame Counts, a terminal shall format one or more frame groups, as defined in Section 2.1.3,
882 containing only those frames indicated in the NAK'ed Frame fields, and shall transmit them to
883 the far end. Within the retransmission, frames shall be ordered in Frame Count sequence (e.g.,
884 frame 10 before frame 11, frame 255 before the next frame 1). A terminal receiving a REPORT
885 message with all seven NAK'ed Frame fields set to the same value shall go back to the frame
886 indicated in the NAK'ed Frame fields and restart transmitting frame groups. The retransmission
887 timer (Section 2.1.6.3) shall be restarted (from its initial value) immediately upon transmission
888 of the EOM following the retransmitted (NAK'ed) frames.

889

890 The ACK'ed Frame Count field of the REPORT message indicates that all frames up to and
891 including the ACK'ed Frame Count have been received successfully by the far-end terminal.
892 The terminal receiving the REPORT can therefore move the start of its transmit window ahead to
893 the frame following the ACK'ed Frame Count, discarding the frame corresponding to the
894 ACK'ed Frame Count and all previous frames. Note that frames shall only be removed from the
895 transmit window after they have been acknowledged; that is, a REPORT message with an

896 ACK'ed Frame Count greater than or equal to the Frame Counts of all frames removed must
897 have been received. The Retransmission Timer shall also be stopped when a REPORT message
898 is received that acknowledges all outstanding frames within the transmit window.
899

900 It should be noted that while a transmitting terminal is required to send a REPORT message
901 upon receipt of an EOM or an unexpected SOM (indicating that a previous EOM was lost) (see
902 Section 2.1.5.1.2), REPORT messages may also be transmitted at other times. Therefore, a
903 terminal shall accept and process REPORT messages as they are encountered in the received
904 frame groups.
905

906 **2.1.6 Message Transmission**

907

908 The Message Transmission function is shown in Figure 2.1-4. The processing of requests to
909 transmit messages (both messages requested by the higher layers and REPORT messages that are
910 internal to the Transport Layer) is discussed in Section 2.1.6.1. Actions taken by the Message
911 Transmission function on receipt of a REPORT message are discussed in Section 2.1.6.2.
912 Actions to be taken on a Retransmit Timeout are discussed in Section 2.1.6.3. A window size of
913 127 is used, i.e., up to 127 unacknowledged frames may be outstanding.
914

915 **2.1.6.1 Transmit Request**

916

917 This section addresses the transmission of messages when requested by the higher layers
918 (including the SCIP Call Setup messages discussed in Section 2.2, the SCIP Call Control
919 messages discussed in Section 2.3, and the framed SCIP Reliable Transport application messages
920 discussed in Sections 3.4.1 and 4.2). It also addresses the transmission of REPORT messages
921 (discussed in Section 2.1.5.1) when requested by the Message Reception function (discussed in
922 Section 2.1.7).
923

924 Messages received from the higher layers shall be transmitted in the order in which the requests
925 for transmission are made. When both are awaiting transmission, REPORT messages shall be
926 transmitted before messages received from the higher layers.
927

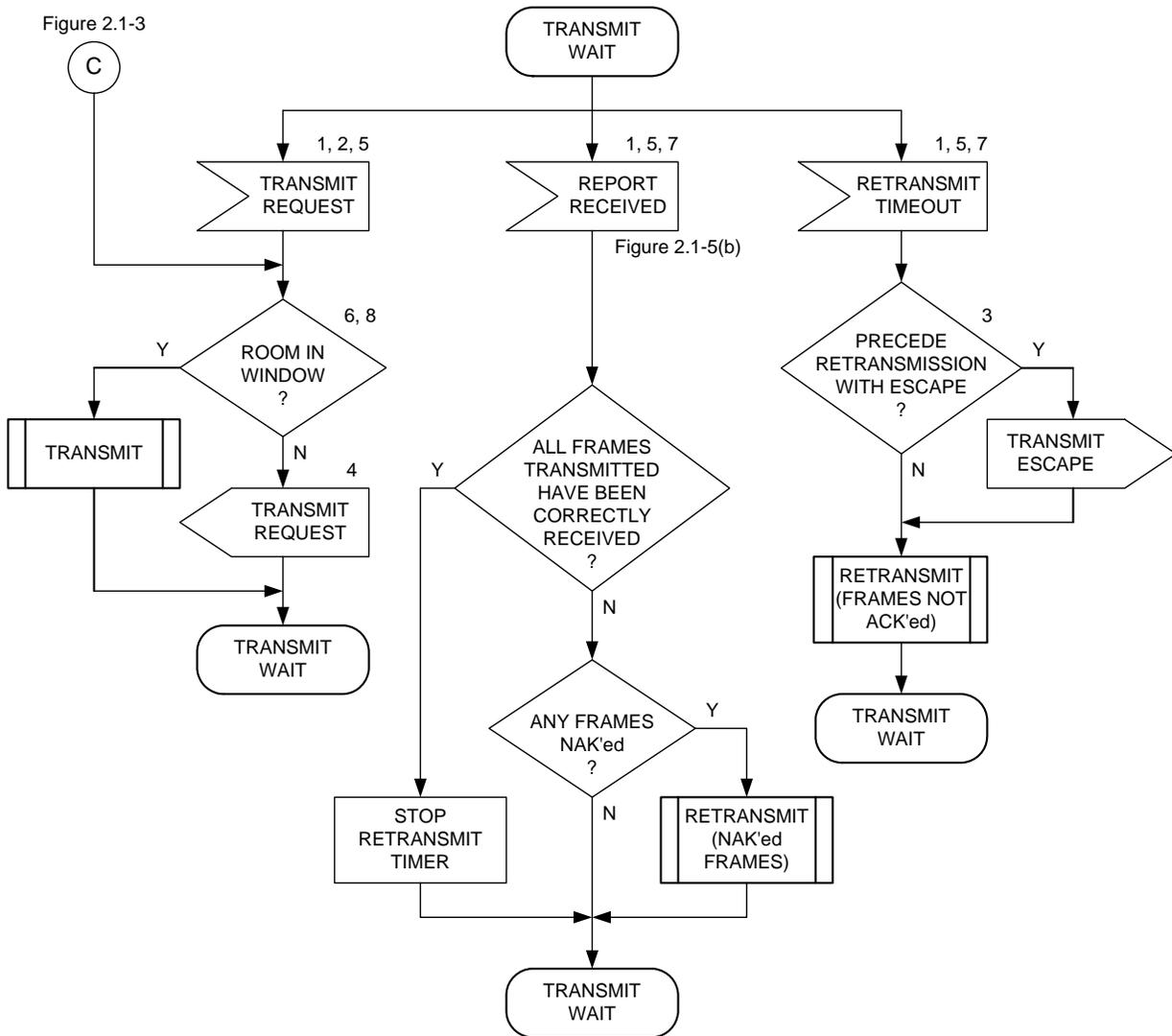
928 When transmission of a message is requested by the higher layers, the Message Transmission
929 function shall check to see if room exists in the window. The window is full if the Frame Count
930 of the next frame to be transmitted minus the Frame Count of the last acknowledged frame,
931 modulo 255, is 128 (i.e., if the difference modulo 255 is 128). However, REPORT messages
932 may be transmitted even if the window is full. If the window is full and the message is not a
933 REPORT, the message is retained. If the window is not full or if the message is a REPORT, a
934 frame group shall be transmitted. In the case where the message being transmitted is not a
935 REPORT, the frame group may contain up to as much of the message as will fit in the window,
936 and the remainder of the message will be retained. If the window is full, the retained message
937 (or the retained portion of a partially transmitted message) shall be transmitted when the window
938 is no longer full.
939
940
941

942 The frame group is formatted as specified in Section 2.1.3. An SOM is transmitted first. Then,
943 while the window constraint permits, message frames are transmitted, followed by the EOM.
944 Frame groups may contain frames from one or more messages. If an entire message does not fit
945 in the current window, that part of the message not transmitted is retained and shall be
946 transmitted when the window is no longer full. When frame transmission is stopped due to a full
947 window, the last frame transmitted shall be followed by an EOM. The next transmission shall
948 then begin with an SOM. Immediately upon transmission of the frame group EOM, unless the
949 message was a REPORT, the Retransmit Timer (Section 2.1.6.3) will be (re)initialized to its
950 initial value and (re)started so that the frames may be retransmitted if no REPORTs are received.
951

Editor's Note: Note that although this specification dictates certain times when frame groups must be terminated (e.g., a full transmit window), other frame groups may be terminated at the transmitter's discretion. A frame group may be any length ≥ 1 frame. Any transmission must be a complete frame group.

952
953 If the transmission occurs subsequent to a transmitted START (i.e., during full bandwidth
954 traffic), the frame group will be preceded by an ESCAPE as specified in Section 2.1.4.
955

956



NOTES:

1. This path includes the case where the event was received prior to entering the Transmit Wait state.
2. See Figure 2.1-3 if the event is recognized and processed during full bandwidth traffic. Transmit Wait, being part of framed operation, is not available during full bandwidth traffic.
3. The condition for retransmission of the ESCAPE is that the ESCAPE was transmitted initially and no REPORTs have since been received. Under this condition the transmitter assumes the receiver is still in full bandwidth traffic and has not reentered framed operation.
4. Queue the request for transmission at a later time.
5. The REPORT Received path is described in Section 2.1.5.1.3, and the Retransmit Timeout path is described in Section 2.1.6.3. The Transmit Request is described in Section 2.1.5.1.2 for REPORTs and in Section 2.1.3 for other Messages.
6. REPORT is always transmitted.
7. If both are pending, the REPORT Received is processed before the Retransmit Timeout. The REPORT Received processing may eliminate the need to perform the Retransmit Timeout processing if NAK'ed frames are retransmitted or if all outstanding frames have been correctly received.
8. A window is full when the Frame Count of the next new frame that will be transmitted minus the ACK'ed Frame Count in the last REPORT received, modulo 255, is 128.

957

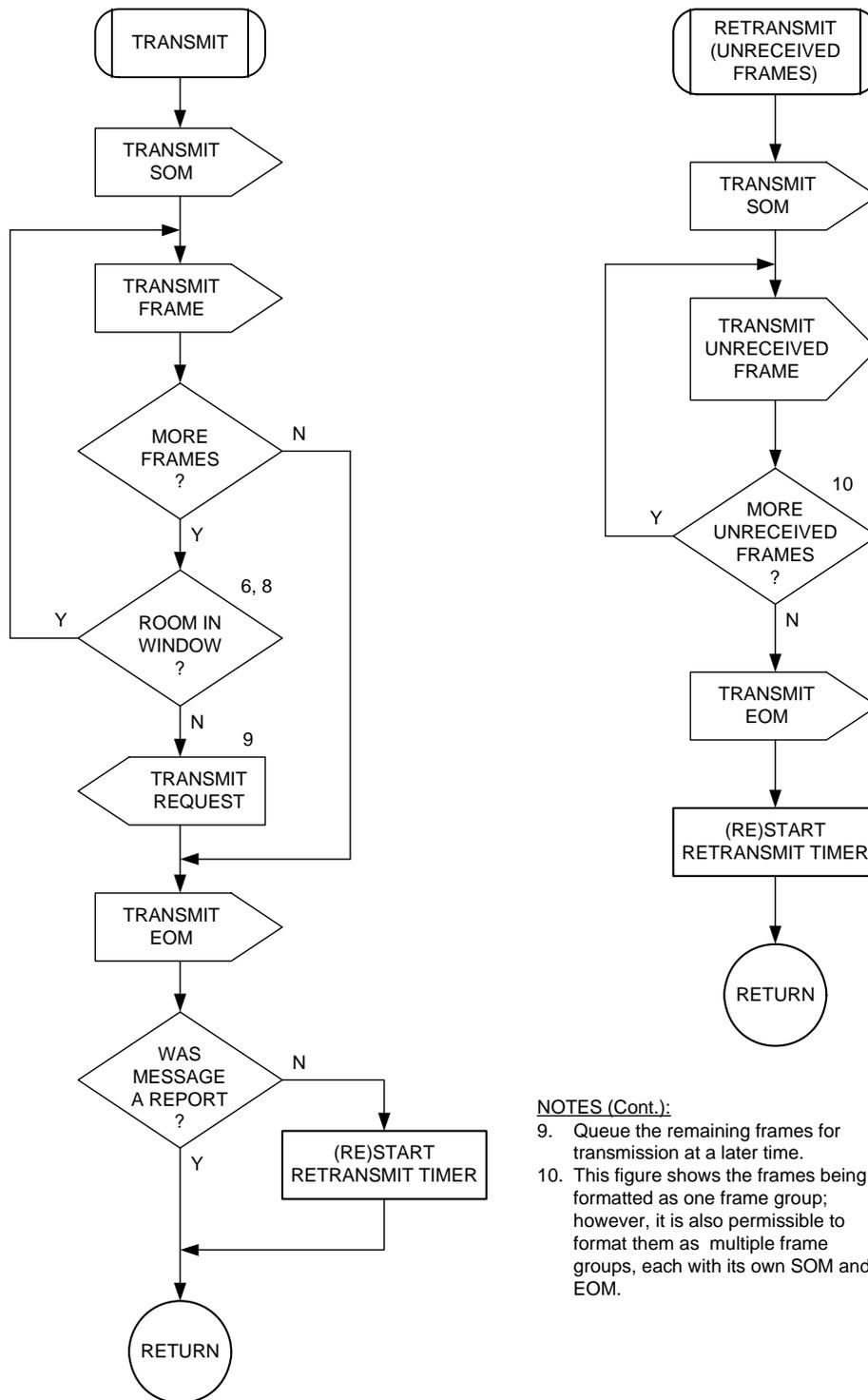
958

959

960

Figure 2.1-4(a) Message Transmission

961



NOTES (Cont.):

- 9. Queue the remaining frames for transmission at a later time.
- 10. This figure shows the frames being formatted as one frame group; however, it is also permissible to format them as multiple frame groups, each with its own SOM and EOM.

962
 963
 964
 965

Figure 2.1-4(b) Message Transmission (Cont.)

966
967 **2.1.6.2 Transmitter Actions on Receipt of a REPORT**
968

969 Upon receipt of a REPORT message, the Message Transmission function will proceed as
970 follows.

971
972 If all frames that have been transmitted are acknowledged by the REPORT, the Retransmit
973 Timer is stopped.

974
975 If frames are NAK'ed by the REPORT, a frame group containing the NAK'ed frames is
976 formatted as specified in Section 2.1.5.1.3 and is transmitted. An SOM is transmitted first. This
977 is followed by the NAK'ed frames and by an EOM. Immediately upon transmission of the frame
978 group EOM, the Retransmit Timer (Section 2.1.6.3) will be (re)initialized to its initial value and
979 (re)started so that the frames may again be transmitted if no REPORT is received.

980
981 Note that if the REPORT does not contain NAK'ed Frames and does not acknowledge all
982 outstanding frames, the Retransmit Timer is neither (re)initialized nor stopped. (For example, if
983 a terminal that has transmitted two frame groups receives a REPORT acknowledging the first of
984 the two groups, it does not stop the Retransmit Timer, since the second of the two groups has not
985 yet been acknowledged.)

986
987 Note also that if the window was full and the REPORT acknowledges frames that had not
988 previously been acknowledged, the window is now no longer full, and frames that previously
989 could not be transmitted may now be sent. (See Section 2.1.6.1.)
990

Editor's Note: If multiple REPORT messages are received before the transmitter can act on
them, the action taken by the transmitter can be based on combining the information contained
in these REPORT messages.

991
992
993 **2.1.6.3 Retransmit Timeout**
994

995 In addition to the retransmission of NAK'ed frames described in Section 2.1.6.2,
996 unacknowledged frames are retransmitted on the expiration of the Retransmit Timer.
997

998 The Retransmit Timer is (re)started at initial transmission and at each retransmission. Upon
999 expiration of the Retransmit Timer, previously transmitted frames that have not yet been
1000 acknowledged shall be formatted as a frame group (see Section 2.1.3) and shall be retransmitted.
1001 (An implementer may choose to transmit only a subset of the outstanding frames.) If one or
1002 more previous frame groups were transmitted preceded by an ESCAPE and no REPORTs have
1003 since been received for frames in those frame groups, the retransmission shall be preceded by an
1004 ESCAPE.

1005
1006 An SOM is transmitted first. The SOM is followed by one or more unacknowledged frames.
1007 Within the retransmission, frames shall be ordered in frame count sequence (e.g., frame 10
1008 before frame 11, frame 255 before the next frame 1). An EOM is then transmitted. Immediately

1009 upon transmission of the frame group EOM, the Retransmit Timer shall be (re)initialized to its
1010 initial value and shall be (re)started so that these frames may again be retransmitted if no
1011 REPORTs are received. The value to use when (re)initializing the Retransmit Timer is discussed
1012 in Section 2.4.

1013

1014 REPORT processing shall be performed before Retransmit Timeout processing if both are
1015 pending. If the REPORT processing results in the Retransmit Timer being "stopped" or
1016 (re)started, the Retransmit Timeout processing is not performed.

1017

Editor's Note: It is expected that an implementer will include logic to determine that transmissions are not getting through in spite of repeated retransmissions. This logic is left to the implementer's discretion. It is suggested that the action taken be Connection Terminate, though this is not required.

1018

1019

1020 2.1.7 Message Reception

1021

1022 The Transport Layer Message Reception function is shown in Figure 2.1-5.

1023

1024 When the SOM is received, the receiver shall parse a 20-octet frame from the incoming data
1025 stream. The receiver may perform an FEC decode and shall use the CRC to verify that the frame
1026 was received correctly or that transmission errors were corrected during FEC decoding.

1027

1028 • If the CRC passes and the Frame Count is not zero (i.e., the message is not a Transport Layer
1029 control message) and is within the expected receive window, the frame shall be marked as
1030 correctly received. Frames that are outside the expected receive window shall be discarded
1031 without any additional processing. The receive window extends from the frame following
1032 the current ACK'ed Frame Count, i.e., the frame following the last receive frame that has
1033 been acknowledged, through 127 frames ahead of the ACK'ed Frame Count.

1034

1035 • If the CRC passes and the Frame Count is zero (i.e., the message is a Transport Layer control
1036 message), the terminal shall determine if a REPORT has been received. Each message type
1037 is recognized by its MID. (See Section 2.1.5 for the formats of these messages.) If a
1038 REPORT has been received, processing continues as defined in Section 2.1.5.1.3.

1039

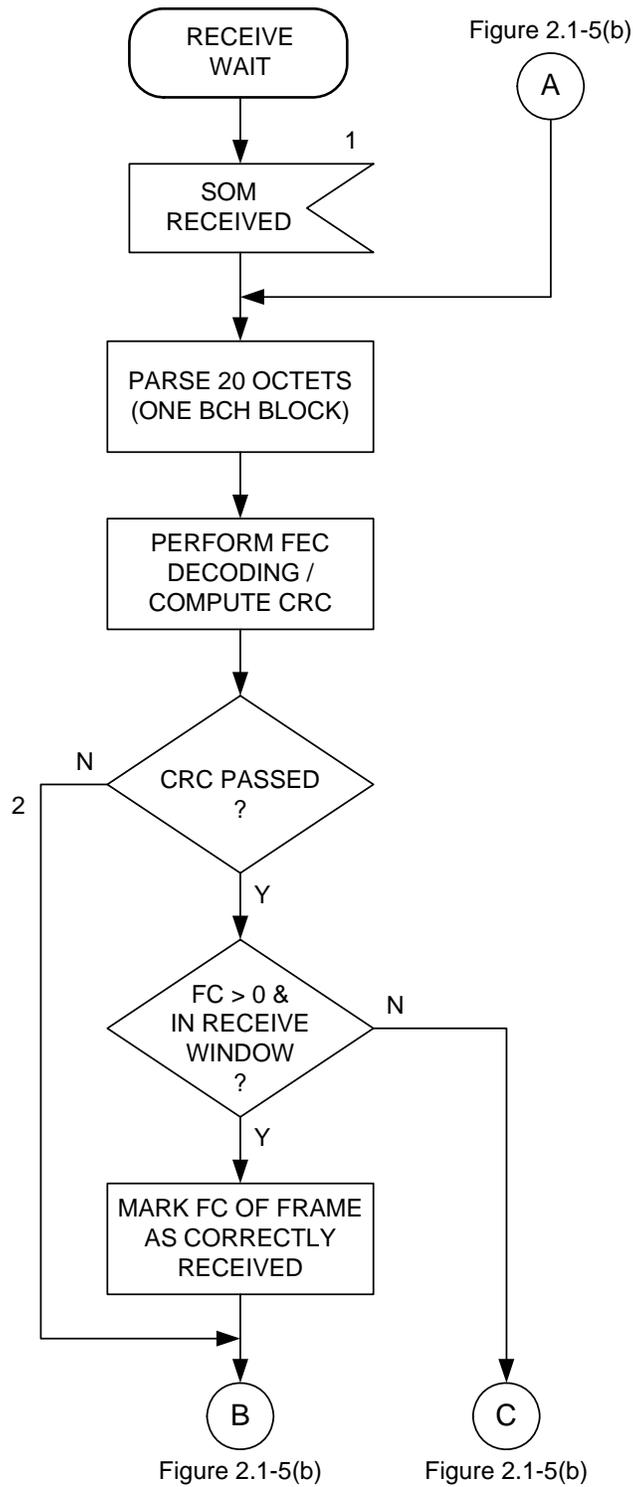
1040 If a REPORT has not been received, also if the CRC does not pass, the following received octets
1041 are checked for an EOM. If an EOM or another SOM does not follow, the receiver shall parse
1042 the next 20-octet frame and repeat the above processing.

1043

Editor's Note: Note that the implementer may opt to consider a frame as being received incorrectly if FEC decoding is not successful. In this case, checking the CRC is not required.

1044

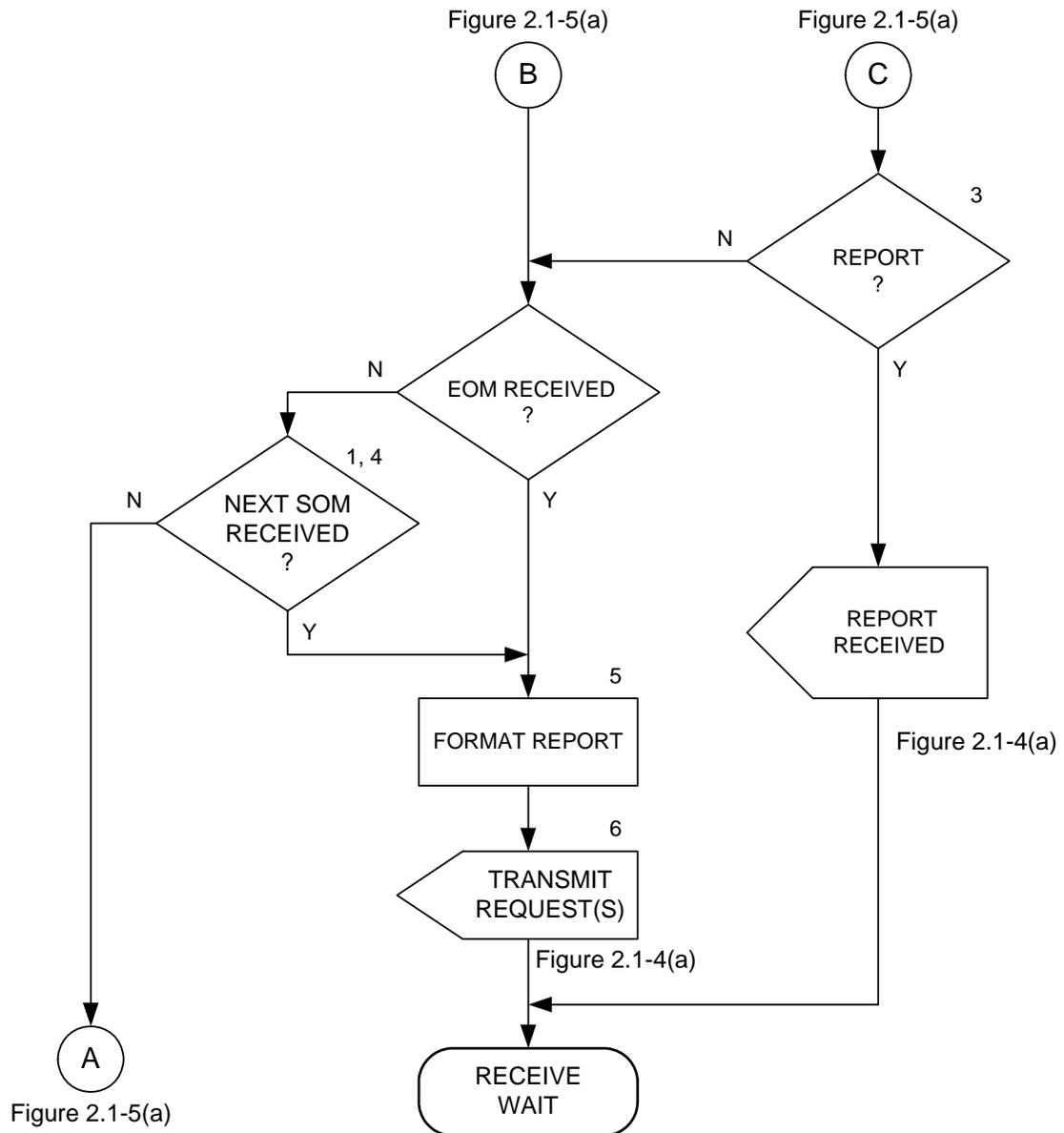
1045



1046
1047
1048
1049

Figure 2.1-5(a) Message Reception

1050



NOTES:

1. This flowchart is entered upon detection of the SOM. Frame groups for which the SOM is not detected may be discarded.
2. If the CRC fails, further attempts to recover useful information may be made at the implementer's discretion.
3. The REPORT message can be recognized by its unique MID.
4. Note that all octets following the SOM must undergo the processing shown in this flowchart.
5. Note that if a frame has previously been ACK'ed, it will not be NAK'ed if it is subsequently received in error.
6. One or more REPORT messages are queued for transmission.

1051
1052
1053
1054
1055

Figure 2.1-5(b) Message Reception (Cont.)

1056
1057 The receiver shall repeat the above process until either the EOM or the next SOM has been
1058 received. Upon receipt of either the EOM or the next SOM, the terminal will format and
1059 transmit a REPORT as specified in Section 2.1.5. Multiple REPORTs may be used, since each
1060 REPORT can identify only seven NAK'ed frames.
1061

Editor's Note: A developer may implement a timer that resends a REPORT if the requested retransmissions are not received. The retransmit logic defined in this Signaling Plan is consistent both with implementations having such a timer and with implementations not having such a timer.

1062
1063 If an EOM is received, the receiver waits for the next SOM. If an SOM is received, the receiver
1064 immediately starts processing the frames that follow the SOM.
1065

Editor's Note: If a receiver is able to recognize and process frames in a frame group even when the SOM is not detected (e.g., by working backward from an EOM that is detected), this is permitted though it is not required.

1066
1067
1068 **2.1.8 Octet Alignment**
1069

1070 The frame group and ESCAPE signaling are shown octet aligned and are expected to be
1071 transmitted octet aligned. However, the signaling may be carried on networks that do not
1072 preserve octet alignment. Therefore, the SCIP receiver shall be capable of recovering and
1073 processing the SCIP signaling shown herein even if it is not octet aligned when it is received.
1074

1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120

2.2 SCIP Call Setup Signaling

This section defines the SCIP call setup signaling. Section 2.2.1 provides an overview of this signaling. Section 2.2.2 describes the Capabilities Exchange which is always required. Sections 2.2.3, 2.2.4 and 2.2.5 describe the Parameters/Certificate Exchange, the F(R) Exchange, and the Cryptosync Exchange which are used to establish a standard secure operational mode. The F(R) Exchange is not used for PPK processing. Section 2.2.6 provides a compilation of standard SCIP Operational Mode specific field definitions and values.

2.2.1 Introduction and Overview

This section defines the SCIP point-to-point call setup signaling. It is assumed that an end-to-end data channel has already been established, using the underlying channel protocols, by means outside the scope of this Signaling Plan. The signaling necessary to establish a SCIP point-to-point Operational Mode is then executed over this data channel. The two SCIP terminals proceed, independently and in parallel, to execute the signaling defined in this section (except in a few specific places which are indicated through the use of Initiator/Responder terminology).

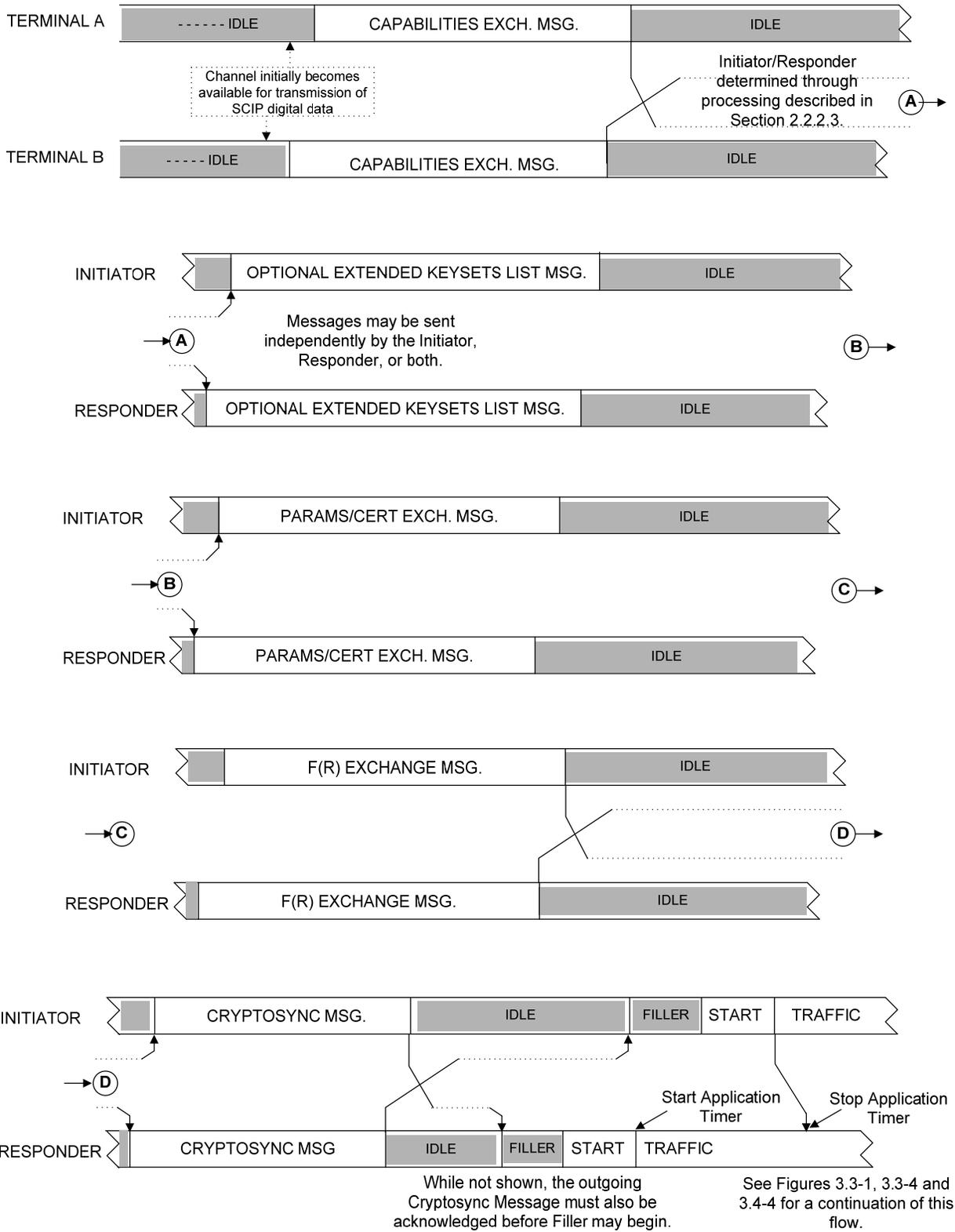
2.2.1.1 Secure Call Setup Signaling Time Line

The following subsections provide examples of the overall flow for setting up a SCIP point-to-point secure call. The secure call setup time lines are shown with no retransmissions. The examples begin with two terminals both transmitting Capabilities Messages. Once they are received, the Initiator and Responder roles are determined from the information contained in the Capabilities Messages. During IDLE periods, there is no transmission of bits by the SCIP application, though there may actually be bits on individual links related to handshaking performed by the underlying data channel protocols.

If a failure occurs at any point during SCIP call setup or if the user selects Nonsecure during call setup, the Native Clear Voice/Connection Idle signaling described in Section 2.3.2.3 will be executed. If the user goes "on-hook" without waiting for call setup to complete, the Connection Terminate signaling described in Section 2.3.2.2 will be executed.

2.2.1.1.1 FIREFLY Example

A normal SCIP call setup using FIREFLY key is shown in Figure 2.2-1(a). One or more application messages are exchanged. The Capabilities Messages are always exchanged. If a standard secure Operational Mode is chosen, the Capabilities Exchange is followed by the exchange of optional Extended Keypsets List Messages, Parameters/Certificate Messages, F(R) Messages, and Cryptosync Messages. These exchanges are described in Sections 2.2.2 through 2.2.5. Under exception conditions, Notification Messages (described in Section 2.3.2) may also be exchanged.



1121
1122
1123

Figure 2.2-1(a) FIREFLY Secure Call Setup Signaling Time Line

1124

Editor's Note: Note that the dotted lines indicate a functional relationship where one message must be received before the second message can be formatted and transmitted.

1125

1126 Capabilities and optional Extended Keysets List Message Exchanges are specified in Section
1127 2.2.2. In the example shown, when a clear data channel has been set up between the two
1128 terminals (the Connection Idle state) using the underlying native mechanisms, and is available to
1129 carry SCIP messages, both terminals simultaneously initiate SCIP call setup. It is also possible
1130 for one terminal to initiate the call setup, with the other terminal responding with a Capabilities
1131 Message only when it receives the Capabilities Message from the Initiator. If a terminal receives
1132 no recognizable response after sending a Capabilities Message, it will time out and reenter the
1133 Connection Idle state as described in Section 2.2.1.2.

1134

1135 Upon receipt of the Capabilities and optional Extended Keysets List Messages, the terminals will
1136 choose a common Operational Mode (generic application) and Keyset Type (a combination of
1137 key management signaling and traffic encryption). If a clear Operational Mode is chosen, the
1138 terminals will begin clear application signaling. In the example shown, a standard secure
1139 Operational Mode and Keyset are chosen and call setup signaling continues with the exchange of
1140 Parameters/Certificate and F(R) Messages. Since not all Operational Mode parameters are
1141 negotiated in the Capabilities Message, it may be necessary to exchange multiple
1142 Parameters/Certificate Messages for multiple Operational Modes before a Mode that both
1143 terminals can support is negotiated. Parameters/Certificate Exchange is specified in Section
1144 2.2.3, and F(R) Exchange is specified in Section 2.2.4.

1145

1146 When the terminals have received the Parameters/Certificate and F(R) Messages, they will use
1147 the Certificate and F(R) for the chosen Keyset to generate a common traffic key. The terminals
1148 will then encode and encrypt information necessary to verify the cryptography and the preceding
1149 clear exchanges, and will enclose these encrypted packets in Cryptosync Messages. The
1150 Cryptosync Message also carries the Application IV for the chosen Operational Mode. The
1151 Cryptosync Messages are now exchanged. If the two terminals have different CKL versions for
1152 the chosen Keyset, the terminal containing the newer CKL will wait until it receives a
1153 Cryptosync Message then transmit its CKL in one or more Notification Messages prior to
1154 transmitting its Cryptosync Message. Once the CKL Transfer is complete, the Cryptosync
1155 Messages have been successfully exchanged, and the "packets" have been verified, the terminals
1156 will initiate the secure application. The CKL Transfer is described in Section 2.3.2.4, the
1157 Cryptosync Exchange is described in Section 2.2.5, the startup of application signaling for
1158 standard applications is described in Section 3.2, and the signaling for each of the standard
1159 secure applications is described in subsequent subsections of Section 3.

1160

1161

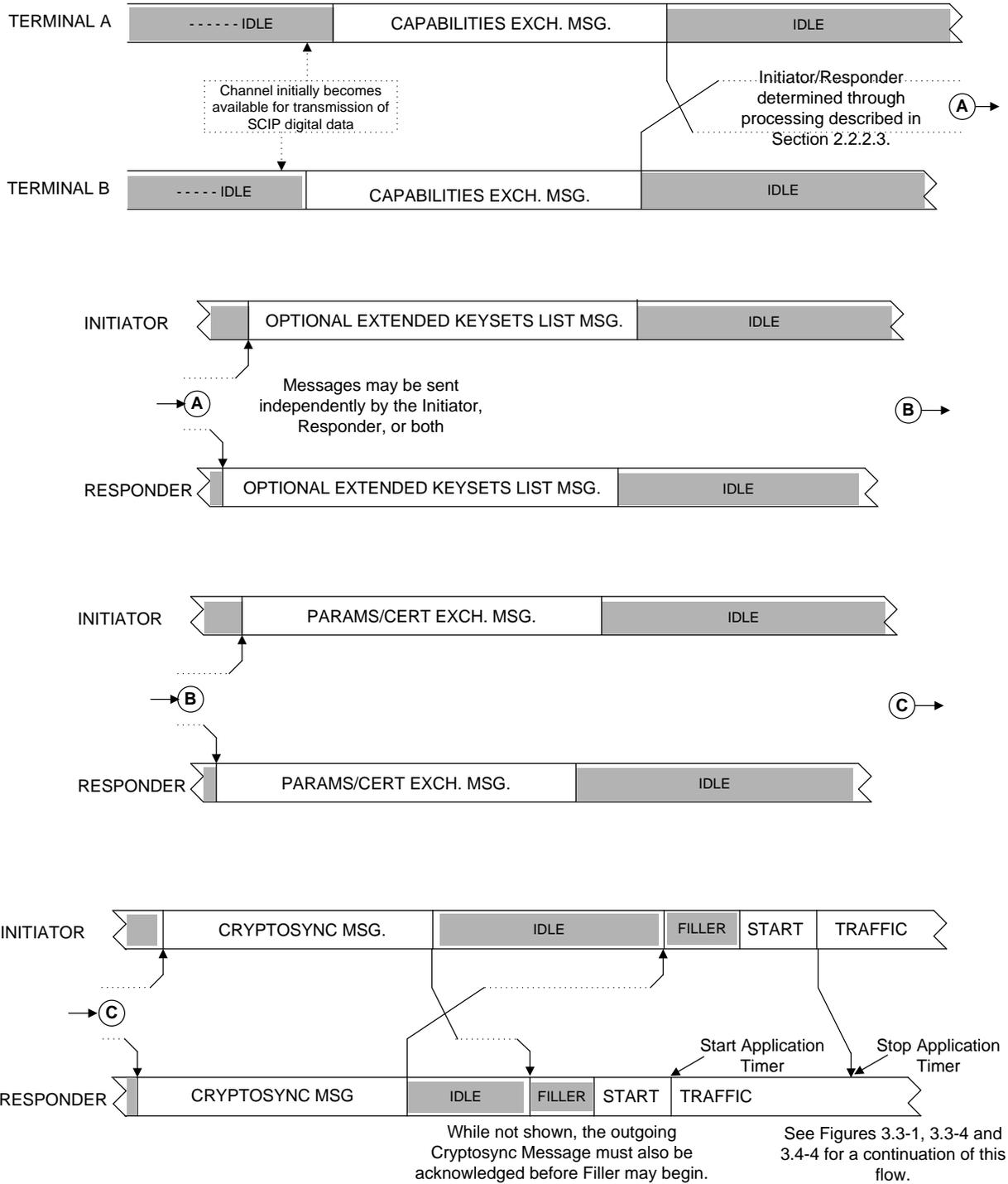
1162 **2.2.1.1.2 PPK Example**

1163

1164 A normal SCIP call setup using PPKs is shown in Figure 2.2-1(b). One or more application
1165 messages are exchanged. The Capabilities Messages are always exchanged. If a standard secure
1166 Operational Mode is chosen, the Capabilities Exchange is followed by the exchange of optional
1167 Extended Keysets List Messages, Parameters/Certificate Messages, and Cryptosync Messages.

1168
1169
1170
1171

These exchanges are described in Sections 2.2.2, 2.2.3, and 2.2.5. Under exception conditions, Notification Messages (described in Section 2.3.2) may also be exchanged.



1172
1173
1174

Figure 2.2-1(b) PPK Secure Call Setup Signaling Time Line

1175
1176

Editor's Note: Note that the dotted lines indicate a functional relationship where one message must be received before the second message can be formatted and transmitted.

1177
1178
1179
1180
1181
1182
1183
1184
1185

Capabilities and optional Extended Keysets List Message Exchanges are specified in Section 2.2.2. In the example shown, when a clear data channel has been set up between the two terminals (the Connection Idle state) using the underlying native mechanisms, and is available to carry SCIP messages, both terminals simultaneously initiate SCIP call setup. It is also possible for one terminal to initiate the call setup, with the other terminal responding with a Capabilities Message only when it receives the Capabilities Message from the Initiator. If a terminal receives no recognizable response after sending a Capabilities Message, it will time out and reenter the Connection Idle state as described in Section 2.2.1.2.

1186
1187
1188
1189
1190
1191
1192
1193
1194
1195

Upon receipt of the Capabilities and optional Extended Keysets List Messages, the terminals will choose a common Operational Mode (generic application) and Keyset Type (in this case, a PPK is chosen). If a clear Operational Mode is chosen, the terminals will begin clear application signaling. In the example shown, a standard secure Operational Mode and PPK Keyset are chosen and call setup signaling continues with the exchange of Parameters/Certificate Messages. Since not all Operational Mode parameters are negotiated in the Capabilities Message, it may be necessary to exchange multiple Parameters/Certificate Messages for multiple Operational Modes before a Mode that both terminals can support is negotiated. Parameters/Certificate Exchange is specified in Section 2.2.3.

1196
1197
1198
1199
1200
1201
1202
1203
1204
1205

When the terminals have received the Parameters/Certificate Messages, they will encode and encrypt information necessary to verify the cryptography and the preceding clear exchanges, and will enclose these encrypted packets in Cryptosync Messages. The Cryptosync Message also carries the Application IV for the chosen Operational Mode. The Cryptosync Messages are now exchanged. Once the Cryptosync Messages have been successfully exchanged, and the "packets" have been verified, the terminals will initiate the secure application. The Cryptosync Exchange is described in Section 2.2.5, the startup of application signaling for standard applications is described in Section 3.2, and the signaling for each of the standard secure applications is described in subsequent subsections of Section 3.

1206
1207

2.2.1.2 First Message Time-Out

1208
1209
1210
1211
1212
1213
1214
1215

A First Message Timer is started when the Capabilities Message is transmitted (see Section 2.2.2). This timer enables the terminal to time out should the far end not respond with a message that is recognizable as a SCIP message. Should this timer expire, the terminal shall execute the Failed Call logic defined in Section 2.3.2.3.1, with an Information Code of *SCIP response not received*, to return to Connection Idle state.

1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261

2.2.1.3 Unrecognized Messages

In the case where a terminal receives an unrecognized message, the terminal may silently discard it or invoke either the Failed Call or Connection Terminate options as defined in Section 2.3. If the decision is to silently discard the message, the terminal shall remain in the same signaling state as prior to receiving it.

2.2.1.4 Message Limitations

To ensure interoperability, terminals implementing SCIP-210, Rev. 3.2 or later shall send SCIP Call Setup and Notification Messages (excluding CKL Transfer) with a message limitation of 1024 octets, except when the ability to send longer messages has been defined and negotiated. Note that this message limitation is on the total message length; no additional limitations are imposed on the length of variable length fields within these messages. Note that if a terminal must include a very long Keysets List in the Capabilities Message that causes the Capabilities Message to surpass this message length limitation, the optional Extended Keysets List Messages must be used (see Section 2.2.2.4). All terminals implementing SCIP-210, Rev 3.2 or later shall be capable of receiving SCIP Call Setup and Notification Messages (excluding CKL Transfer) with a total message length of at least 1024 octets.

If the terminals offer multiple Operational Modes in the Capabilities Messages and the Parameters/Certificate Messages resulting from the chosen Operational Mode do not have compatible Parameters, the terminals will continue to negotiate Operational Modes (see Section 2.2.2.3.2) and transmit Parameters/Certificate Messages until either compatible Parameters are identified or Failed Call processing is executed. A terminal shall be capable of receiving at least three Parameters/Certificate Messages resulting from Operational Mode negotiations. Terminals may send more than three Parameters/Certificate Messages; however, interoperability is not guaranteed if more than three Parameters/Certificate Messages are sent.

2.2.2 Capabilities Message

The first step in SCIP Call Setup is the exchange of Capabilities Messages. This exchange permits the terminals to negotiate a clear or secure Operational Mode which both support. For secure Operational Modes it also permits the terminals to choose compatible Keysets for which Credentials will be subsequently exchanged.

2.2.2.1 Capabilities Message Definition

The format of the Capabilities Message is shown in Table 2.2-1. The Version 0 Capabilities Message contains the fields shown in Table 2.2-1(a) and Table 2.2-1(b), i.e., through the optional Keysets List field. The Version 1 or higher additions, shown in Table 2.2-1(c), begin with the Security Data Length field.

1262
1263
1264

Table 2.2-1(a) Capabilities Message Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
0-msb	0	0	0	0	0	0	0	1
Source ID				MID				
0	0	0	0	0	0	1	0-lsb	2
X-msb	X	X	X	X	X	X	X	3
Message Length								
X	X	X	X	X	X	X	X-lsb	4
X-msb	X	X	X	X	X	X	X-lsb	5
Message Version								
X	X	X	X	X	X	X	X	6
X	X	X	X	X	X	X	X	
Initiator Negotiation								
X	X	X	X	X	X	X	X	
I/R Bit	Random Number							
Signaling Plan Version								
0	0	0	0	0	0	0	1	7
X	X	X	X	X	X	X	X	8
ID Information								
Source ID								
X	X	X	X	X	X	X	X	9
ID Value								
X	X	X	X	X	X	X	X	15
Modes Length								
X-msb	X	X	X	X	X	X	X	16
X	X	X	X	X	X	X	X-lsb	17
Operational Modes List								
X-msb	X	X	X	X	X	X	X	18
Source ID								
X	X	X	X	X	X	X	X-lsb	19
First Operational Mode Entry								
ID Value								
X-msb	X	X	X	X	X	X	X	17+2L-1
Source ID								
X	X	X	X	X	X	X	X-lsb	17+2L
L'th Operational Mode Entry								

1265

1266
1267
1268

Table 2.2-1(b) Capabilities Message Format (Cont.)

8 (msb)	7	6	5	4	3	2	1 (lsb)	⇐ Bits Octets ↓
X-msb	X	X	Keysets Length		X	X	X	18+2L
X	X	X	X	X	X	X	X-lsb	19+2L
X	X	X	Keysets List (Optional)		X	X	X	20+2L
			•••					
X	X	X	X	X	X	X	X	19+2L+M

L = Number of Operational Mode Entries. M = Length of Keysets List field.

1269
1270
1271
1272
1273

Table 2.2-1(c) Capabilities Message Format – Version 1 or Higher (Cont.)

8 (msb)	7	6	5	4	3	2	1 (lsb)	⇐ Bits Octets ↓
X-msb	X	X	Security Data Length		X	X	X	20+2L+M
X	X	X	X	X	X	X	X-lsb	21+2L+M
X-msb	X	X	Security Data		X	X	X	22+2L+M
			•••					
X	X	X	X	X	X	X	X-lsb	21+2L+M+N
X-msb	X	X	Terminal Priority COI		X	X	X-lsb	22+2L+M+N
X-msb	X	X	Terminal Priority		X	X	X-lsb	23+2L+M+N
X-msb I/R Bit	X	X	Alternate Initiator Negotiation		X	X	X	24+2L+M+N
X	X	X	X	X	X	X	X-lsb	25+2L+M+N
			Random Number					

L = Number of Operational Mode Entries. M = Length of Keysets List field. N = Length of Security Data.

1275
1276
1277
1278
1279
1280
1281
1282
1283

- For the Capabilities Message the value of the MID is 0x0002.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- The Message Version field shall be an unsigned binary integer with the high order bit being bit 8 and the low order bit being bit 1.

Editor's Note: A later version message than what is implemented in a terminal can be processed by discarding the newer information; i.e., changes must be made so that the message is backwardly compatible. This applies to all messages.

1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316

A terminal in the Interoperable Terminal Priority COI, defined in Table 2.2-3(c), shall set the Initiator Negotiation field as follows. If the terminal has the Standard Terminal Priority (see Table 2.2-3(d)), or if it transmits a Version 0 Capabilities Message, the I/R Bit shall contain a 1 for an initiating terminal and a 0 for a responding terminal. The lower 7 bits shall contain a Random Number to resolve the case where both terminals initially view themselves as Initiators or Responders. If the terminal is transmitting a Version 1 or higher Capabilities Message and has a priority other than the Standard Terminal Priority, the Initiator Negotiation field shall be set to 0x00.

- The value of Signaling Plan Version is 0x01 for this version of the Signaling Plan.
- The ID Information field may be used to identify the terminal's "security element". Content, processing, and format may vary from implementation to implementation. The high order 5 bits of the first octet identify a Source for the ID Information definition. Currently identified Source ID values are defined in Section 2.5.1. The terminal may set all bits of this field to 0 if no ID Information is to be transmitted. **[Deviation Notice:** *The value 0x28 in octet 8 (i.e., Source ID = 0x05 in bits 4 - 8 and bits 1 - 3 set to zero) indicates the terminal requires special formatting for CKL Transfer (see Section 2.3.2.1).]*
- The Modes Length shall contain the actual length of the Operational Modes List (plus the length of the Modes Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 16 and the low order bit being bit 1 of octet 17.
- The Operational Modes List shall contain one or more Operational Mode ID Entries. The Operational Mode ID Entries shall occur in order of preference; the ID of the preferred Operational Mode is placed in octets 18 and 19, the ID (if present) of the second choice Operational Mode is placed in octets 20 and 21, etc. The first entry on the Initiator's List which is also on the Responder's List is the Operational Mode chosen. Each Operational Mode ID in the Operational Modes List is 2 octets in length. The high order 5 bits of the first octet identify a Source for the Operational Mode definition. Currently identified Source ID values are defined in Section 2.5.1. The Source ID value plus the next 11 bits constitute an Operational Mode ID. The high order bit of the Operational Mode ID is placed in bit 8 of the first octet of the

1317 Operational Mode List entry and the low order bit of the Operational Mode ID is
 1318 placed in bit 1 of the second octet of the Operational Mode List entry. Currently
 1319 defined standard Operational Mode IDs are identified in Table 2.2-2. In order to
 1320 prevent the Secure Electronic Rekey Operational Mode (0x000E) from being
 1321 negotiated on calls between two standard SCIP devices, this mode shall only be
 1322 offered by a SCIP Line Interface Terminal (SCIP-LIT); furthermore, this is the only
 1323 Operational Mode that will be offered by a SCIP-LIT.
 1324

Table 2.2-2 SCIP Standard Operational Modes

Operational Mode ID	Operational Mode Definition
0x0001	Secure Voice
0x0002	Secure Data
0x0003	Enhanced Secure Data
0x0004	Clear MELP Voice
0x0008	Native Clear Voice
0x000E	Secure Electronic Rekey (offered only by the SCIP-LIT)

- 1328
- 1329 • The Keysets Length shall contain the actual length of the Keysets List (plus the
 1330 length of the Keysets Length field itself) in octets. The value of the field shall be an
 1331 unsigned binary integer with the high order bit being bit 8 of the first octet of the field
 1332 and the low order bit being bit 1 of the second octet of the field.
 - 1333 • The Keysets List contains Keysets List Entries of the form given in Table 2.2-3(a).
 1334 Each Keyset shall have a Keysets List Entry on the Keysets List. If only clear modes
 1335 are offered, no Keysets need be listed on the Keysets List, i.e., the optional Keysets
 1336 List need not be present. Keysets List Entries shall be in prioritized order per the
 1337 rules defined by the controlling Terminal Priority COI. SCIP-230 or SCIP-232,
 1338 Section 2.1.1.1.2; or SCIP-231, Section 2.1.1.2 defines the rules for the Standard
 1339 Terminal Priority. Each entry shall consist of a Keyset Type, followed by a Keyset
 1340 Parameters Length and Keyset Parameters (if parameters are defined) for a single
 1341 Keyset. The length of the Keysets List is the sum of the lengths of the individual
 1342 Keysets List Entries.

1343
1344
1345

Table 2.2-3(a) Keysets List Entry - General Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Keyset Type								
X-msb	X	X	X	X	X	X	X	1
Source ID								
X	X	X	X	X	X	X	X-lsb	2
Keyset Parameters Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Keyset Parameters (Optional)								
X	X	X	X	X	X	X	X	5
...								
X	X	X	X	X	X	X	X	4+M

1346

M = Length of Keyset Parameters field.

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

- The first field of a Keysets List Entry shall contain a Keyset Type. The high order 5 bits of the first octet constitute a Source for the Keyset Type definition. Current Source IDs are defined in Section 2.5.1. The next 11 bits identify a unique Keyset Type within that Source. Currently defined values for Keyset Type are listed in Table 2.2-3(b). The high order bit of the Keyset Type is placed in bit 8 of the first octet, and the low order bit of the Keyset Type is placed in bit 1 of the second octet.
- The second field of a Keysets List Entry shall contain a Keyset Parameters Length. This shall contain the actual length, in octets, of the Keyset Parameters (plus the length of the Keyset Parameters Length itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- The third field of a Keysets List Entry shall contain the Keyset Parameters. The Keyset Parameters field is variable length, and its contents are unique to each Keyset Type for which it is defined. For each standard Keyset Type, the format of the corresponding Keyset Parameters is defined in Section 2.2.6.1.1. This field is optional and is not present unless Keyset Parameters are defined for a given Keyset Type.

1366
1367
1368

Table 2.2-3(b) SCIP Standard Keyset Types

Keyset Type Code	Keyset Type Definition
0x0001	Key Management/Signaling: Type 1 Basic FIREFLY Key Exchange without Call Setup Encryption (CSE). Traffic Encryption: Type 1 traffic encryption algorithm specified in SCIP-230.
0x0002 (Note 1)	Key Management/Signaling: Type 1 Enhanced FIREFLY Key Exchange without Call Setup Encryption. Traffic Encryption: Type 1 traffic encryption algorithm specified in SCIP-230.
0x0004	Key Management/Signaling: Type 1 Basic FIREFLY Key Exchange with Call Setup Encryption. Traffic Encryption: Type 1 traffic encryption algorithm specified in SCIP-230.
0x0007 (Note 1)	Key Management/Signaling: Type 1 Enhanced FIREFLY Key Exchange with Call Setup Encryption. Traffic Encryption: Type 1 traffic encryption algorithm specified in SCIP-230.
0x0008	Key Management/Signaling: Type 1 U.S. Generic Pre-Placed Key (PPK) without Call Setup Encryption. Traffic Encryption: Type 1 traffic encryption algorithm specified in SCIP-230.
0x0009	Key Management/Signaling/Traffic Encryption: Non-Type 1 ECMQV/AES without Call Setup Encryption – Phase 1 as specified in SCIP-231.
0x000A	Key Management/Signaling/Traffic Encryption: Non-Type 1 ECMQV/AES with Call Setup Encryption – Phase 1 as specified in SCIP-231.
0x000B	Key Management/Signaling/Traffic Encryption: NATO ECMQV/AES without Call Setup Encryption as specified in SCIP-232.
0x000C	Key Management/Signaling/Traffic Encryption: NATO ECMQV/AES with Call Setup Encryption as specified in SCIP-232.
0x000D	Key Management/Signaling/Traffic Encryption: NATO PPK/AES without Call Setup Encryption as specified in SCIP-232.
0x0010	Reserved.
0x07FF	Extended Keysets List Support.

1369
1370
1371

Note 1: Enhanced FIREFLY is a U.S. defined interoperable cryptographic mode. Any future use of Enhanced FIREFLY and release of supporting documentation to U.S. partners will be through bilateral agreements.

1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392

- The Security Data Length shall contain the actual length of the Security Data (plus the length of the Security Data Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.
- The Security Data field shall be populated per SCIP-230, Sections 2.1.1.3.1.5 and 3.4.2, SCIP-231, Sections 2.1.2.1.2 and 3.2.2, or SCIP-232, Sections 2.1.1.3.1.5 and 3.2.2.1. The Security Data's most significant bit (as defined in SCIP-230, Section 3.1.2.1, SCIP-231, Section 3.1.2, or SCIP-232, Section 3.2.2.1) shall be placed in bit 8 of the first octet, and its least significant bit shall be placed in bit 1 of the N'th octet.
- The Terminal Priority COI is a unique value assigned to each Community of Interest that independently defines keyset selection rules. These rules are intended to provide a mechanism for SCIP devices to determine which terminal's keyset ordering has priority. It identifies the community that controls the Terminal Priority and provides a mechanism for each community to control its terminals' priorities without international agreements. Currently defined Terminal Priority COI values are listed in Table 2.2-3(c).

Table 2.2-3(c) Terminal Priority COI Values

Terminal Priority COI	Value	Keyset Selection Rules
Interoperable	0x80	Specified in SCIP-230 or SCIP-232, Section 2.1.1.1.1; or SCIP-231, Section 2.1.1.1

1393
1394
1395
1396
1397
1398
1399
1400
1401

- The Terminal Priority is a value, assigned by the Terminal Priority COI, that identifies the relative keyset ordering priority of a class of terminals within the community. Currently defined Terminal Priority values in the Interoperable Terminal Priority COI are listed in Table 2.2-3(d).

Table 2.2-3(d) Terminal Priority Values

Terminal Priority	Value	Keyset Prioritization Rules
Standard	0x80	Specified in SCIP-230 or SCIP-232, Section 2.1.1.1.2; or SCIP-231, Section 2.1.1.2
Non-Type 1/Type 1	0x40	To be defined elsewhere

1402
1403

Editor's Note: Except for special cases, it is anticipated that terminals will use the Interoperable Terminal Priority COI and Standard Terminal Priority values. Keyset selection rules for terminals not in the Interoperable Terminal Priority COI are outside the scope of this document.

1404

- 1405 • The I/R bit in the Alternate Initiator Negotiation field shall contain a 1 for an initiating
1406 terminal and a 0 for a responding terminal. The lower 15 bits shall contain a Random
1407 Number to resolve the case where both terminals initially view themselves as Initiators or
1408 Responders.

1409
1410 Table 2.2-3(e) provides an example of a Capabilities Message that is appropriate for
1411 transmission by an Enhanced FIREFLY (FF) capable terminal. In the example shown, the
1412 terminal is loaded with US, CCEB, NATO, NATO Nations, and Coalition key material. The US
1413 and CCEB Keysets include a Current and a Next Universal Edition; the US Keysets are
1414 Enhanced FF capable. The US and NATO Keysets have optional Call Setup Encryption
1415 capability; therefore, they are offered with and without CSE. The US Keysets have two CSE
1416 keys associated with each Universal. The terminal offers one US and one NATO Pre-Placed
1417 Key. Finally, the terminal is also NATO ECMQV/AES and ECMQV/AES capable with optional
1418 CSE keys.

1419

1420
1421
1422

Table 2.2-3(e) Example of Capabilities Message Contents – Enhanced FF Capable

Capabilities Message Field	Value	Length (octets)	Notes
MID	0x0002	2	
Message Length	0xXXXX	2	
Message Version	0x01	1	
Initiator Negotiation	0xXX	1	
Signaling Plan Version	0x01	1	
ID Information	0xXX...XX	8	
Modes Length	0x0006	2	
Secure Voice	0x0001	2	Note 1
Secure Data	0x0002	2	Note 1
Keysets Length	0x00ED	2	
Keypad Type (Enhanced FF with CSE)	0x0007	2	Note 2
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CSE SPI - 1	0xXX...XX	4	Note 2
CKL Version (Next)	0x01	1	
Keypad Type (Enhanced FF with CSE)	0x0007	2	Note 2
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CSE SPI - 2	0xXX...XX	4	Note 2
CKL Version (Next)	0x01	1	
Keypad Type (Enhanced FF with CSE)	0x0007	2	Note 2
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CSE SPI - 1	0xXX...XX	4	Note 2
CKL Version (Current)	0x01	1	
Keypad Type (Enhanced FF with CSE)	0x0007	2	Note 2
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	

1423

1424
1425
1426

Table 2.2-3(e) Capabilities Message Contents – Enhanced FF Capable (Cont.)

Capabilities Message Field	Value	Length (octets)	Notes
Universal Edition (Current)	0x01	1	
CSE SPI - 2	0xXX...XX	4	Note 2
CKL Version (Current)	0x01	1	
Keypad Type (Enhanced FF w/o CSE)	0x0002	2	Note 2
Keypad Parameters Length	0x0006	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CKL Version (Next)	0x01	1	
Keypad Type (Enhanced FF w/o CSE)	0x0002	2	Note 2
Keypad Parameters Length	0x0006	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CKL Version (Current)	0x01	1	
Keypad Type (Basic FF with CSE)	0x0004	2	Notes 2, 4
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CSE SPI - 1	0xXX...XX	4	Note 2
CKL Version (Next)	0x01	1	
Keypad Type (Basic FF with CSE)	0x0004	2	Notes 2, 4
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CSE SPI - 2	0xXX...XX	4	Note 2
CKL Version (Next)	0x01	1	
Keypad Type (Basic FF with CSE)	0x0004	2	Notes 2, 4
Keypad Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CSE SPI - 1	0xXX...XX	4	Note 2
CKL Version (Current)	0x01	1	

1427

1428
1429
1430

Table 2.2-3(e) Capabilities Message Contents – Enhanced FF Capable (Cont.)

Capabilities Message Field	Value	Length (octets)	Notes
Keypset Type (Basic FF with CSE)	0x0004	2	Notes 2, 4
Keypset Parameters Length	0x000A	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CSE SPI - 2	0xXX...XX	4	Note 2
CKL Version (Current)	0x01	1	
Keypset Type (Basic FF w/o CSE)	0x0001	2	Note 4
Keypset Parameters Length	0x0006	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CKL Version (Next)	0x01	1	
Keypset Type (Basic FF w/o CSE)	0x0001	2	Note 4
Keypset Parameters Length	0x0006	2	
Universal ID (US)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CKL Version (Current)	0x01	1	
Keypset Type (Basic FF w/o CSE)	0x0001	2	Note 4
Keypset Parameters Length	0x0006	2	
Universal ID (CCEB)	(Note 3)	2	
Universal Edition (Next)	0x02	1	
CKL Version (Next)	0x01	1	
Keypset Type (Basic FF w/o CSE)	0x0001	2	Note 4
Keypset Parameters Length	0x0006	2	
Universal ID (CCEB)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CKL Version (Current)	0x01	1	
Keypset Type (NATO ECMQV/AES w/CSE)	0x000C	2	Note 2
Keypset Parameters Length	0x000A	2	
Universal ID (NATO)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CSE SPI	0xXX...XX	4	
CKL Version (Current)	0x01	1	

1431

1432
1433
1434

Table 2.2-3(e) Capabilities Message Contents – Enhanced FF Capable (Cont.)

Capabilities Message Field	Value	Length (octets)	Notes
Keyset Type (NATO ECMQV/AES w/o CSE)	0x000B	2	Note 2
Keyset Parameters Length	0x0006	2	
Universal ID (NATO)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CKL Version (Current)	0x01	1	
Keyset Type (Basic FF w/o CSE)	0x0001	2	Note 4
Keyset Parameters Length	0x0006	2	
Universal ID (NATO Nations)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CKL Version (Current)	0x01	1	
Keyset Type (Basic FF w/o CSE)	0x0001	2	Note 4
Keyset Parameters Length	0x0006	2	
Universal ID (Coalition)	(Note 3)	2	
Universal Edition (Current)	0x01	1	
CKL Version (Current)	0x01	1	
Keyset Type (U.S. Generic PPK w/o CSE)	0x0008	2	
Keyset Parameters Length	0x0006	2	
PPK SPI	0xXX...XX	4	Note 5
Keyset Type (NATO PPK/AES w/o CSE)	0x000D	2	Note 2
Keyset Parameters Length	0x0006	2	
PPK SPI	0xXX...XX	4	Note 5
Keyset Type (ECMQV/AES with CSE – 1)	0x000A	2	Note 2, 6
Keyset Parameters Length	0x0007	2	
Keyset ID	0xXX	1	Note 7
CSE SPI – 1	0xXX...XX	4	Note 2, 8
Keyset Type (ECMQV/AES with CSE – 1)	0x000A	2	Note 2, 6
Keyset Parameters Length	0x0007	2	
Keyset ID	0xXX	1	Note 7
CSE SPI - 2	0xXX...XX	4	Note 2, 8
Keyset Type (ECMQV/AES w/o CSE – 1)	0x0009	2	Note 2, 6
Keyset Parameters Length	0x0003	2	
Keyset ID	0xXX	1	Note 7

1435

1436
1437
1438

Table 2.2-3(e) Capabilities Message Contents – Enhanced FF Capable (Cont.)

Capabilities Message Field	Value	Length (octets)	Notes
Security Data Length	0XXXXX	2	
Security Data	0XX...XX	N	
Terminal Priority COI	0x80	1	
Terminal Priority	0x80	1	
Alternate Initiator Negotiation	0XXXXX	2	

1439
1440
1441
1442
1443
1444
1445
1446
1447
1448

Notes:

1. Mode implies only the generic application that will be used.
2. See SCIP-230 or SCIP-232, Section 2.1.1.1; or SCIP-231, Section 2.1.1, for keyset ordering rules.
3. See SCIP-230 or SCIP-232, Section 2.1.1.2.1, for applicable Universal ID values.
4. Basic FF Keyset Types are offered for backward compatibility.
5. See SCIP-230 or SCIP-232, Section 2.1.1.2.2, for the PPK attributes defined for the SPI.
6. ECMQV/AES – 1 indicates an ECMQV/AES – Phase 1 Keyset Type.
7. See SCIP-231, Section 2.1.1.2.
8. See SCIP-231, Section 2.1.2.1.1.2.

1449
1450
1451

2.2.2.2 Capabilities Message Transmission

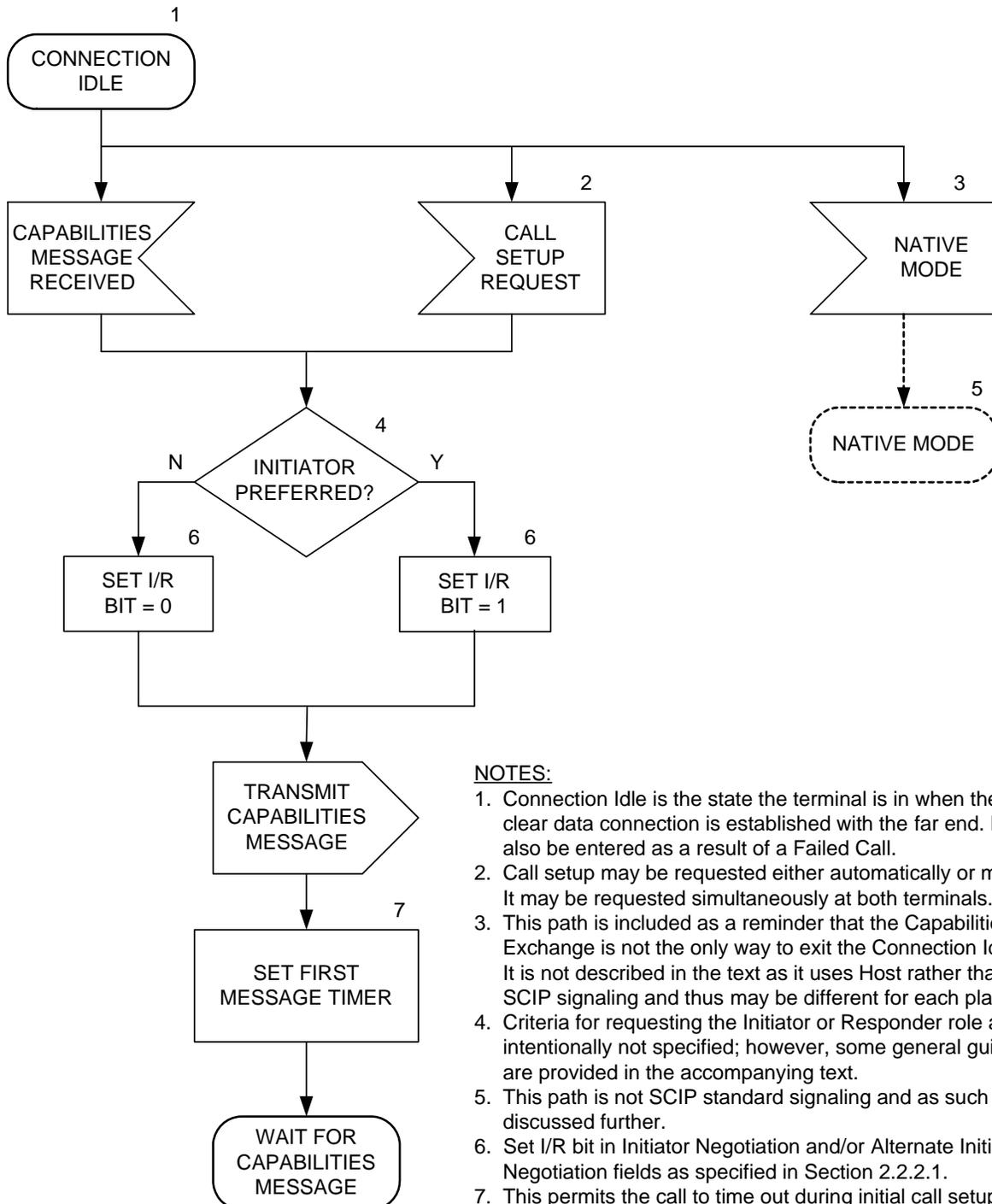
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465

Capabilities Message transmission is shown in Figure 2.2-2. This signaling occurs at the beginning of SCIP point-to-point call establishment. It starts from the Connection Idle state. This signaling will also be executed when the terminal receives a request to transition from a SCIP clear application to a SCIP secure application or vice versa, and to negotiate a new application after a terminal transitions to Connection Idle on an error condition. The Notification exchange will bring both terminals to Connection Idle. Messages sent by a Follower prior to the Notification exchange may arrive after the Leader has entered Connection Idle and even after the Leader has entered the Wait for Capabilities Message state. Since only incoming Notification Messages and Capabilities Messages are valid during a transition, other messages received during the Connection Idle and Wait for Capabilities Message states must be discarded.

Editor's Note: Connection Idle is the state the terminals are in when a data channel exists between them, but no signaling is in process. It bridges the SCIP signaling specified herein and Native Mode signaling on the underlying network. Both SCIP and Native Mode signaling may be entered from this state. It is also the state the terminals can enter when a problem occurs while an attempt is made to resolve the problem. The terminal will be able to return from Connection Idle to try another Capabilities Exchange. It will also be able to execute some Notification related functions (e.g., Connection Terminate, Attention) defined in Section 2.3.2 and any non-SCIP native functions that are available from this state.

1466

1467



NOTES:

1. Connection Idle is the state the terminal is in when the initial clear data connection is established with the far end. It may also be entered as a result of a Failed Call.
2. Call setup may be requested either automatically or manually. It may be requested simultaneously at both terminals.
3. This path is included as a reminder that the Capabilities Exchange is not the only way to exit the Connection Idle state. It is not described in the text as it uses Host rather than SCIP signaling and thus may be different for each platform.
4. Criteria for requesting the Initiator or Responder role are intentionally not specified; however, some general guidelines are provided in the accompanying text.
5. This path is not SCIP standard signaling and as such is not discussed further.
6. Set I/R bit in Initiator Negotiation and/or Alternate Initiator Negotiation fields as specified in Section 2.2.2.1.
7. This permits the call to time out during initial call setup should the far-end device not be SCIP compatible.

Figure 2.2-3

Figure 2.2-2 Capabilities Message Transmission

1468
1469
1470
1471
1472

1473
1474 A terminal will determine through some mechanism (e.g., a Capabilities Message is received, a
1475 button on the console is pressed, automatic start of SCIP call establishment when the data
1476 channel becomes available, etc.) that a point-to-point SCIP call is to be established. It shall then
1477 format a Capabilities Message as specified in Section 2.2.2.1 and transmit it to the far end. All
1478 Operational Modes and Keysets available for use during the SCIP call are offered. Vendor
1479 unique native Operational Modes may also be offered and negotiated. The Initiator/Responder
1480 (I/R) bit of the Initiator Negotiation field and/or Alternate Initiator Negotiation field (see Section
1481 2.2.2.1) is set for the role (Initiator or Responder) desired for mode negotiation. There are no
1482 specific requirements for setting Initiator and Responder roles; however, the roles should be set
1483 in such a manner as to minimize the possibility of glare, i.e., two Initiators or two Responders.
1484 Possible approaches include setting to the opposite role of that indicated in a received
1485 Capabilities Message when a Capabilities Message is received before one is transmitted, setting
1486 the calling terminal as Initiator and the called terminal as Responder, setting a terminal as
1487 Initiator when no other information is available and a local call setup request occurs prior to
1488 receiving a Capabilities Message, etc. If a glare condition exists, it will be resolved using the
1489 Random Number portion of the Initiator Negotiation field or Alternate Initiator Negotiation field
1490 of the Capabilities Message as specified in Section 2.2.2.3.1.

1491
1492 The terminal shall then set a First Message Timer, since at this point during initial call setup it
1493 may not yet know that there is a SCIP compatible terminal at the far end. After setting the First
1494 Message Timer, the terminal will then wait to receive a Capabilities Message from the far end.
1495 Processing of the received Capabilities Message is specified in Section 2.2.2.3.

1496
Editor's Note: The starting and stopping of the First Message Timer may be done by a
terminal that has already received a Capabilities Message just to retain commonality of code,
but it is functionally superfluous to do this.

1497
1498
1499 **2.2.2.3 Capabilities Message Reception**

1500
1501 The processing of the Capabilities Message consists of Unique Processing and Common
1502 Processing. Unique Processing (see 2.2.2.3.1) of a received Capabilities Message occurs when
1503 the message is initially received. Common Processing (see 2.2.2.3.2) of the received
1504 Capabilities Message occurs

- 1505
1506
- when the message is initially received, and
 - when the received Capabilities Message must be reexamined because a
Parameters/Certificate Exchange determined that compatible Parameters do not exist
for the negotiated Operational Mode, there is a security incompatibility, or there is an
Access Control failure.
- 1507
1508
1509
1510
1511

1512 If either terminal transmits a Version 0 Capabilities Message, both terminals shall process
1513 Version 0 fields only. If both terminals transmit Message Version 1 or higher Capabilities
1514 Messages, the terminals shall process the entire portion of the message corresponding to the
1515 highest common version.

1516

1517

1518 **2.2.2.3.1 Capabilities Message Reception Unique Processing**

1519

1520 This Section discusses the processing of the Capabilities Message when it is received.

1521

1522 The Capabilities Message reception is shown in Figure 2.2-3. The processing of the timeout,
1523 that occurs if no message is received from the far end, was previously described in Section
1524 2.2.1.2.

1525

1526 If a terminal receives the far end's Capabilities Message before it has transmitted its own, it may
1527 begin processing the received Capabilities Message in parallel with generating its own so long as
1528 this does not delay transmission of its own Capabilities Message.

1529

1530 Upon receipt of a Capabilities Message, the terminal shall stop the First Message Timer, since it
1531 is now known that the far end terminal is SCIP compatible.

1532

1533 The Initiator terminal is now determined as follows. If both the transmitted and received
1534 Capabilities Messages are Version 1 or higher, the Alternate Initiator Negotiation field shall be
1535 used to determine the Initiator. If either Capabilities Message is Version 0, the Initiator
1536 Negotiation field shall be used. The Initiator Negotiation fields or Alternate Initiator
1537 Negotiation fields, treated as unsigned numbers, are compared. If the two values are equal, the
1538 terminal shall execute the Failed Call logic defined in Section 2.3.2.3.1 with the Information
1539 Code set to *no initiator defined*. Otherwise the Initiator and Responder roles will be adopted for
1540 Operational Mode choice in Section 2.2.2.3.2. The Initiator is the terminal that set the larger
1541 value in the field. (Note that if distinct Initiator and Responder roles were defined prior to this
1542 point, these roles are not changed. It is only when both terminals enter the Capabilities
1543 Exchange as Initiators or as Responders that this step has an impact and forces one of them to be
1544 an Initiator and one of them to be a Responder in subsequent steps.)

1545

1546 Signaling then continues as defined in Section 2.2.2.3.2.

1547

1548

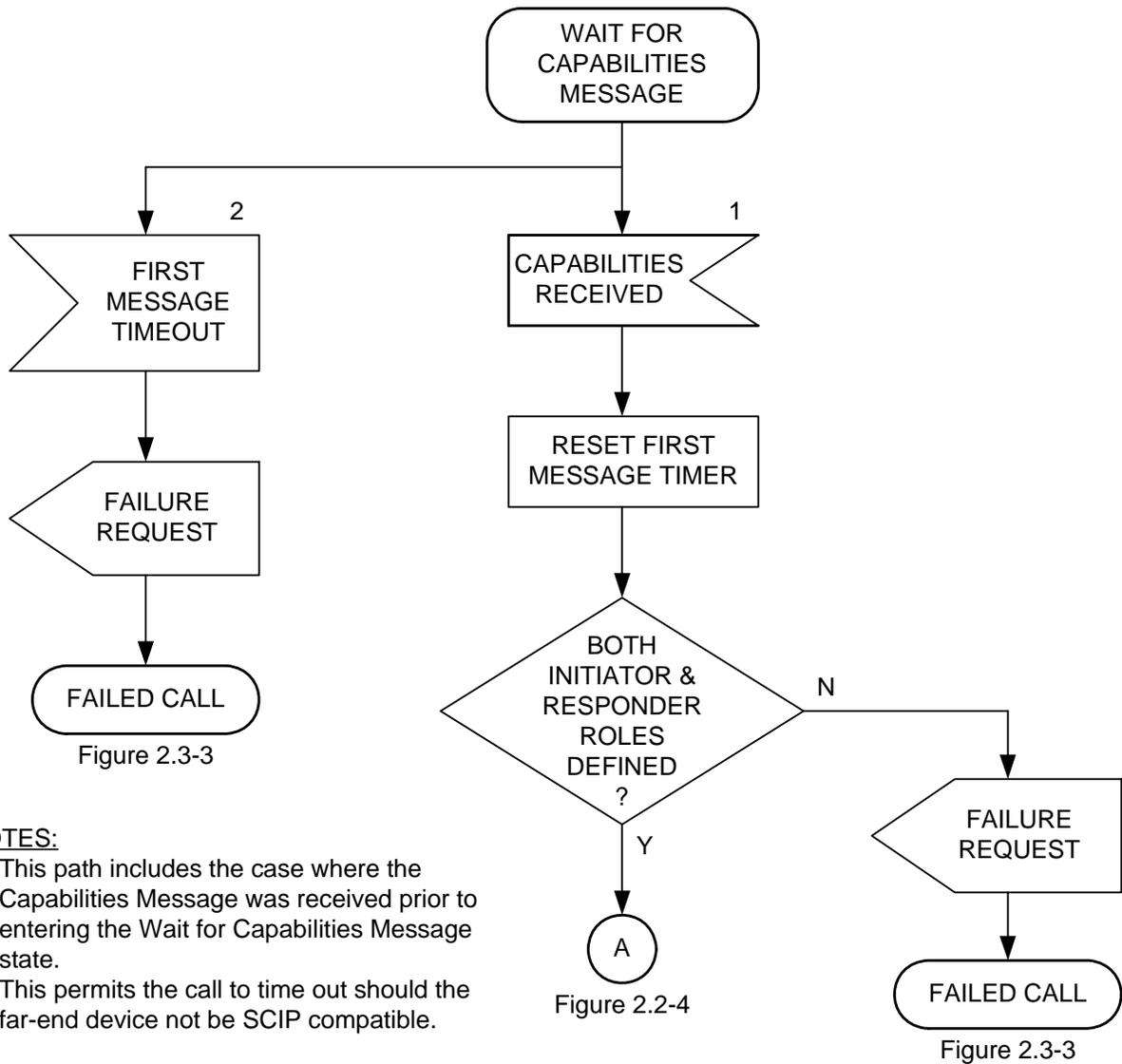


Figure 2.2-3 Capabilities Message Reception Unique Processing

2.2.2.3.2 Common Capabilities Message Processing

The signaling described in this section is shown in Figure 2.2-4 and starts with the Operational Mode choice process. This processing can be entered from two places, and the rules for choosing the Operational Mode are slightly different in the two cases. The Initiator and Responder terminals for purposes of Operational Mode choice shall be determined as specified in Section 2.2.2.3.1. For any secure Operational Mode to be chosen, Keysets compatible with the Operational Mode and with each other shall exist in the Keysets Lists of the two terminals.

1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562

1563 **Case 1.** The initial entry point is from Capabilities Message Reception (Section
1564 2.2.2.3.1). At this point the terminal has received and processed the far end's Capabilities
1565 Message. The terminal shall choose the first Operational Mode Entry on the Initiator's
1566 Operational Modes List which is also on the Responder's Operational Modes List. Note
1567 that if Electronic Rekey is offered by the far-end terminal (indicating it is a SCIP-LIT),
1568 this mode shall be chosen, since it is the only Operational Mode offered by the LIT.
1569

1570 **Case 2.** The two terminals have performed a Parameters/Certificate Exchange. At this
1571 point they discover that while they share a common Operational Mode, they do not have
1572 compatible Parameters, there is a security incompatibility, or there is an Access Control
1573 failure for that Mode (see Section 2.2.3.3). If any of these occur, the terminals shall
1574 choose the next entry on the Initiator's Operational Mode List which is also on the
1575 Responder's List.
1576

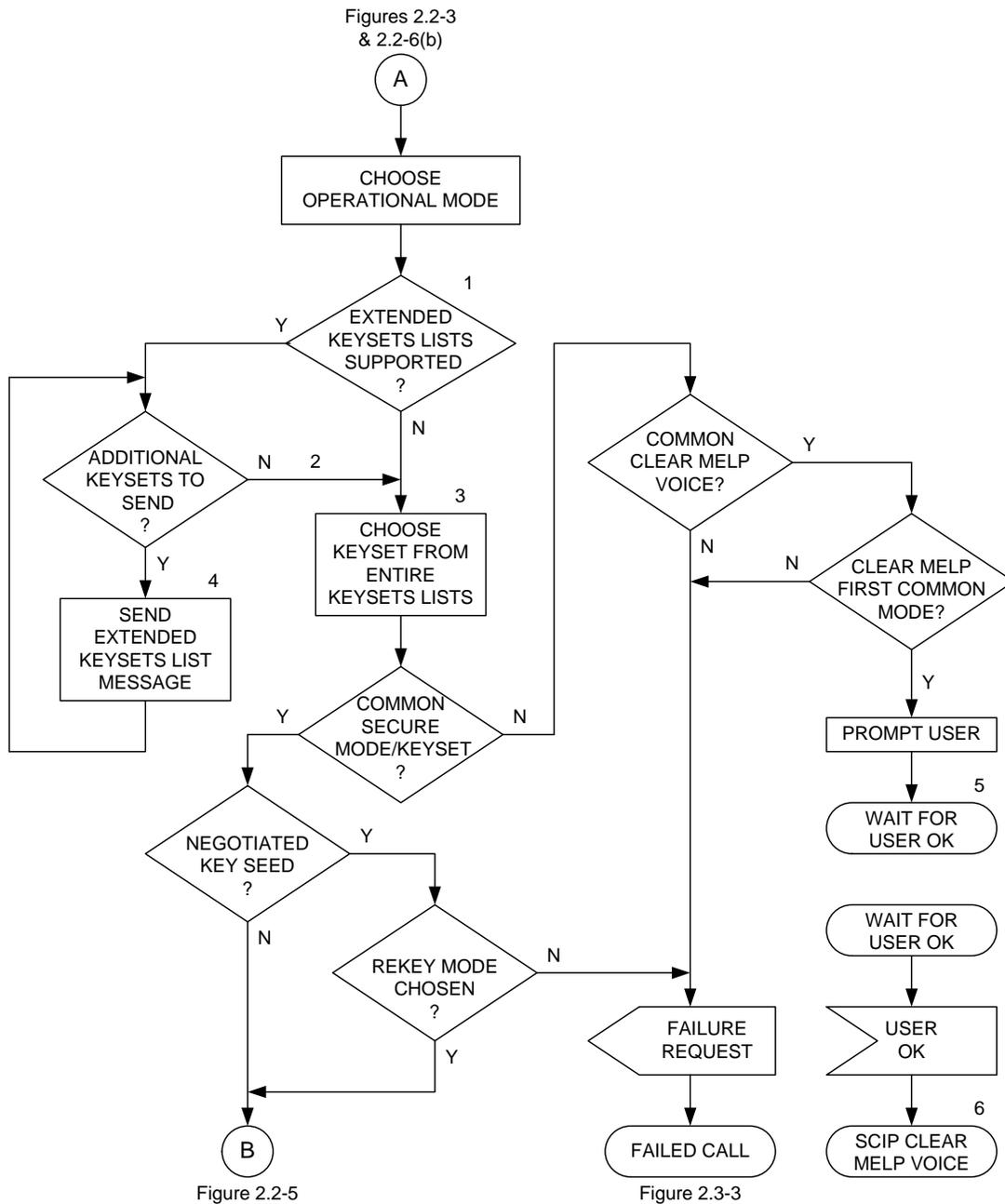
1577 If there is no common Operational Mode (Case 1) on the Initiator's list that meets the choice
1578 process as specified above (other than Native Clear Voice, which is entered via Failed Call), the
1579 terminal shall execute Failed Call processing (defined in Section 2.3.2.3.1) with an Information
1580 Code of *no common operational modes*. If there is no alternate common Operational Mode
1581 (Case 2) that meets the choice process, the terminal shall execute Failed Call processing with an
1582 Information Code of *no matching parameters, security incompatibility, or access control failure*,
1583 as is appropriate for the problem encountered with the Operational Mode chosen initially.
1584

Editor's Note: It is expected that the Capabilities Exchange will be the first exchange for all
SCIP terminals. However, except for choosing a common non-standard Operational Mode
using this exchange, the signaling associated with non-standard Operational Modes is not
addressed in this Signaling Plan. Part of the definition of a non-standard Operational Mode is
the definition of the associated call setup and call control signaling. While many non-standard
Operational Modes will choose to piggyback on the standard call setup and call control
exchanges, this is not required.

1585
1586 If a standard secure Operational Mode is chosen, a Keyset that is compatible with a Keyset in the
1587 other terminal shall also be chosen. If both the transmitted and received Capabilities Messages
1588 are Version 1 or higher and both terminals are in the Interoperable Terminal Priority COI, the
1589 Keyset Initiator shall be determined as follows.

- 1590 • If the Terminal Priority fields contain different values, the terminal with the larger
1591 Terminal Priority value shall be the Keyset Initiator.
- 1592 • If the Terminal Priority fields are the same, the Keyset Initiator shall be the same as the
1593 Initiator determined in Section 2.2.2.3.1.

1594 If either of the Capabilities Messages is Version 0, the Keyset Initiator shall be the same as the
1595 Initiator determined in Section 2.2.2.3.1. Finally, a terminal in the Interoperable Terminal
1596 Priority COI shall be the Keyset Initiator when it attempts to communicate with a terminal that is
1597 not in the Interoperable Terminal Priority COI.
1598
1599



NOTES:

1. Extended Keysets List Messages may be received and optionally transmitted if an Extended Keysets List Support Keysset is identified in the Capabilities Messages of both terminals.
2. The keysset cannot be chosen until all Keysets List(s) have been received.
3. A terminal cannot choose another keysset if there is a problem with the negotiated keysset.
4. Messages may be sent independently by the Initiator, Responder, or both.
5. The user also has the option of choosing to terminate the call.
6. SCIP Clear MELP Voice as specified in Section 3.3. Vendor unique SCIP clear voice modes may also follow this path.

1600
1601
1602

Figure 2.2-4 Common Capabilities Message Processing

1603
1604 Terminals implementing SCIP-210, Rev. 3.2 or later shall support the ability to receive and
1605 process the Extended Keysets List Messages as specified below. This capability is indicated
1606 using the Extended Keysets List Support Keyset Type and associated Additional Keysets
1607 parameter, as specified in Section 2.2.6.1.1.9. Fielded terminals that implement prior versions of
1608 SCIP-210 may not support this capability and may only negotiate the keyset using the keysets
1609 listed in the Keysets List of the Capabilities Message. The ability to transmit an Extended
1610 Keysets List Message is optional for all SCIP products.

1611
1612 Extended Keysets List Messages are transmitted only after the exchanged Capabilities Messages
1613 indicate that both terminals support Extended Keysets List Messages. A SCIP terminal shall
1614 only send an Extended Keysets List Message to a SCIP terminal that has indicated, in its
1615 Capabilities Message, that it supports Extended Keysets List Messages. If no Extended Keysets
1616 List Messages are exchanged, the Keyset shall be chosen using the keyset selection rules
1617 specified in SCIP-230 and SCIP-232, Section 2.1.1.1, and SCIP-231, Section 2.1.1

1618
1619 If Extended Keysets List Messages are supported by both terminals and there are remaining
1620 keysets that are not included within the Keysets List of the Capabilities Message, the terminal
1621 shall set the Additional Keysets parameter within the Extended Keysets List Support Keyset to
1622 indicate that it has more keysets to send, as specified in Section 2.2.6.1.1.9. The terminal shall
1623 then transmit an Extended Keysets List Message, as specified in Section 2.2.2.4. These
1624 messages are sent independently by the Initiator, Responder, or both.

1625
1626 The last entry in the Extended Keysets List shall be the Extended Keysets List Support Keyset.
1627 The Additional Keysets parameter within this Keyset will indicate if any more keysets exist that
1628 need to be sent in additional Extended Keysets List Message(s). In such cases, additional
1629 Extended Keysets List Messages shall be sent until all of the necessary keysets have been
1630 transmitted. The terminal shall then set the Additional Keysets parameter within the Extended
1631 Keysets List Support Keyset of the last Extended Keysets List Message to indicate that it does
1632 not have any more keysets to send.

1633
1634 Note that the entire Keysets List need not be transmitted during call setup. There are many
1635 reasons that a terminal may choose to identify a subset of its keysets in a given call setup
1636 message exchange. However, care must be taken when implementing any specific optimization
1637 for sending keysets. The final result of the optimized keyset exchange shall be the same keyset
1638 selection as if both terminals exchanged their entire Keysets Lists. For example, a terminal may
1639 recognize that the remote terminal has already transmitted all of its keysets and can, therefore,
1640 identify the specific keyset that should be negotiated, thereby eliminating the need to send
1641 keysets that will never be selected. If this specific keyset entry has already been transmitted, the
1642 terminal may transmit an Extended Keysets List Message which only contains the Extended
1643 Keysets List Support Keyset indicating that the terminal does not have any more keysets to send.

1644
1645 The receiving terminal shall process the Extended Keysets List Messages as they are received, as
1646 specified in Sections 2.2.2.4 and 2.2.6.1.1.9. If the Additional Keysets parameter (received in
1647 the Capabilities or Extended Keysets List Message) indicates that additional keysets will not be
1648 offered, all the keysets lists have been received and the terminal shall choose the Keyset as if the

1649 keysets list in the Capabilities Message, and the keysets lists from all the Extended Keysets List
1650 Messages were appended in the order received, and had been sent in the original Capabilities
1651 Message. The Extended Keysets List Support Keysets shall not be included in the appended
1652 Keysets List since these Keysets are only used to determine subsequent keyset processing and
1653 are, therefore, never negotiated. Negotiation proceeds according to the keyset selection rules
1654 specified in SCIP-230 and SCIP-232, Section 2.1.1.1; and SCIP-231, Section 2.1.1. Note that
1655 since there is no limit on the number of Extended Keysets List Messages, these Messages may be
1656 processed as they are received and then discarded as long as these keyset selection rules are
1657 followed.

1658
1659 If the terminals do not have compatible Keysets and Clear MELP Voice is not supported by both
1660 terminals, the terminal shall execute Failed Call processing with an Information Code of *no*
1661 *compatible keysets*. If there is a common secure Operational Mode, the terminals have
1662 compatible Keysets, and the negotiated Keyset is not seed key, processing continues as defined
1663 in Section 2.2.3.2.

1664
1665 If no common secure Operational Mode and/or Keyset is available but both terminals support
1666 Clear MELP Voice, the terminal shall proceed as follows. If Clear MELP Voice is the first
1667 common Operational Mode, the terminal shall prompt the user and wait for an acknowledgment
1668 before entry into the Clear MELP Voice Operational Mode. The terminal shall initiate the SCIP
1669 Clear MELP Voice application as specified in Section 3.2.1. If Clear MELP Voice is not the
1670 first common Operational Mode, the terminal shall execute Failed Call processing with an
1671 Information Code of either *no common operational modes* or *no compatible keysets*, as
1672 appropriate (i.e., Clear MELP Voice cannot be an alternate mode choice from a Capabilities
1673 exchange; it is entered via Failed Call and another Capabilities exchange – see Section
1674 2.3.2.3.1.).

1675
Editor's Note: Note that as defined in Section 3.3.1.3, the Clear MELP Voice application is
not actually entered until all outstanding framed messages are acknowledged.

1676
1677 If the negotiated Keyset is seed key and the chosen Operational Mode is not Rekey, the terminal
1678 shall execute Failed Call processing with an Information Code of *seed key held*. If the
1679 negotiated Keyset is seed key and the chosen Operational Mode is Rekey, processing continues
1680 as defined in Section 2.2.3.2.

1681 1682 1683 **2.2.2.4 Extended Keysets List Message Definition**

1684
1685 If the Capabilities Message exceeds the total message length limitation specified in Section
1686 2.2.1.4, any remaining keysets that will not fit within the Keysets List of the Capabilities
1687 Message may be listed in one or more Extended Keysets List Message(s). The rules for
1688 processing the Extended Keysets List Messages are specified in Section 2.2.2.3.2. The format of
1689 the Extended Keysets List Message is shown in Table 2.2-3(f).
1690

1691
1692
1693

Table 2.2-3(f) Extended Keysets List Message Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
Source ID								
0	0	0	0	0	0	1	1-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
Sequence Number								
X-msb	X	X	X	X	X	X	X-lsb	6
Extended Keysets List Length								
X-msb	X	X	X	X	X	X	X	7
X	X	X	X	X	X	X	X-lsb	8
Extended Keysets List								
X	X	X	X	X	X	X	X	9
•••								
X	X	X	X	X	X	X	X	8+M

1694 M = Length of Extended Keysets List field.

1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710

- For the Extended Keysets List Message the value of the MID is 0x0003.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the Extended Keysets List Message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The Sequence Number shall contain a unique number assigned to each Extended Keysets List Message. The value of the field shall be monotonically incremented from 0x01 for each sequential Extended Keysets List Message.
- The Extended Keysets List Length shall contain the actual length of the Extended Keysets List (plus the length of the Extended Keysets List Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.

- 1711
- The Extended Keypsets List contains keysets list entries of the form given in Table
1712 2.2-3(a). Only Keypsets not previously listed shall have a keysets list entry on the
1713 Extended Keypsets List. Keypsets list entries in the Extended Keypsets List Message
1714 shall be listed as specified for the keysets list entries in the Capabilities Message.
- 1715

1716

1717 **2.2.3 Parameters/Certificate Message**

1718

1719 If a secure Operational Mode is chosen, the second step in SCIP Call Setup is the exchange of
1720 Parameters/Certificate Messages. The Credentials used by the SCIP Signaling have two parts, a
1721 Certificate and an F(R). These are exchanged in separate messages. Any parameters which must
1722 be negotiated for the chosen secure Operational Mode are also negotiated at this time. If a PPK
1723 Keypset is chosen, the Parameters/Certificate Messages are exchanged without the Certificate. If
1724 Clear MELP Voice is chosen, Credentials will not be exchanged (see also Section 2.2.6.5).

1725

1726

1727 **2.2.3.1 Parameters/Certificate Message Definition**

1728

1729 The format of the Parameters/Certificate Message is shown in Table 2.2-4.

1730

1731
1732
1733

Table 2.2-4 Parameters/Certificate Message Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
Source ID								
0	0	0	1	0	0	0	0-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
Operational Mode								
X-msb	X	X	X	X	X	X	X	6
Source ID								
X	X	X	X	X	X	X	X-lsb	7
Keyset Type								
X-msb	X	X	X	X	X	X	X	8
Source ID								
X	X	X	X	X	X	X	X-lsb	9
Keyset ID Length								
X-msb	X	X	X	X	X	X	X	10
X	X	X	X	X	X	X	X-lsb	11
Keyset ID								
X	X	X	X	X	X	X	X	12
...								
X	X	X	X	X	X	X	X	11+N
Parameters Length								
X-msb	X	X	X	X	X	X	X	12+N
X	X	X	X	X	X	X	X-lsb	13+N
Operational Mode Parameters (Optional)								
X	X	X	X	X	X	X	X	14+N
...								
X	X	X	X	X	X	X	X	13+N+L
Certificate Length								
X-msb	X	X	X	X	X	X	X	14+N+L
X	X	X	X	X	X	X	X-lsb	15+N+L
Certificate (Optional)								
X	X	X	X	X	X	X	X	16+N+L
...								
X	X	X	X	X	X	X	X	15+N+L+K

1734 N = Length of Keyset ID. L = Length of Operational Mode Parameters. K = Length of Certificate.

1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774

- For the Parameters/Certificate Message the value of the MID is 0x0010.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the Parameters/Certificate Message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The Operational Mode field shall contain the ID of the chosen Operational Mode. For the format and values of these IDs, see the definition of Operational Mode IDs in Section 2.2.2.1. The high order bit of the Operational Mode ID is placed in bit 8 of octet 6, and the low order bit of the Operational Mode ID is placed in bit 1 of octet 7.
- The Keypset Type field shall identify the type of the chosen Keypset. For the format and values of these Types, see the definition of Keypset Type in Section 2.2.2.1.
- The Keypset ID Length field shall contain the length, in octets, of the Keypset ID field (plus the length of the Keypset ID Length itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 10 and the low order bit being bit 1 of octet 11.
- The Keypset ID field shall contain the ID of the chosen Keypset. Keypset IDs are unique to each Keypset Type. For each standard Keypset Type, the length and format of the corresponding Keypset ID are defined in Section 2.2.6.
- The Parameters Length field shall contain the length, in octets, of the Operational Mode Parameters field (plus the length of the Parameters Length itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.
- The Operational Mode Parameters shall contain parameters for the chosen Operational Mode. The length, format and values of the Operational Mode Parameters are unique to each Operational Mode and are defined in Section 2.2.6 for each standard Operational Mode having a Parameters/Certificate Exchange. This field is optional and is not present unless Parameters are defined for a given Operational Mode.
- The Certificate Length field shall contain the length, in octets, of the Certificate field (plus the length of the Certificate Length itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.
- The Certificate field shall contain the Certificate for the chosen Keypset. The length, format and contents are unique to each key exchange type and are defined in Section 2.2.6 for each key exchange type. This field is optional and is not present when a PPK Keypset is chosen.

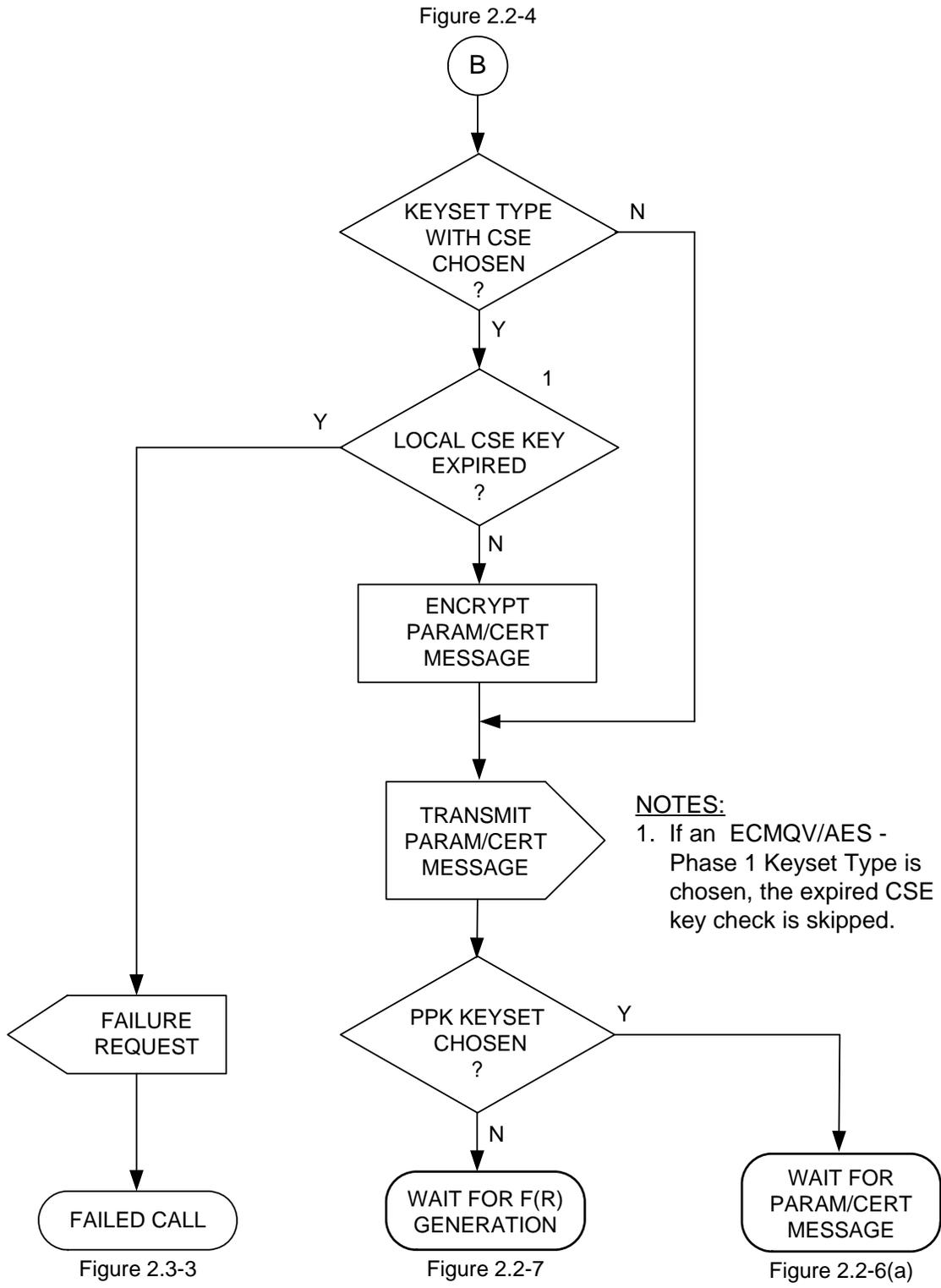
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796

2.2.3.2 Parameters/Certificate Message Transmission

Parameters/Certificate Message transmission is shown in Figure 2.2-5.

If a standard secure Operational Mode was chosen via the processing defined in Section 2.2.2.3.2, the following shall be performed. Both the parameters for the chosen Operational Mode and the Certificate (if applicable) of the chosen Keyset shall be transmitted in a Parameters/Certificate Message formatted as defined in Section 2.2.3.1. If the chosen Keyset is a Keyset Type with CSE and the local CSE key is not expired, the Parameters/Certificate Message shall be encrypted as specified in SCIP-230 or SCIP-231, Section 4.1.4; or SCIP-232, Section 4.4, prior to transmission. Signaling then continues as defined in Section 2.2.4.2. If a PPK Keyset is chosen, signaling continues as defined in Section 2.2.3.3.

The check for expired CSE key consists of comparing the Expiration Date of the local CSE key with the terminal's System Date (year/month) as specified in SCIP-230 or SCIP-232, Section 2.1.1.3.1.3. If the Expiration Date of the local CSE key is earlier, the terminal shall execute Failed Call processing (see Section 2.3.2.3.1) with an Information Code of *Local CSE key expired*. If an ECMQV/AES - Phase 1 Keyset Type is chosen, the expired CSE key check is skipped.



1797
 1798
 1799
 1800

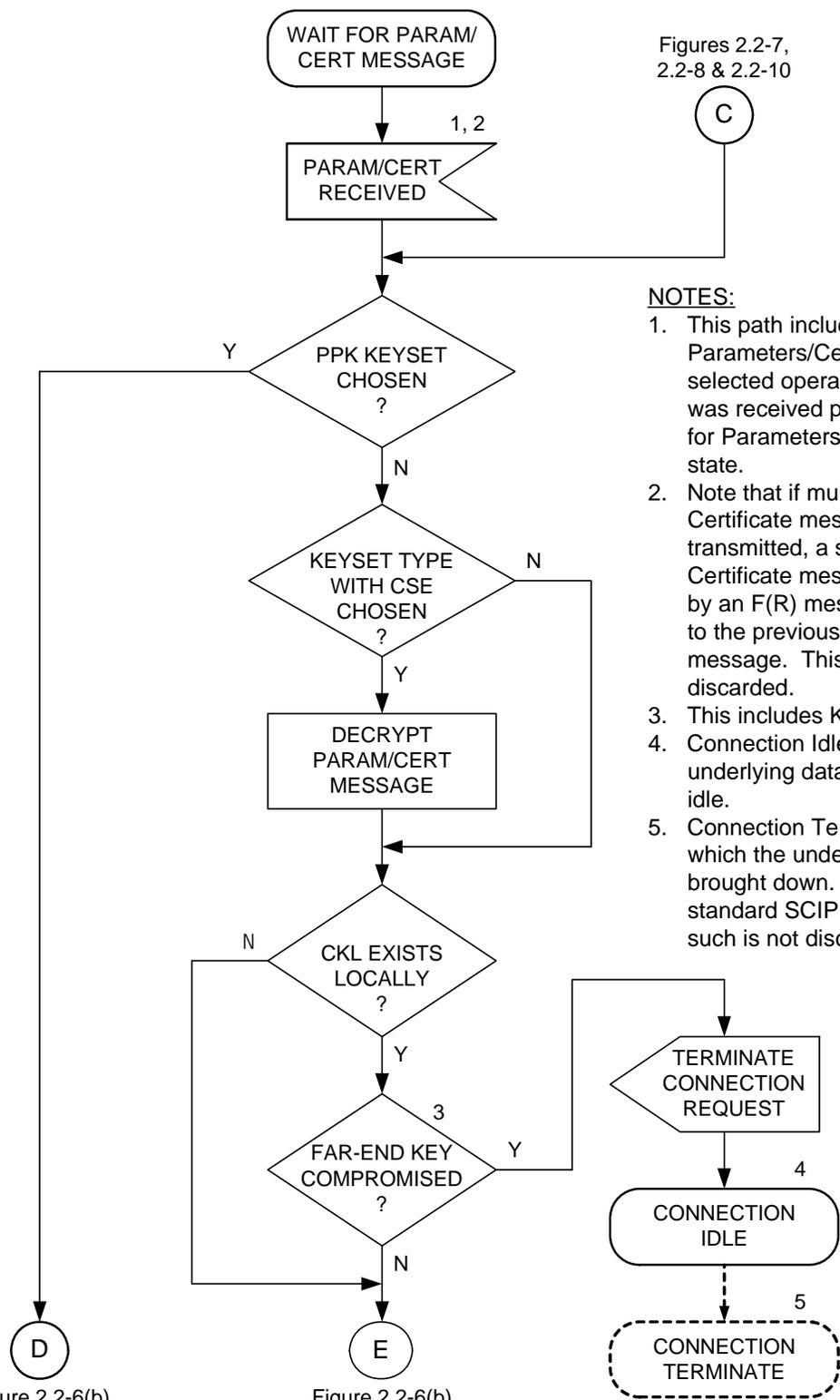
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819

2.2.3.3 Parameters/Certificate Message Reception

Parameters/Certificate Message reception is shown in Figure 2.2-6(a), Figure 2.2-6(b), and Figure 2.2-6(c).

The terminal may begin processing the far end's Parameters/Certificate Message, for the chosen Operational Mode and Keypset, when it is received. If a terminal receives the far end's Parameters/Certificate Message before it has transmitted its own Parameters/Certificate Message, it may begin processing the received Parameters/Certificate Message in parallel with generating its own Parameters/Certificate Message so long as this does not delay the transmission of its own message.

If the chosen Keypset is a Keypset Type with CSE, the received Parameters/Certificate Message is encrypted. Prior to processing, it shall be decrypted as specified in SCIP-230 or SCIP-231, Section 4.1.4; or SCIP-232, Section 4.4.



Figures 2.2-7,
2.2-8 & 2.2-10

NOTES:

1. This path includes the case where the Parameters/Certificate message for the selected operational mode and keyset was received prior to entering the 'Wait for Parameters/Certificate Message' state.
2. Note that if multiple Parameters/Certificate messages must be transmitted, a subsequent Parameters/Certificate message may be preceded by an F(R) message that corresponds to the previous Parameters/Certificate message. This F(R) message is discarded.
3. This includes Key Cutoff Date failure.
4. Connection Idle is a state in which the underlying data channel is alive but idle.
5. Connection Terminate is a state in which the underlying data channel is brought down. This path is not standard SCIP signaling, and as such is not discussed any further.

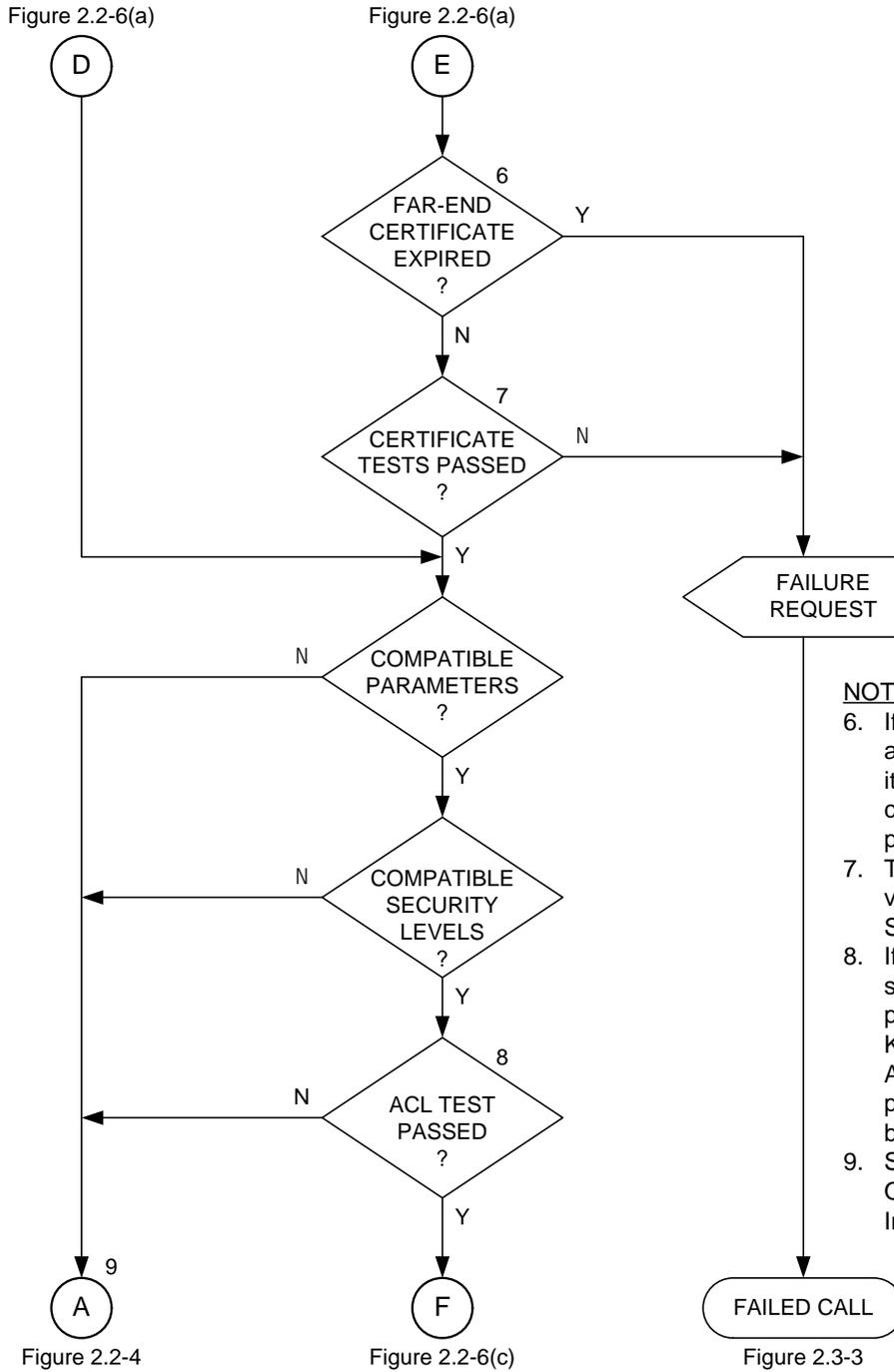
Figure 2.2-6(b)

Figure 2.2-6(b)

Figure 2.2-6(a) Parameters/Certificate Message Reception

1820
1821
1822

1823



NOTES:

- 6. If the terminal has neither a CKL nor a System Date, it cannot perform the expired certificate test, and the "No" path is taken.
- 7. These are the Certificate verification tests defined in SCIP-23x.
- 8. If the chosen Keyset Type is not supported by an ACL, the "Yes" path is taken. If the chosen Keyset Type is supported by an ACL, the ACL test will be performed only if the ACL has been activated in the terminal.
- 9. Select next common Operational Mode on Initiator's list.

1824
1825
1826

Figure 2.2-6(b) Parameters/Certificate Message Reception (Cont.)

1827

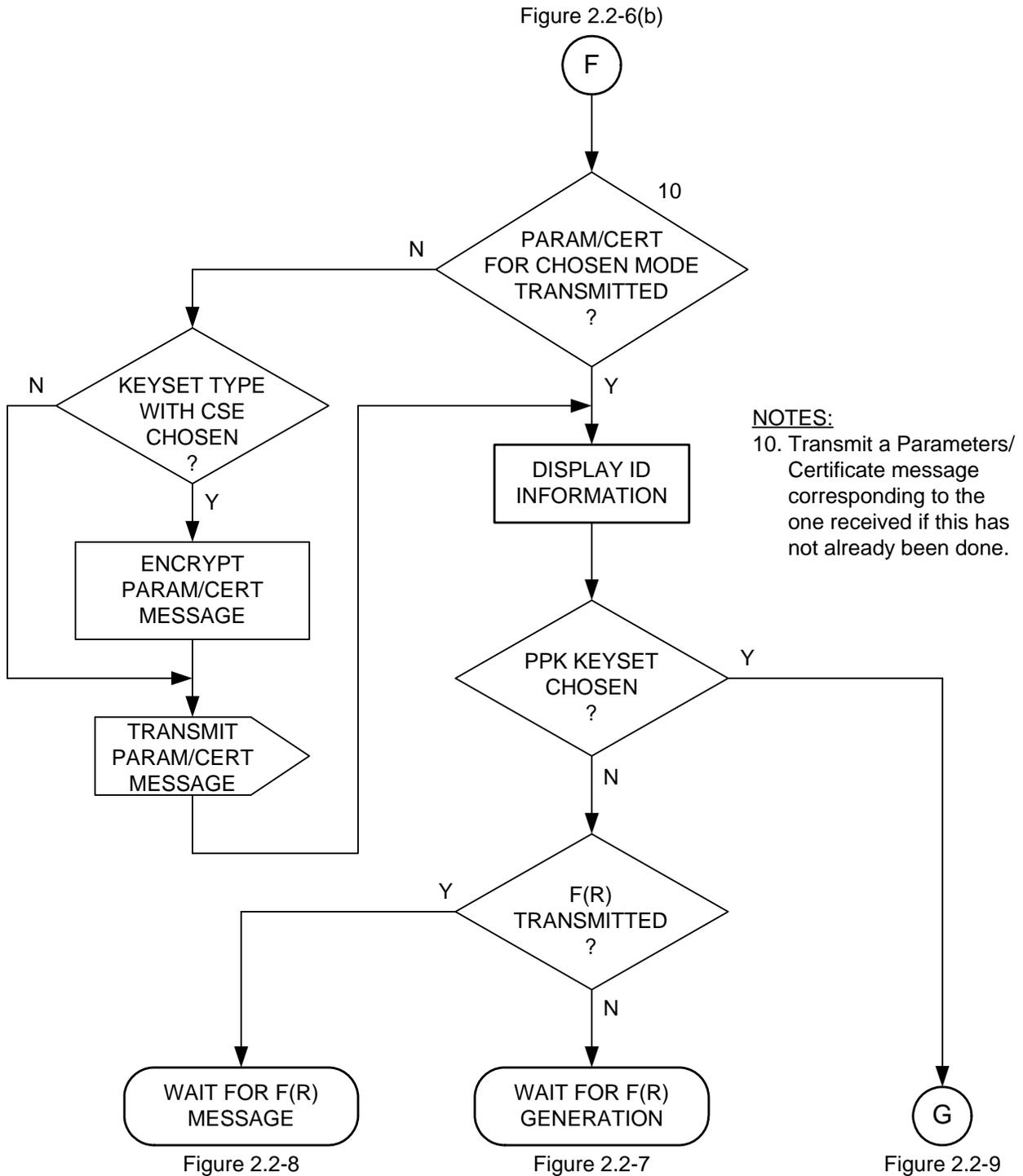


Figure 2.2-6(c) Parameters/Certificate Message Reception (Cont.)

1828
 1829
 1830
 1831
 1832

1833
1834 The following applies to FIREFLY (defined in SCIP-230) and NATO ECMQV (defined in
1835 SCIP-232) Certificates only.

1836
1837 If the terminal does not have a local CKL for the chosen Universal Edition, the
1838 compromised key check below is skipped. If the terminal has no System Date (see SCIP-
1839 230, Section 2.1.2.3.2, or SCIP-232, Section 2.1.2.3.1), the expired key check below is
1840 also skipped.

1841
1842 The test for compromised key consists of checking for the received Certificate's KMID
1843 on the CKL and also comparing the Expiration Date in the Certificate with the Key
1844 Cutoff Date in the CKL (see SCIP-230 or SCIP-232, Sections 2.1.2.2.1 and 2.1.2.2.2). If
1845 the received Certificate's KMID is on the local CKL for that Universal Edition, or if the
1846 Expiration Date in the Certificate is earlier than the Key Cutoff Date in the CKL, the
1847 terminal shall terminate the connection immediately and without providing the far end
1848 terminal with any indication (i.e., without sending a Notification Message).

1849
Editor's Note: A compromised Certificate is no longer carried on a CKL when its expiration
date is earlier than the CKL's Key Cutoff Date. The CKL design assumes that such an expired
key will not be communicated with.

1850
1851 The check for expired key consists of comparing the Expiration Date in the Certificate
1852 with the terminal's System Date (year/month) as specified in SCIP-230 or SCIP-232,
1853 Section 2.1.2.2.3. If the Expiration Date in the Certificate is earlier, the terminal shall
1854 execute Failed Call processing (see Section 2.3.2.3.1) with an Information Code of
1855 *Certificate expired*.

1856
1857 The Certificate verification tests specified in SCIP-230, Sections 2.1.1.4.2 and 2.1.1.4.3,
1858 or SCIP-232, Appendix F, are now performed. For Electronic Rekey, an additional test is
1859 performed to verify that the far-end terminal is a SCIP-LIT (See SCIP-230, Section 6.1,
1860 or SCIP-232, Appendix E.1). If the received Certificate fails any of these tests, the
1861 terminal shall execute Failed Call processing with an Information Code of *Certificate*
1862 *verification failure*.

1863
1864 The following applies only to the Phase 1 X.509 Certificate as defined in SCIP-231.

1865
1866 The Certificate verification tests specified in SCIP-231, Sections 2.1.3.3.2 and 2.1.3.3.4,
1867 are now performed. If the received Certificate fails any of these tests, the terminal shall
1868 execute Failed Call processing with an Information Code of *Certificate verification*
1869 *failure*.

1870
1871

1872 The Operational Mode Parameters are now checked. For standard secure modes, the Operational
1873 Mode Parameters contain an Options List. (See Section 2.2.6.2 for Secure Voice Options,
1874 Section 2.2.6.3 for Secure Data Options, and Section 2.2.6.4 for Secure Electronic Rekey
1875 Options.) The Options List Entries will be examined in the order in which they appear. The first
1876 entry on the Initiator's Options List that is supported by the Responder shall be chosen.

1877
1878 If no entry on the Initiator's Options List is supported by the Responder, the Operational Mode is
1879 noted as one that has no compatible parameters and is not to be chosen. Processing then
1880 continues as specified in Section 2.2.2.3.2 (Case 2), where the terminals will attempt to choose
1881 another Operational Mode.

1882
1883 For Secure Data or Secure Voice if there is no overlap among the local and far-end Security
1884 Level settings and key classifications for the chosen Operational Mode and Keyset (see SCIP-
1885 230 or SCIP-232, Section 2.1.3.2), the Operational Mode is noted as one that has a security
1886 incompatibility and is not to be chosen. Processing again continues as specified in Section
1887 2.2.2.3.2 (Case 2) where the terminals will attempt to choose another Operational Mode.

1888
1889 If the Access Control List (ACL) has been activated for the chosen Operational Mode and the
1890 chosen Keyset Type is supported by an ACL, the terminal performs the ACL test as specified in
1891 SCIP-230 or SCIP-232, Section 2.1.3.1.1. If the ACL test fails, the Operational Mode is noted
1892 as one for which access is denied and is not to be chosen. Processing then continues as specified
1893 in Section 2.2.2.3.2 (Case 2) where the terminals will attempt to choose another Operational
1894 Mode. If the ACL has not been activated for the chosen Operational Mode and/or the chosen
1895 Keyset Type is not supported by an ACL, the ACL check is skipped.

1896
1897 If all the above tests pass, the terminal shall verify that the received Parameters/Certificate
1898 Message just processed corresponds to (i.e., contains the same Operational Mode as) the last
1899 Parameters/Certificate Message it transmitted. If it does not, the terminal shall transmit a
1900 Parameters/Certificate Message corresponding to the one just processed. If the chosen Keyset is
1901 a Keyset Type with CSE, the Parameters/Certificate Message shall be encrypted as specified in
1902 SCIP-230 or SCIP-231, Section 4.1.4; or SCIP-232, Section 4.4, prior to transmission. The
1903 authentication information is displayed to the user, as defined for each specific Keyset Type in
1904 SCIP-230, Sections 2.1.1.4.2.3 and 2.1.1.8.1, SCIP-231, Section 2.1.3.4, or SCIP-232, Sections
1905 2.1.1.4.1.4 and 2.1.1.8.1. If a PPK Keyset is chosen, processing proceeds as defined in Section
1906 2.2.5.2 (no F(R) message is transmitted). Otherwise, processing proceeds as defined in Section
1907 2.2.4.2 (if the F(R) has not yet been transmitted) or Section 2.2.4.3 (if the F(R) has already been
1908 transmitted).

1909
1910

1911 **2.2.4 F(R) Message**

1912
1913

1914 **2.2.4.1 F(R) Message Definition**

1915

1916 The format of the F(R) Message is shown in Table 2.2-5.

1917

1918
1919
1920

Table 2.2-5 F(R) Message - General Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
Source ID								
0	0	0	0	0	1	0	0-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
Operational Mode								
X-msb	X	X	X	X	X	X	X	6
Source ID								
X	X	X	X	X	X	X	X-lsb	7
Keyset Type								
X-msb	X	X	X	X	X	X	X	8
Source ID								
X	X	X	X	X	X	X	X-lsb	9
Keyset ID Length								
X-msb	X	X	X	X	X	X	X	10
X	X	X	X	X	X	X	X-lsb	11
Keyset ID								
X	X	X	X	X	X	X	X	12
...								
X	X	X	X	X	X	X	X	11+N
F(R) Length								
X-msb	X	X	X	X	X	X	X	12+N
X	X	X	X	X	X	X	X-lsb	13+N
F(R)								
X	X	X	X	X	X	X	X	14+N
...								
X	X	X	X	X	X	X	X	13+N+L

1921 N = Length of Keyset ID. L = Length of F(R) field.

1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965

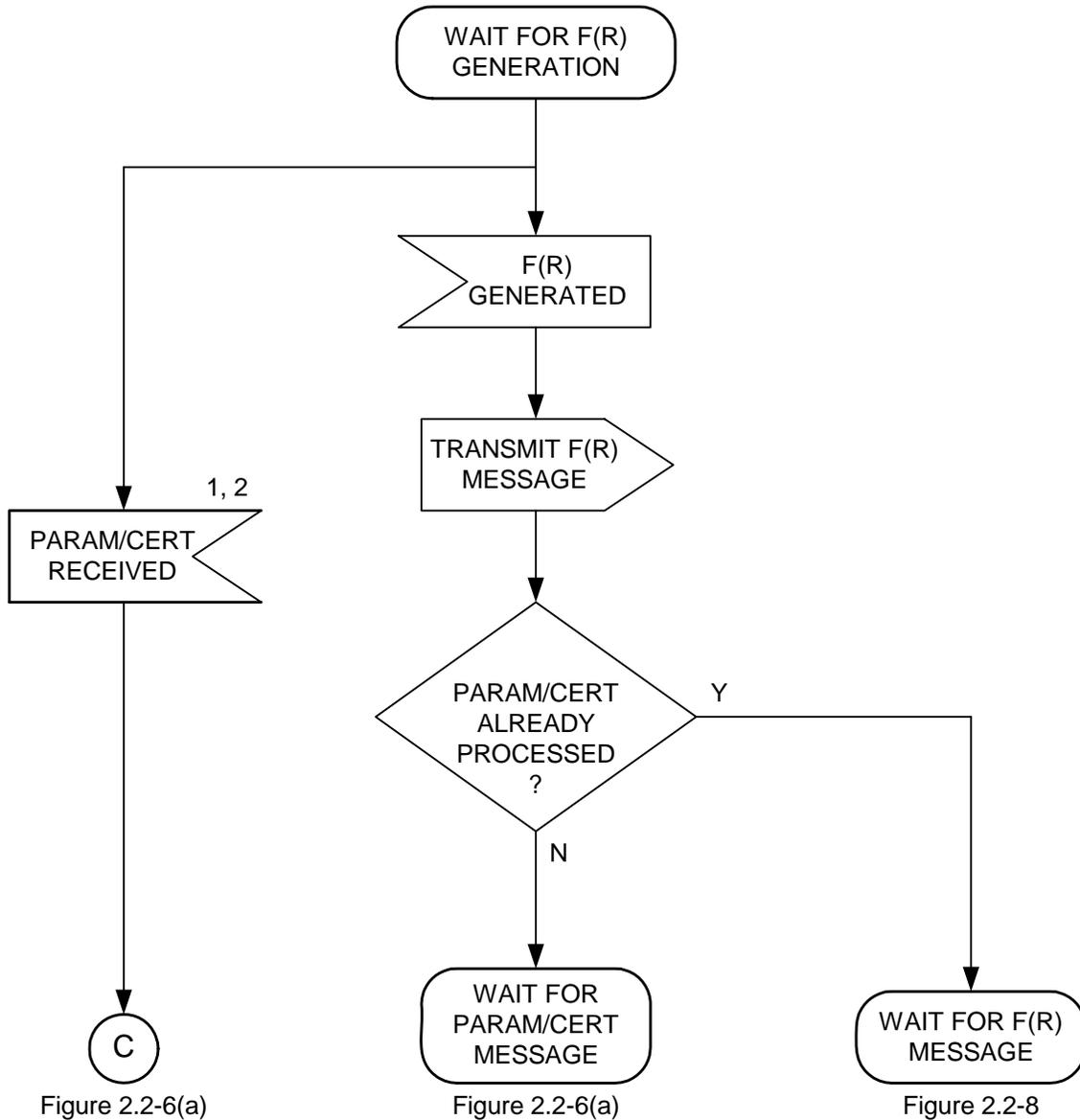
- For the F(R) Message the value of the MID is 0x0004.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the F(R) Message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The Operational Mode field shall contain the ID of the chosen Operational Mode. For the format and values of these IDs, see the definition of Operational Mode IDs in Section 2.2.2.1. The high order bit of the Operational Mode ID is placed in bit 8 of octet 6 and the low order bit of the Operational Mode ID is placed in bit 1 of octet 7.
- The Keyset Type field shall identify the type of the chosen Keyset. For the format and values of these Types, see the definition of Keyset Type in Section 2.2.2.1.
- The Keyset ID Length shall contain the actual length of the Keyset ID field (plus the length of the Keyset ID Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 10 and the low order bit being bit 1 of octet 11.
- The Keyset ID field shall contain the ID of the chosen Keyset. Keyset IDs are unique to each Keyset Type. For each standard Keyset Type, the length and format of the corresponding Keyset ID are defined in Section 2.2.6.1.
- The F(R) Length shall contain the actual length of the F(R) field (including the length of the F(R) Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.
- The F(R) field shall contain an F(R) corresponding to the chosen Keyset. The length, format and contents are unique to each key exchange type and are defined in Section 2.2.6.1 for each key exchange type.

2.2.4.2 F(R) Message Transmission

F(R) Message transmission is shown in Figure 2.2-7. At this point the Parameters/Certificate Message has been formatted and transmitted. If the far-end Parameters/Certificate Message arrives before the F(R) has been generated, the incoming Parameters/Certificate Message is first processed as described in Section 2.2.3.3.

If F(R) is not already available, F(R) generation proceeds to completion. An F(R) Message containing the F(R) for the chosen Keyset, and formatted as defined in Section 2.2.4.1, shall be transmitted to the far end. If the incoming Parameters/Certificate Message has already been processed, the terminal proceeds as defined in Section 2.2.4.3. Else the terminal waits until it receives the Parameters/Certificate Message from the far end, at which point it proceeds as defined in Section 2.2.3.3.

1966



NOTES:

1. This path includes the case where the Parameters/Certificate Message for the selected Operational Mode and Keypad was received prior to entering the Wait for F(R) Generation state.
2. Note that if multiple Parameters/Certificate Messages must be transmitted, a subsequent Parameters/Certificate Message may be preceded by an F(R) Message that corresponds to the previous Parameters/Certificate Message. This F(R) Message is discarded.

1967
1968
1969
1970

Figure 2.2-7 F(R) Message Transmission

1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006

2.2.4.3 F(R) Message Reception

F(R) Message reception is shown in Figure 2.2-8. At this point the terminal has processed the received Parameters/Certificate Message for the chosen Operational Mode and has determined that the Operational Mode and its parameters, and the Certificate, are acceptable.

The terminal may begin processing the far end's F(R), for the chosen Operational Mode and Keypset, when it is received. This is discussed in Section 2.2.4.3.1. If a terminal receives the far end's F(R) before it has transmitted its own, it may begin processing the received F(R) in parallel with generating its own so long as this does not delay transmission of its own F(R). Note that multiple F(R) Messages may have been sent, but only one of them should have the chosen Operational Mode and Keypset.

Under exceptional conditions the terminal may receive another Parameters/Certificate Message at this point. Processing in this exception case is described in Section 2.2.4.3.2.

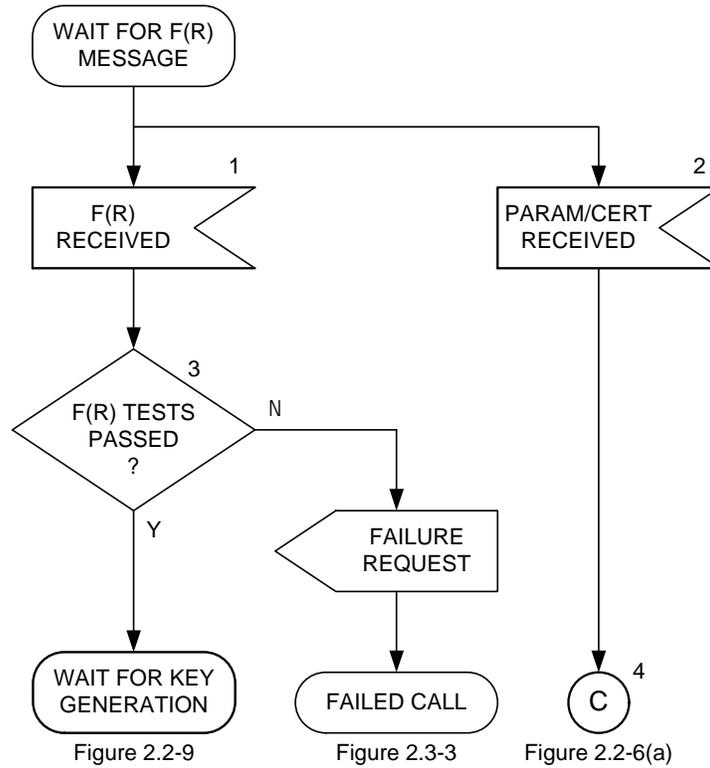
2.2.4.3.1 F(R) Message Received

Upon receipt of the F(R), the F(R) verification tests specified in SCIP-230, Section 2.1.1.4.3, SCIP-231, Section 2.1.5, or SCIP-232, Appendix F.3 are now performed. If the received F(R) fails any of these tests, the terminal shall execute Failed Call processing. If the tests pass, key generation is initiated, and signaling continues as defined in Section 2.2.5. The generation of the traffic key from the Certificate and the F(R) is defined in SCIP-230 or SCIP-232, Section 2.1.1.7; or SCIP-231, Section 2.1.6.

2.2.4.3.2 Parameters/Certificate Message Received

If a Parameters/Certificate Message is received, this indicates that the far end has determined that there are no compatible parameters, there is a security incompatibility, or there is an Access Control failure for the previously chosen Operational Mode, and is attempting to proceed using an alternate Operational Mode. In this case, the incoming Parameters/Certificate Message is processed as specified in Section 2.2.3.3.

2007



NOTES:

1. This path includes the case where the F(R) Message for the chosen Operational Mode was received prior to entering the Wait for F(R) Message state.
2. This path includes the case where the Parameters/Certificate Message was received prior to entering the Wait for F(R) Message state.
3. These are the F(R) verification tests defined in SCIP-23x.
4. Process incoming Parameters/Certificate Message.

2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018

Figure 2.2-8 F(R) Message Reception

2.2.5 Cryptosync Exchange

The third step in SCIP Call Setup is the exchange of Cryptosync Messages. Application IVs are exchanged together with encrypted data that permits the receiver to verify the negotiated parameters, the session key, and that encryption and decryption are operating properly.

2019
2020
2021
2022
2023
2024
2025
2026

2.2.5.1 Cryptosync Message Definition

The format of the Cryptosync Message is shown in Table 2.2-6.

Table 2.2-6 Cryptosync Message - General Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								1
0-msb	0	0	0	0	0	0	0	
Source ID								2
0	0	0	0	1	0	0	0-lsb	
Message Length								3
X-msb	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X-lsb	4
Message Version								5
0	0	0	0	0	0	0	0	
IV Length								6
X-msb	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X-lsb	7
Application IV								8
X-msb	X	X	X	X	X	X	X	
...								7+N
X	X	X	X	X	X	X	X-lsb	
Packet Length								8+N
X-msb	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X-lsb	9+N
Encrypted Packet (optional)								10+N
X-msb	X	X	X	X	X	X	X	
...								9+N+M
X	X	X	X	X	X	X	X-lsb	

2027 N = Length of Application IV. M = Length of Encrypted Packet.

2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072

- For the Cryptosync Message the value of the MID is 0x0008.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the Cryptosync Message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The IV Length shall contain the length of the Application IV field in octets (plus the length of the IV Length field itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 6 and the low order bit being bit 1 of octet 7.
- The Application IV shall contain the IV to be used with the application that has been negotiated. Details of the length, format, and content are found in SCIP-230, Section 3.5, SCIP-231, Section 3.3, or SCIP-232, Section 3.6. The msb of the IV (as defined in SCIP-23x) is placed in bit 8 of octet 8.
- The Packet Length shall contain the length of the Encrypted Packet in octets (plus the length of the Packet Length field itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.
- Inclusion of the Encrypted Packet is mandatory when the Cryptosync Message is used as part of SCIP call setup and Mode Change. The msb of the Encrypted Packet (as defined in SCIP-23x) is placed in Bit 8 of the first octet of the Encrypted Packet field. The length, the encryption algorithm and mode to be used, and the content and format of the plaintext data to be encrypted are defined in SCIP-230, Section 3.4, SCIP-231, Section 3.2, or SCIP-232, Section 3.5.
- The Encrypted Packet is not included when the Message is used for Two-Way Resync (Section 2.3.4).

2.2.5.2 Cryptosync Message Transmission

Cryptosync Message transmission during SCIP call setup is shown in Figure 2.2-9.

When the Traffic Encryption Key (TEK) has been generated or if a PPK Keyset is chosen, the terminal shall format the data to be verified as defined in SCIP-230, Section 3.4, SCIP-231, Section 3.2, or SCIP-232, Section 3.5. This data shall be encrypted (using a cryptographic algorithm and mode defined in SCIP-23x).

If a CKL exists locally and the local CKL version is later than the CKL version in the received Capabilities Message (see SCIP-230 or SCIP-232, Sections 2.1.2.1.2.1 and 2.1.2.3), the terminal shall wait until it receives a Cryptosync Message from the far end. Otherwise, the terminal shall transmit a Cryptosync Message, formatted as defined in Section 2.2.5.1, to the far end and wait until it receives a Cryptosync Message from the far end. In either case, signaling proceeds as defined in Section 2.2.5.3.

2073

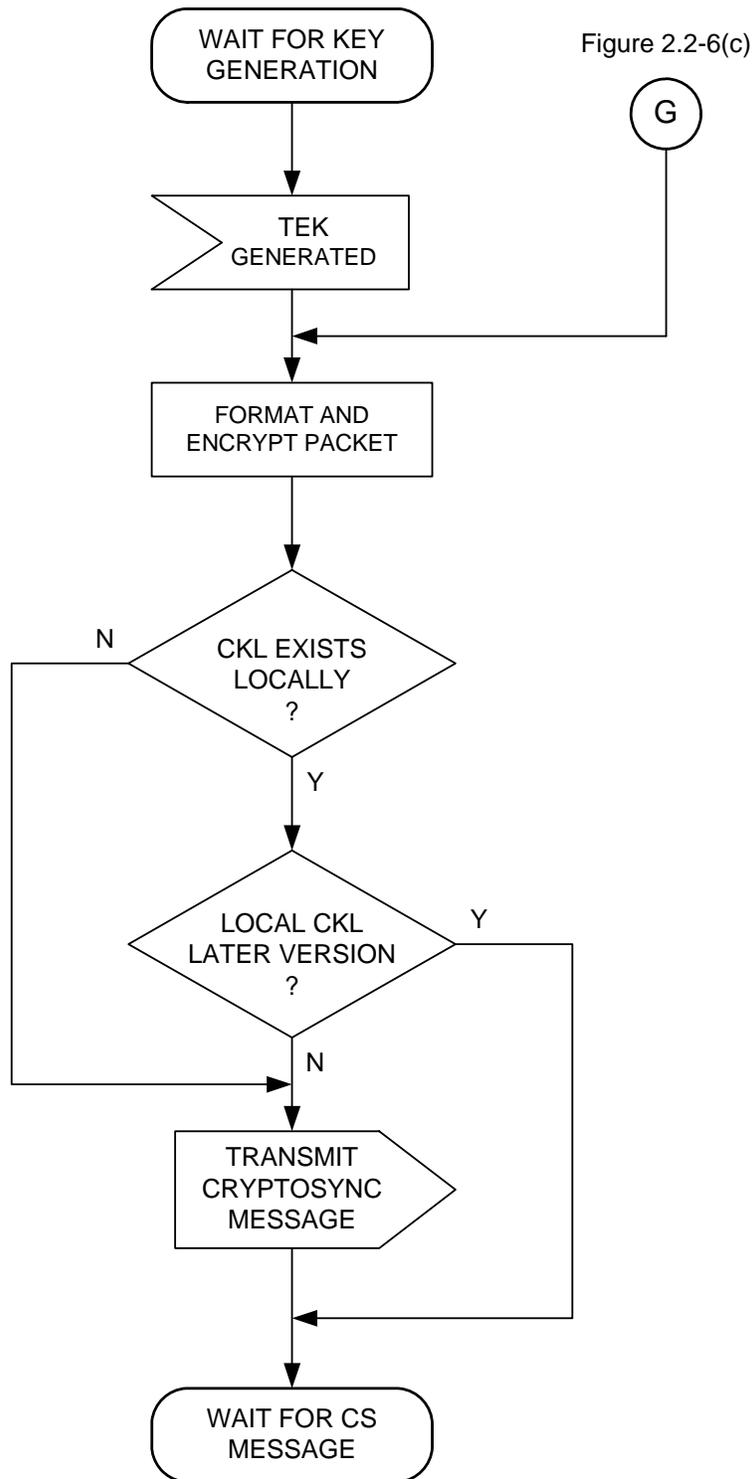


Figure 2.2-10

Figure 2.2-9 Cryptosync Message Transmission

2074
2075
2076

2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106

2.2.5.3 Cryptosync Message Reception

Cryptosync Message reception during SCIP call setup is shown in Figure 2.2-10.

The terminal will process the far end's Cryptosync Message when it is received. This is discussed in Section 2.2.5.3.1. If a terminal receives the far end's Cryptosync Message before it has transmitted its own, it may begin processing the received Cryptosync Message in parallel with generating its own.

Under exceptional conditions the terminal may receive another Parameters/Certificate Message at this point. Processing in this exception case is described in Section 2.2.5.3.2.

2.2.5.3.1 Cryptosync Message Received

If a CKL exists locally and the local CKL version is later than the CKL version in the received Capabilities Message, one or more CKL Transfers shall be performed, as specified in Section 2.3.2.4, to transmit the local CKL to the far end. The terminal shall then transmit a Cryptosync Message, formatted as defined in Section 2.2.5.1, to the far end. Once transmission of the Cryptosync Message is complete, processing continues as follows.

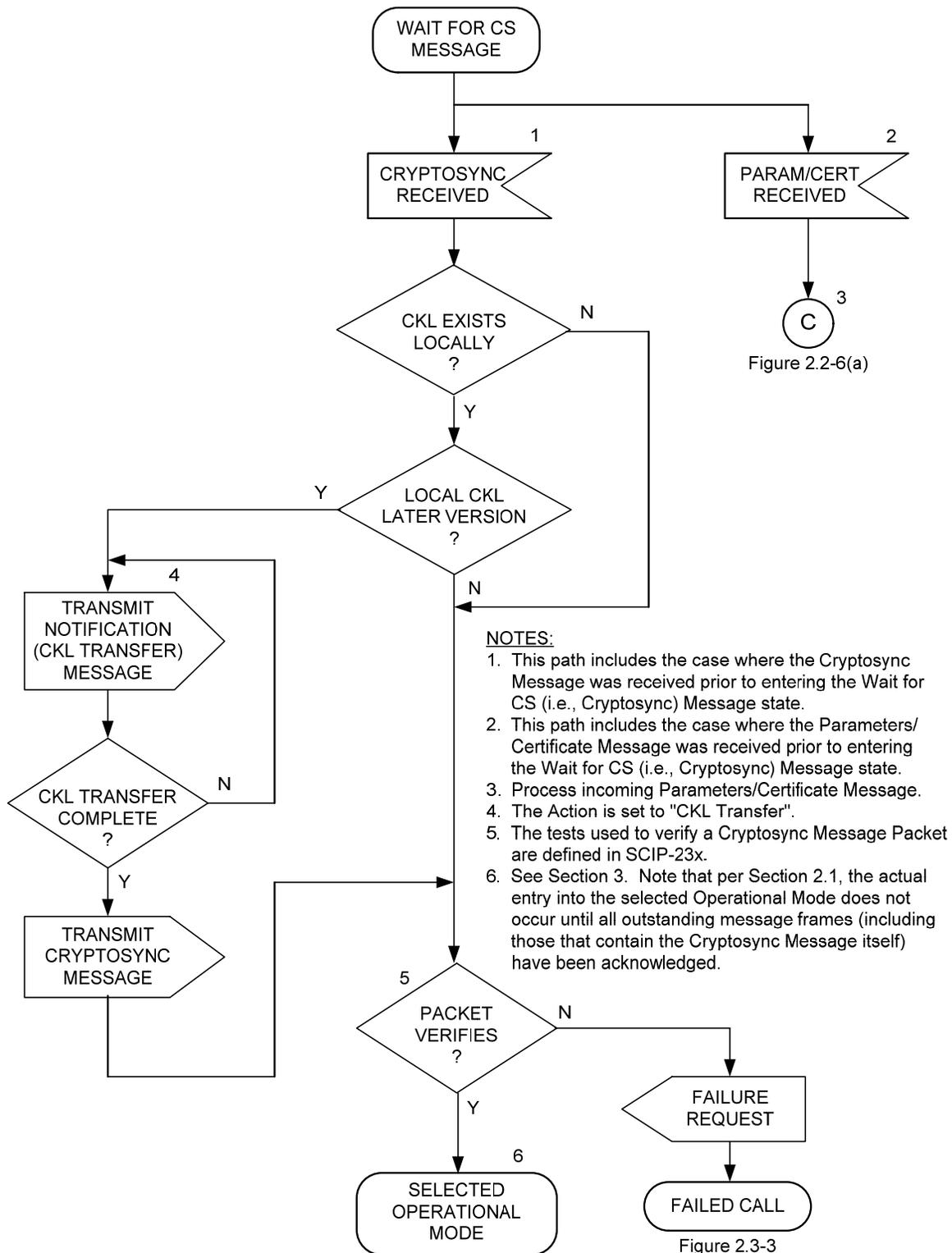
The terminal shall verify the Encrypted Packet contained in the Cryptosync Message as specified in SCIP-230, Section 3.4.1, SCIP-231, Section 3.2.1, or SCIP-232, Section 3.5.1. If this check is not passed, the terminal shall execute Failed Call processing, defined in Section 2.3.2.3.1, with an Information Code of *sync message verification failure*.

For standard secure Operational Modes, the terminal shall then initiate the chosen application, using the exchanged Application IVs, as specified in Section 3.2.

Editor's Note: Note that as defined in Section 3.2, the application is not actually entered until all outstanding framed messages are acknowledged. In particular, the application is not entered until all frames of the Cryptosync Message have been acknowledged.

2107

2108



2109
2110
2111

Figure 2.2-10 Cryptosync Message Reception

2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154

2.2.5.3.2 Parameters/Certificate Message Received

If a Parameters/Certificate Message is received, this indicates that the far end has determined that there are no compatible parameters, there is a security incompatibility, or there is an Access Control failure for the previously chosen Operational Mode, and is attempting to proceed using an alternate Operational Mode. In this case, the incoming Parameters/Certificate Message is processed as specified in Section 2.2.3.3.

2.2.6 Operational Mode and Keypset Type Specific Instantiations

This section defines the Operational Mode and Keypset Type specific use of fields in SCIP call setup and call control messages.

A conservative approach has been taken when determining what is generic to all standard messages and what is Operational Mode or Keypset Type specific. In general, a field or value is considered to be generic if it does not vary for the currently anticipated standard secure Operational Modes and Keypset Types.

Section 2.2.6.1 discusses Key Agreement specific fields and values, Section 2.2.6.2 discusses Secure Voice specific fields and values, Section 2.2.6.3 discusses Secure Data specific fields and values, Section 2.2.6.4 discusses Secure Electronic Rekey specific fields and values, and Section 2.2.6.5 discusses Clear MELP Voice specific fields and values.

2.2.6.1 Key Agreement Specifics

This section provides detailed information for setting Key Agreement specific message fields in SCIP call setup messages.

2.2.6.1.1 Capabilities and Extended Keypsets List Messages

2.2.6.1.1.1 Type 1 FIREFLY Without CSE

In the Capabilities and Extended Keypsets List Messages, a Type 1 Basic FIREFLY w/o CSE Entry in the keysets list is recognized by a value of 0x0001 in the Keypset Type field, and a Type 1 Enhanced FIREFLY w/o CSE Entry in the keysets list is recognized by a value of 0x0002 in the Keypset Type field. For both of these, each corresponding Keypset Parameters Entry has the format defined in Table 2.2-7(a).

2155
2156
2157
2158

Table 2.2-7(a) Keypad Parameters Entry – Type 1 Basic and Enhanced FF w/o CSE Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Keypad ID								
X	X	X	X		X	X	X	1
Nibble 1			Nibble 2					
X	X	X	X		X	X	X	2
Nibble 3			Nibble 4					
Universal ID								
X	X	X	X		X	X	X	3
Nibble 5			Nibble 6					
Universal Edition								
CKL Version								
X	X	X	X	X	X	X	X	4

2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181

- The Keypad ID is treated as 6 nibbles. Each nibble contains an unsigned number from 0x0 to 0x9 with the high order bit of the number in bit 8 or 4 of the octet and the low order bit of the number in bit 5 or 1 of the octet. The upper 4 nibbles shall contain the Universal ID with the first digit of the Universal ID in Nibble 1 and the last digit of the Universal ID in Nibble 4. The lower 2 nibbles shall contain the Universal Edition with the first digit of the Edition in Nibble 5 and the second digit of the Edition in Nibble 6.
- The CKL Version shall be treated as an 8 bit unsigned number with the high order bit of the Version number in bit 8 of the octet and the low order bit of the Version number in bit 1 of the octet. The CKL Version shall be set to 0x00 if no CKL is resident locally in the terminal. See SCIP-230, Section A.2 for additional details pertaining to the CKL Version.

2.2.6.1.1.2 Type 1 FIREFLY With CSE

In the Capabilities and Extended Keypads List Messages, a Type 1 Basic FIREFLY w/CSE Entry in the keypads list is recognized by a value of 0x0004 in the Keypad Type field, and a Type 1 Enhanced FIREFLY w/CSE Entry in the keypads list is recognized by a value of 0x0007 in the Keypad Type field. For both of these, each corresponding Keypad Parameters Entry has the format defined in Table 2.2-7(b).

2182
2183
2184

Table 2.2-7(b) Keypset Parameters Entry – Type 1 Basic and Enhanced FF w/CSE Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Keypset ID								
X	X	X	X		X	X	X	1
Nibble 1			Nibble 2					
X	X	X	X		X	X	X	2
Nibble 3			Nibble 4					
Universal ID								
X	X	X	X		X	X	X	3
Nibble 5			Nibble 6					
Universal Edition								
CSE SPI								
X	X	X	X	X	X	X	X	4
X	X	X	X	X	X	X	X	5
X	X	X	X	X	X	X	X	6
X	X	X	X	X	X	X	X	7
CKL Version								
X	X	X	X	X	X	X	X	8

2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202

- The Keypset ID is treated as 6 nibbles. Each nibble contains an unsigned number from 0x0 to 0x9 with the high order bit of the number in bit 8 or 4 of the octet and the low order bit of the number in bit 5 or 1 of the octet. The upper 4 nibbles shall contain the Universal ID with the first digit of the Universal ID in Nibble 1 and the last digit of the Universal ID in Nibble 4. Nibbles 5 and 6 shall contain the Universal Edition with the first digit of the Universal Edition in Nibble 5 and the second digit of the Universal Edition in Nibble 6.
- The CSE Security Parameters Index (SPI) is a 32-bit value defined in SCIP-230, Section 2.1.1.3.1.2. Its most significant bit, as defined in SCIP-230, Section 2.1.1.3.1.2.2, shall be placed in bit 8 of octet 4, and its least significant bit shall be placed in bit 1 of octet 7.
- The CKL Version shall be treated as an 8 bit unsigned number with the high order bit of the Version number in bit 8 of the octet and the low order bit of the Version number in bit 1 of the octet. The CKL Version shall be set to 0x00 if no CKL is resident locally in the terminal. See SCIP-230, Section A.2 for additional details pertaining to the CKL Version.

2203
2204
2205
2206
2207
2208
2209
2210
2211
2212

2.2.6.1.1.3 Type 1 U.S. Generic PPK Without CSE

In the Capabilities and Extended Keysets List Messages, a Type 1 U.S. Generic PPK w/o CSE Entry in the keysets list is recognized by a value of 0x0008 in the Keyset Type field. The corresponding Keyset Parameters Entry has the format defined in Table 2.2-7(c).

Table 2.2-7(c) Keyset Parameters Entry – Type 1 U.S. Generic PPK w/o CSE Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Keyset ID								
X	X	X	X	X	X	X	X	1
X	X	X	X	X	X	X	X	2
X	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X	4
PPK SPI								

2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228

- The Keyset ID is the PPK Security Parameters Index (SPI), a 32-bit value defined in SCIP-230, Section 2.1.1.2.2. Its most significant bit, as defined in SCIP-230, Section 2.1.1.2.2.3, shall be placed in bit 8 of octet 1, and its least significant bit shall be placed in bit 1 of octet 4.

2.2.6.1.1.4 ECMQV/AES Without CSE – Phase 1

In the Capabilities and Extended Keysets List Messages, an ECMQV/AES w/o CSE – Phase 1 Entry in the keysets list is recognized by a value of 0x0009 in the Keyset Type field. The corresponding Keyset Parameters Entry has the format defined in Table 2.2-7(d).

Table 2.2-7(d) Keyset Parameters Entry – ECMQV/AES w/o CSE – Phase 1 Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Keyset ID								
X	X	X	X	X	X	X	X	1

2229
2230
2231
2232
2233

- The Keyset ID is an 8-bit value defined in SCIP-231, Section 2.1.1.2. Its most significant bit, as defined in SCIP-231, Section 2.1.1.2, shall be placed in bit 8 of octet 1, and its least significant bit shall be placed in bit 1 of octet 1.

2234
 2235
 2236
 2237
 2238
 2239
 2240
 2241
 2242
 2243
 2244
 2245
 2246
 2247
 2248
 2249
 2250
 2251
 2252
 2253
 2254
 2255
 2256
 2257
 2258

2.2.6.1.1.5 ECMQV/AES With CSE – Phase 1

In the Capabilities and Extended Keysets List Messages, an ECMQV/AES w/CSE – Phase 1 Entry in the keysets list is recognized by a value of 0x000A in the Keyset Type field. The corresponding Keyset Parameters Entry has the format defined in Table 2.2-7(e).

Table 2.2-7(e) Keyset Parameters Entry – ECMQV/AES w/CSE – Phase 1 Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	
Keyset ID								← Bits
X	X	X	X	X	X	X	X	Octets ↓
CSE SPI								
X	X	X	X	X	X	X	X	1
X	X	X	X	X	X	X	X	2
X	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X	4
X	X	X	X	X	X	X	X	5

- The Keyset ID is an 8-bit value defined in SCIP-231, Section 2.1.1.2. Its most significant bit, as defined in SCIP-231, Section 2.1.1.2, shall be placed in bit 8 of octet 1, and its least significant bit shall be placed in bit 1 of octet 1.
- The CSE Security Parameters Index (SPI) is a 32-bit value defined in SCIP-231, Section 2.1.2.1.1.2. Its most significant bit, as defined in SCIP-231, shall be placed in bit 8 of octet 4, and its least significant bit shall be placed in bit 1 of octet 7.

2.2.6.1.1.6 NATO ECMQV/AES Without CSE

In the Capabilities and Extended Keysets List Messages, a NATO ECMQV/AES w/o CSE Entry in the keysets list is recognized by a value of 0x000B in the Keyset Type field. The corresponding Keyset Parameters Entry has the format defined in Table 2.2-7(f).

2259
2260
2261

Table 2.2-7(f) Keypset Parameters Entry – NATO ECMQV/AES w/o CSE Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
X	X	X	X		X	X	X	1
	Nibble 1				Nibble 2			
X	X	X	X		X	X	X	2
	Nibble 3				Nibble 4			
	Universal ID							
X	X	X	X		X	X	X	3
	Nibble 5				Nibble 6			
	Universal Edition							
X	X	X	X	X	X	X	X	4
	CKL Version							

2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282

- The Keypset ID is treated as 6 nibbles. Each nibble contains an unsigned number from 0x0 to 0x9 with the high order bit of the number in bit 8 or 4 of the octet and the low order bit of the number in bit 5 or 1 of the octet. The upper 4 nibbles shall contain the Universal ID with the first digit of the Universal ID in Nibble 1 and the last digit of the Universal ID in Nibble 4. The lower 2 nibbles shall contain the Universal Edition with the first digit of the Edition in Nibble 5 and the second digit of the Edition in Nibble 6.
- The CKL Version shall be treated as an 8 bit unsigned number with the high order bit of the Version number in bit 8 of the octet and the low order bit of the Version number in bit 1 of the octet. The CKL Version shall be set to 0x00 if no CKL is resident locally in the terminal. See SCIP-232, Section E.3 for additional details pertaining to the CKL Version.

2.2.6.1.1.7 NATO ECMQV/AES With CSE

In the Capabilities and Extended Keypsets List Messages, a NATO ECMQV/AES w/CSE Entry in the keysets list is recognized by a value of 0x000C in the Keypset Type field. The corresponding Keypset Parameters Entry has the format defined in Table 2.2-7(g).

2283
2284
2285

Table 2.2-7(g) Keypset Parameters Entry – NATO ECMQV/AES w/CSE Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
X	X	X	X		X	X	X	1
Nibble 1			Nibble 2					
X	X	X	X		X	X	X	2
Nibble 3			Nibble 4					
Universal ID								
X	X	X	X		X	X	X	3
Nibble 5			Nibble 6					
Universal Edition								
CSE SPI								
X	X	X	X	X	X	X	X	4
X	X	X	X	X	X	X	X	5
X	X	X	X	X	X	X	X	6
X	X	X	X	X	X	X	X	7
CKL Version								
X	X	X	X	X	X	X	X	8

2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303

- The Keypset ID is treated as 6 nibbles. Each nibble contains an unsigned number from 0x0 to 0x9 with the high order bit of the number in bit 8 or 4 of the octet and the low order bit of the number in bit 5 or 1 of the octet. The upper 4 nibbles shall contain the Universal ID with the first digit of the Universal ID in Nibble 1 and the last digit of the Universal ID in Nibble 4. Nibbles 5 and 6 shall contain the Universal Edition with the first digit of the Universal Edition in Nibble 5 and the second digit of the Universal Edition in Nibble 6.
- The CSE Security Parameters Index (SPI) is a 32-bit value defined in SCIP-232, Section 2.1.1.3.1.2. Its most significant bit, as defined in SCIP-232, Section 2.1.1.3.1.2.2, shall be placed in bit 8 of octet 4, and its least significant bit shall be placed in bit 1 of octet 7.
- The CKL Version shall be treated as an 8 bit unsigned number with the high order bit of the Version number in bit 8 of the octet and the low order bit of the Version number in bit 1 of the octet. The CKL Version shall be set to 0x00 if no CKL is resident locally in the terminal. See SCIP-232, Section E.3 for additional details pertaining to the CKL Version.

2304
2305
2306
2307
2308
2309
2310
2311
2312
2313

2.2.6.1.1.8 NATO PPK/AES Without CSE

In the Capabilities and Extended Keysets List Messages, a NATO PPK/AES w/o CSE Entry in the keysets list is recognized by a value of 0x000D in the Keyset Type field. The corresponding Keyset Parameters Entry has the format defined in Table 2.2-7(h).

Table 2.2-7(h) Keyset Parameters Entry – NATO PPK/AES w/o CSE Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Keyset ID								
X	X	X	X	X	X	X	X	1
X	X	X	X	X	X	X	X	2
X	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X	4
PPK SPI								

2314
2315
2316
2317
2318
2319
2320

- The Keyset ID is the PPK Security Parameters Index (SPI), a 32-bit value defined in SCIP-232, Section 2.1.1.2.2. Its most significant bit, as defined in SCIP-232, Section 2.1.1.2.2.3, shall be placed in bit 8 of octet 1, and its least significant bit shall be placed in bit 1 of octet 4.

2.2.6.1.1.9 Extended Keysets List Support

In the Capabilities and Extended Keysets List Messages, an Extended Keysets List Support Entry in the keysets list is recognized by a value of 0x07FF in the Keyset Type field. The terminal shall include an Extended Keysets List Support Entry as the last keysets list entry in the keysets list, if the terminal supports the ability to receive and optionally transmit the Extended Keysets List Message. Note that this Keyset Type is listed even if the terminal can send all of its keysets in the Capabilities Message without surpassing the message length limitation specified in Section 2.2.1.4. The Additional Keysets parameter of the Extended Keysets List Support Keyset Type indicates if additional keysets actually need to be sent. The corresponding Keyset Parameters Entry has the format defined in Table 2.2-7(i).

Table 2.2-7(i) Keyset Parameters Entry – Extended Keysets List Support Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Additional Keysets								
0	0	0	0	0	0	0	X	1

2336
2337

- The Additional Keysets parameter shall be set to 0x01 if the terminal has additional keysets to offer in an Extended Keysets List Message. The Additional Keysets parameter shall be set to 0x00 if the terminal does not have additional keysets to offer in an Extended Keysets List Message.

2.2.6.1.2 Parameters/Certificate Message

In the Parameters/Certificate Message, a Certificate has the format defined in Table 2.2-8.

Table 2.2-8 Certificate Field Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Certificate								
X	X	X	X	X	X	X	X	1
...								
X	X	X	X	X	X	X	X	K

K = Length of Certificate.

2.2.6.1.2.1 Type 1 FIREFLY

In the Parameters/Certificate Message, a Type 1 FIREFLY Keyset ID has the same format as shown for the Capabilities Message. If the negotiated Keyset Type is Basic FF, the terminal shall process a received Certificate using the Basic FF rules. If the negotiated Keyset Type is Enhanced FF, the terminal shall process a received Certificate using the Enhanced FF rules. Processing rules for both Keyset Types are specified in SCIP-230, Section 2.1.1.4.

The Certificate field shall contain CC1/CC2 for the negotiated Keyset. CC1 shall precede CC2. The most significant bit of CC1 (as defined in the arithmetic calculation) shall be placed in bit 8 of the first octet, and the least significant bit of CC2 shall be placed in bit 1 of the last octet.

2.2.6.1.2.2 Type 1 U.S. Generic PPK

In the Parameters/Certificate Message, a Type 1 U.S. Generic PPK Keyset ID has the same format as shown for the Capabilities Message. The Certificate Length field has a value of 0x0002, and the Certificate field is not present in the message.

2.2.6.1.2.3 ECMQV/AES – Phase 1

In the Parameters/Certificate Message, the ECMQV/AES – Phase 1 Keyset ID has the same format as shown for the Capabilities Message. If the negotiated Keyset Type is ECMQV/AES –

2378 Phase 1, the terminal shall process a received Certificate using the ECMQV rules specified in
2379 SCIP-231, Section 2.1.3.3.

2380
2381 The Certificate field shall contain the ASN.1/DER encoded Certificate contents defined in SCIP-
2382 231, Section 2.1.3.3.1. The most significant bit of the ASN.1/DER encoded initial SEQUENCE
2383 (see SCIP-231, Appendix A) shall be placed in bit 8 of the first octet, and the least significant bit
2384 of the ASN.1/DER encoded Signature Value at the end of the final SEQUENCE shall be placed
2385 in bit 1 of the last octet.

2386
2387
2388 **2.2.6.1.2.4 NATO ECMQV/AES**

2389
2390 In the Parameters/Certificate Message, a NATO ECMQV/AES Keyset ID has the same format as
2391 shown for the Capabilities Message. Rules for processing a received Certificate are specified in
2392 SCIP-232, Section 2.1.1.4.

2393
2394 The Certificate field shall contain CC1/CC2 for the negotiated Keyset. CC1 shall precede CC2.
2395 The most significant bit of CC1 (as defined in the arithmetic calculation) shall be placed in bit 8
2396 of the first octet, and the least significant bit of CC2 shall be placed in bit 1 of the last octet.

2397
2398
2399 **2.2.6.1.2.5 NATO PPK/AES**

2400
2401 In the Parameters/Certificate Message, a NATO PPK/AES Keyset ID has the same format as
2402 shown for the Capabilities Message. The Certificate Length field has a value of 0x0002, and the
2403 Certificate field is not present in the message.

2404
2405
2406 **2.2.6.1.3 F(R) Message**

2407
2408 In the F(R) Message, the F(R) field has the format defined in Table 2.2-9.

2409
2410
2411 **Table 2.2-9 F(R) Field Format**

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets
X	X	X	X	F(R) ...	X	X	X	X
X	X	X	X	X	X	X	X	L

2413 L = Length of F(R).

2414

2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461

2.2.6.1.3.1 Type 1 FIREFLY

In the F(R) Message, a Type 1 FIREFLY Keyset ID has the same format as shown for the Capabilities Message, and a FIREFLY F(R), calculated as defined in SCIP-230, Section 2.1.1.6, is included in the F(R) field. If the Keyset Type of the negotiated Keyset is Basic FF, the field shall contain a Basic FF F(R). If the Keyset Type of the negotiated Keyset is Enhanced FF, the field shall contain an Enhanced FF F(R).

The F(R) field shall contain either a Basic FF F(R) or an Enhanced FF F(R) for the Universal Edition negotiated. In terms of SCIP signaling, the only difference is the length of the field. The F(R)'s most significant bit (as defined in SCIP-230, Section 2.1.1.6) shall be placed in bit 8 of the first octet, and its least significant bit shall be placed in bit 1 of the L'th octet.

2.2.6.1.3.2 Type 1 U.S. Generic PPK

The F(R) Message does not apply to the Type 1 U.S. Generic PPK Keyset Type.

2.2.6.1.3.3 ECMQV/AES – Phase 1

In the F(R) Message, the ECMQV/AES – Phase 1 Keyset ID has the same format as shown for the Capabilities Message, and an ECMQV F(R), calculated as defined in SCIP-231, Section 2.1.4, is included in the F(R) field.

The F(R) field shall contain the ECMQV F(R) and a Nonce. The most significant bit of the first octet of the F(R) (as defined in SCIP-231, Section 2.1.4.3) shall be placed in bit 8 of the first octet, and the least significant bit of the Nonce shall be placed in bit 1 of the L'th octet.

2.2.6.1.3.4 NATO ECMQV/AES

In the F(R) Message, a NATO ECMQV/AES Keyset ID has the same format as shown for the Capabilities Message, and an ECMQV F(R), calculated as defined in SCIP-232, Section 2.1.1.6, is included in the F(R) field.

The F(R) field shall contain an ECMQV F(R) for the Universal Edition negotiated. In terms of SCIP signaling, the only difference is the length of the field. The ECMQV F(R)'s most significant bit (as defined in SCIP-232, Section 2.1.1.6) shall be placed in bit 8 of the first octet, and its least significant bit shall be placed in bit 1 of the L'th octet.

2.2.6.1.3.5 NATO PPK/AES

The F(R) Message does not apply to the NATO PPK/AES Keyset Type.

2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475

2.2.6.2 Secure Voice Specifics

Secure Voice is chosen by negotiating Operational Mode 0x0001.

The only Operational Mode specific field is the Operational Mode Parameters for Secure Voice in the Parameters/Certificate Message. This field has three subfields: Security Levels, Secure Voice Options List Length, and a Secure Voice Options List. The format of this field for Secure Voice is given in Table 2.2-10. Note that there may be multiple Secure Voice Options within this field.

Table 2.2-10 Operational Mode Parameters – Secure Voice

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Security Levels								1
X	X	X	X	X	X	X	X	
Max				Min				
Secure Voice Options List Length								2
X-msb	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X-lsb	3
Secure Voice Options List								4
X-msb	X	X	X	X	X	X	X	
Source ID								
X	X	X	X	X	X	X	X-lsb	5
First Secure Voice Option ID								
...								
X-msb	X	X	X	X	X	X	X	4+2L-2
Source ID								
X	X	X	X	X	X	X	X-lsb	5+2L-2
L'th Secure Voice Option ID								

L = Number of Secure Voice Option Entries.

2476
2477
2478
2479
2480
2481
2482
2483
2484
2485

- The Security Levels field defines a range of security levels compatible with the Operational Mode. The upper nibble in octet 1 shall identify the Maximum Security Level, and the lower nibble shall identify the Minimum Security Level for that combination. The nibble values and corresponding Security Levels are defined in Table 2.2-11. If a Type 1 Keyset is negotiated, only interoperable security levels in the Type 1 Keyset ID Family shall be offered. If a Non-Type 1 Keyset is negotiated, only interoperable security levels in the Non-Type 1 Keyset ID Family shall be offered.

2486
2487
2488

Table 2.2-11 Interoperable Security Levels

Nibble Values	Definition	Keypset ID Family
0xF	reserved	Non-Type 1
0xE	reserved	
0xD	reserved	
0xC	reserved	
0xB	Protected	
0xA	reserved	
0x9	reserved	
0x8	reserved	
0x7	reserved4	Type 1
0x6	Top Secret	
0x5	Secret	
0x4	Confidential	
0x3	reserved3	
0x2	Restricted	
0x1	Unclassified	
0x0	reserved1	

2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505

- The Secure Voice Operational Mode Parameters field shall contain a Secure Voice Options List Length. This shall contain the actual length, in octets, of the Secure Voice Options List (plus the length of the Secure Voice Options List Length itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 2 and the low order bit being bit 1 of octet 3.
- The Secure Voice Options List shall contain the IDs of the supported options for the chosen Operational Mode. Each ID is 2 octets per option. The format of each ID is as follows. The high order 5 bits of the first octet identify the Source where the Voice Option is defined. Currently identified Sources and their IDs are defined in Section 2.5.1. After the Source ID, the next 11 bits identify a unique Secure Voice Option (see Table 2.2-12). The high order bit of the Option ID is placed in bit 8 of the first octet of the Voice Options List Entry, and the low order bit of the Option ID is placed in bit 1 of the second octet of the Voice Options List Entry. Secure Voice Options are listed in order of preference, and the first option on the Initiator's List that is also supported by the Responder shall be chosen.

2506
2507
2508

Table 2.2-12 Secure Voice Options

Option ID	Option
0x0002	Secure 2400 bps MELP Voice – Blank & Burst (DTX)
0x0003	Secure 2400 bps MELP Voice – Blank & Burst (FCT)
0x0004	Secure MELP Voice – Burst w/o Blank (DTX)
0x0005	Secure MELP Voice – Burst w/o Blank (FCT)
0x0009	Reserved for compatibility with legacy terminals
0x000E	Reserved for Secure G.729F Voice – Burst w/o Blank (DTX)
0x000F	Secure G.729D Voice – Burst w/o Blank (FCT)
0x1800	Secure Advanced Multi-Band Excitation (AMBE) Voice

2509
2510
2511
2512
2513
2514
2515
2516
2517
2518

2.2.6.2.1 Secure MELP and Secure G.729D Voice Options

The Secure MELP and Secure G.729D Voice applications are defined in Section 3.3 of this Signaling Plan. Two options are defined for Secure MELP Voice – Blank and Burst, and Burst w/o Blank. Only one option is defined for Secure G.729D Voice – Burst w/o Blank. Secure G.729F Voice is **TBSL**.

2519
2520
2521
2522
2523

2.2.6.2.2 Secure AMBE Voice Specific Option

Secure Advanced Multi-Band Excitation (AMBE) Voice, as indicated by the Source bits, is a General Dynamics defined Operational Mode.

2524
2525
2526

2.2.6.3 Secure Data Specifics

2527
2528
2529
2530
2531
2532

Two variants of secure data Operational Modes are defined. Secure Data, specified in Section 2.2.6.3.1, is chosen by negotiating Operational Mode 0x0002. Enhanced Secure Data, specified in Section 2.2.6.3.2, is chosen by negotiating Operational Mode 0x0003. The difference between the Operational Modes is that Secure Data has one set of Security Level values that apply to all data options offered, while Enhanced Secure Data has one set of Security Level values for each data option offered.

2533
2534
2535
2536
2537
2538

During call setup, one of the two secure data Operational Modes in the Capabilities Messages is first negotiated and then the associated Operational Mode Parameters in the Parameters/Certificate Messages are negotiated. During Mode Change, the negotiation takes place with the Mode Change Request and Response Messages. It is recommended that if a terminal offers both secure data Operational Modes in the Capabilities Message that Enhanced

2539 Secure Data be offered first. Otherwise, Enhanced Secure Data may never be negotiated since
2540 terminals will offer the SCIP MER data application in Secure Data.

2541
2542 The data options listed in Table 2.2-13 may be used for multiple data applications. For example,
2543 Fax via Secure Reliable Transport Asynchronous Data, Chat via Secure Reliable Transport
2544 Asynchronous Data, etc., may be defined as additional data options in the future. These data
2545 options may be listed in the Operational Mode Parameters associated with the Secure Data,
2546 Enhanced Secure Data, or both Operational Mode(s).

2547
2548
2549 **Table 2.2-13 Secure Data/Enhanced Secure Data Options**

2550

Option ID	Option
0x0002	Secure Best Effort Transport Asynchronous Data without error extension
0x0004	Secure Reliable Transport Asynchronous Data without error extension
0x0005	Secure Reliable Transport Asynchronous Data with error extension

2551
2552 Secure data applications are defined in Section 3.4 of this Signaling Plan. The use of error
2553 extension applies to the cryptography, as defined in SCIP-230, Section 4.1.2, or SCIP-232,
2554 Section 4.2.1, and is transparent to the signaling.

2555
2556
2557 **2.2.6.3.1 Secure Data Operational Mode Parameters**

2558
2559 The Operational Mode Parameters field for Secure Data has three subfields: Security Levels,
2560 Secure Data Options List Length, and a Secure Data Options List. The format of this field is
2561 given in Table 2.2-14. The Secure Data format allows only one Security Level range for all
2562 Option IDs. This limits all offered Secure Data Options to one Security Level range. Note that
2563 there may be multiple Secure Data Options within this field.

2564

2565
2566
2567

Table 2.2-14 Operational Mode Parameters – Secure Data

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓	
Security Levels									
X	X	X	X	X	X	X	X	1	
Max				Min					
Secure Data Options List Length									
X-msb	X	X	X	X	X	X	X	2	
X	X	X	X	X	X	X	X-lsb	3	
Secure Data Options List									
X-msb	X	X	X	X	X	X	X	4	
Source ID									
X	X	X	X	X	X	X	X-lsb	5	
First Secure Data Option ID									
...									
X-msb	X	X	X	X	X	X	X	4+2L-2	
Source ID									
X	X	X	X	X	X	X	X-lsb	5+2L-2	
L'th Secure Data Option ID									

L = Number of Secure Data Option Entries.

2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589

- The Security Levels field defines a range of security levels compatible with the Operational Mode. The upper nibble in octet 1 shall identify the Maximum Security Level, and the lower nibble shall identify the Minimum Security Level for that combination. The nibble values and corresponding Security Levels are defined in Table 2.2-11. If a Type 1 Keyset is negotiated, only interoperable security levels in the Type 1 Keyset ID Family shall be offered. If a Non-Type 1 Keyset is negotiated, only interoperable security levels in the Non-Type 1 Keyset ID Family shall be offered.
- The Secure Data Operational Mode Parameters field shall contain a Secure Data Options List Length. This shall contain the actual length, in octets, of the Secure Data Options List (plus the length of the Secure Data Options List Length itself). The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 2 and the low order bit being bit 1 of octet 3.
- The Secure Data Options List shall contain the IDs of the supported options for the chosen Operational Mode. Each ID is 2 octets per option. The format of each ID is as follows. The high order 5 bits of the first octet identify the Source where the Data Option is defined. Currently identified Sources and their IDs are defined in Section 2.5.1. After the Source ID, the next 11 bits identify a unique Secure Data Option (see Table 2.2-13). The high order bit of the Option ID is placed in bit 8 of the first octet of the Data Options List Entry, and the low order bit of the Option ID is placed in bit

2590 1 of the second octet of the Data Options List Entry. Secure Data Options are listed
2591 in order of preference, and the first option on the Initiator's List that is also supported
2592 by the Responder shall be chosen.

2593
2594

2.2.6.3.2 Enhanced Secure Data Operational Mode Parameters

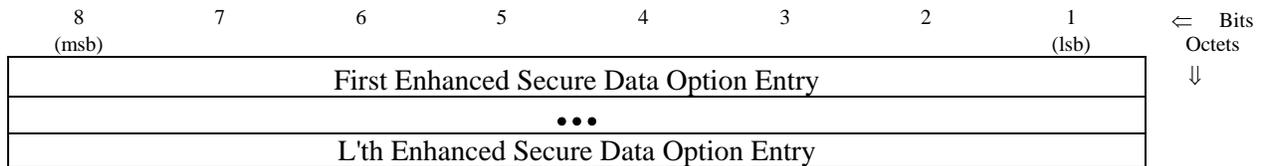
2595
2596
2597
2598
2599
2600
2601
2602

The Operational Mode Parameters field for Enhanced Secure Data, shown in Table 2.2-15(a), has two subfields in each Enhanced Secure Data Option Entry as shown in Table 2.2-15(b): Option ID and Security Level. This added flexibility allows all offered Enhanced Secure Data Options to be at different Security Level ranges, since Enhanced Secure Data Options may have different security requirements. Note that there may be multiple Enhanced Secure Data Options within this field.

2603
2604

Table 2.2-15(a) Operational Mode Parameters – Enhanced Secure Data

2605
2606

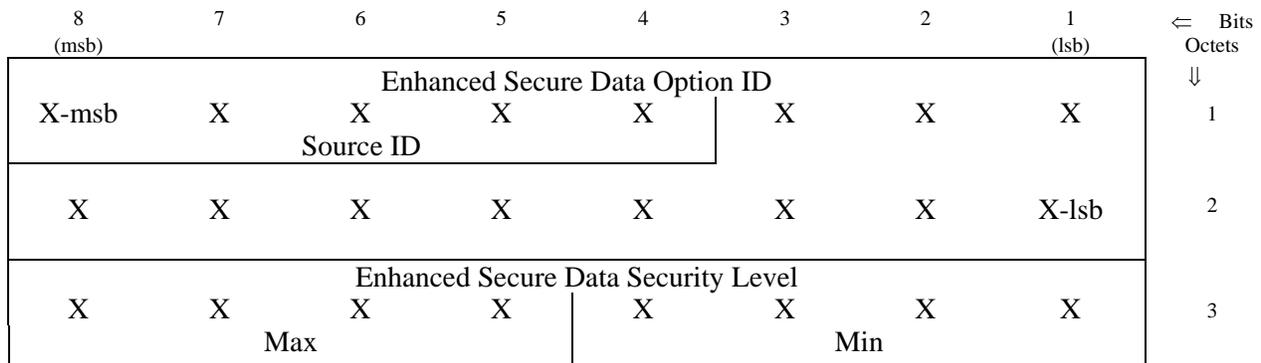


2607 L = Number of Enhanced Secure Data Option Entries.

2608
2609

Table 2.2-15(b) Enhanced Secure Data Option Entry

2610
2611



2612
2613

- The Enhanced Secure Data Option ID field shall contain the ID of the supported option for the chosen Operational Mode. Each ID is 2 octets per option. The format of each ID is as follows. The high order 5 bits of the first octet identify the Source where the Data Option is defined. Currently identified Sources and their IDs are defined in Section 2.5.1. After the Source ID, the next 11 bits identify a unique Enhanced Secure Data Option (see Table 2.2-13). The high order bit of the Option ID is placed in bit 8 of the first octet, and the low order bit of the Option ID is placed in bit 1 of the second octet. Enhanced Secure Data Options are listed in order of

2619
2620

2621 preference, and the first option on the Initiator's List that is also supported by the
2622 Responder shall be chosen.

- 2623 • The Enhanced Secure Data Security Level field defines a range of security levels
2624 compatible with the Enhanced Secure Data Option ID. The upper nibble shall
2625 identify the Maximum Security Level, and the lower nibble shall identify the
2626 Minimum Security Level for that combination. The nibble values and corresponding
2627 Security Levels are defined in Table 2.2-11. If a Type 1 Keyset is negotiated, only
2628 interoperable security levels in the Type 1 Keyset ID Family shall be offered. If a
2629 Non-Type 1 Keyset is negotiated, only interoperable security levels in the Non-Type
2630 1 Keyset ID Family shall be offered.

2633 **2.2.6.4 Secure Electronic Rekey Specifics**

2635 Secure Electronic Rekey is chosen by negotiating Operational Mode 0x000E.

2637 The only Operational Mode specific field is the Operational Mode Parameters for Secure
2638 Electronic Rekey in the Parameters/Certificate Message. This field has three subfields: Security
2639 Levels, Electronic Rekey Options List Length, and an Electronic Rekey Options List. The
2640 format of this field for Electronic Rekey is given in Table 2.2-16.

2643 **Table 2.2-16 Operational Mode Parameters – Secure Electronic Rekey**

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓	
X	X	X	X	X	X	X	X	1	
Max			Security Levels					Min	
X-msb	X	X	X	X	X	X	X	2	
Electronic Rekey Options List Length								3	
X	X	X	X	X	X	X	X-lsb	3	
X-msb	X	X	X	X	X	X	X	4	
Source ID				Electronic Rekey Options List				5	
X	X	X	X	X	X	X	X-lsb	5	
First Electronic Rekey Option ID								...	
X-msb	X	X	X	X	X	X	X	4+2L-2	
Source ID				L'th Electronic Rekey Option ID				5+2L-2	

2645 L = Number of Electronic Rekey Option Entries.

- 2647 • The Security Levels field defines a range of security levels compatible with the
2648 Operational Mode. The Maximum and Minimum Security Levels shall be set as
2649 specified in SCIP-230 or SCIP-232, Section 2.1.3.2. The upper nibble in octet 1 shall
2650 contain the Maximum Security Level, and the lower nibble shall contain the
2651 Minimum Security Level. The nibble values and corresponding Security Levels are
2652 defined in Table 2.2-11. Only interoperable security levels in the Type 1 Keypset ID
2653 Family shall be offered.
- 2654 • The Secure Electronic Rekey Operational Mode Parameters field shall contain an
2655 Electronic Rekey Options List Length. This shall contain the actual length, in octets,
2656 of the Electronic Rekey Options List (plus the length of the Electronic Rekey Options
2657 List Length itself). The value of the field shall be an unsigned binary integer with the
2658 high order bit being bit 8 of octet 2 and the low order bit being bit 1 of octet 3.
- 2659 • The Electronic Rekey Options List shall contain the IDs of the supported options for
2660 Electronic Rekey. Each ID is 2 octets per option. The format of each ID is as
2661 follows. The high order 5 bits of the first octet identify the Source where the Rekey
2662 Option is defined. Currently identified Sources and their IDs are defined in Section
2663 2.5.1. After the Source ID, the next 11 bits identify a unique Electronic Rekey
2664 Option (see Table 2.2-17). The high order bit of the Option ID is placed in bit 8 of
2665 the first octet of the Rekey Options List Entry, and the low order bit of the Option ID
2666 is placed in bit 1 of the second octet of the Rekey Options List Entry. Electronic
2667 Rekey Options are listed in order of preference, and the first option on the Initiator's
2668 List that is also supported by the Responder shall be chosen.

Table 2.2-17 Electronic Rekey Options

Option ID	Option
0x0004	Rekey via secure RT messages w/o error extension, w/o 32-bit CRC
0x0006	Rekey via secure RT messages w/o error extension, with 32-bit CRC

2673
2674
2675 The Electronic Rekey application is defined in Section 4 of this Signaling Plan. The Rekey
2676 APDUs are encrypted then encapsulated in the Rekey Message structure defined in Section 4.2.
2677 Data ordering and encryption are specified in SCIP-230, Section 6.2, or SCIP-232, Appendix
2678 E.2. Error extension is not used. For Rekey Option 0x0006 (with 32-bit CRC), the CRC check
2679 bits are computed prior to encryption as specified in SCIP-230, Section 6.2.1, or SCIP-232,
2680 Appendix E.2.1. The Rekey Messages are transported via the reliable message transport
2681 mechanisms specified in Section 2.1.

2.2.6.5 Clear MELP Voice Specifics

2682
2683
2684 Clear MELP Voice is chosen by negotiating Operational Mode 0x0004 and is defined in Section
2685 3.3.1.3 of this Signaling Plan. There is no Parameters/Certificate Exchange, F(R) Exchange, or
2686 Cryptosync Exchange.
2687
2688

2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733

2.3 SCIP Call Control Signaling

When invoked, either by an internal indication or a user-initiated request, the terminal executes Call Control signaling to perform such functions as terminating a call, changing the application, alerting the far-end terminal, and cryptographic resynchronization. This section specifies the signaling for each of the Call Control functions and the interaction between each Call Control function and the applications active at the time that the Call Control function is executed. Some Call Control functions, such as Connection Terminate, may be executed at any time; during application traffic processing, during Call Setup, or even during Call Control processing. Other Call Control functions, such as Mode Change, are only performed during secure application traffic processing.

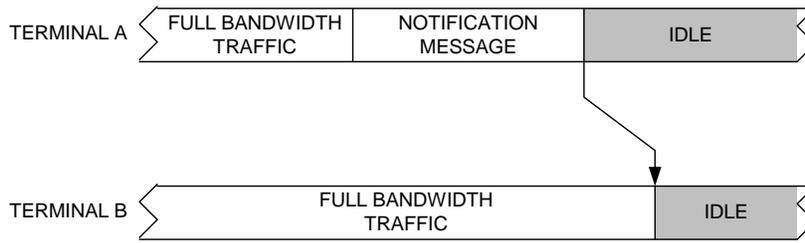
Call Control signaling involves four different messages: Notification, Mode Change Request, Mode Change Response, and Cryptosync. The Notification Message, with the Action set to Connection Terminate, Native Clear Voice, Secure Update, or Connection Idle, has a higher priority and upon receipt shall interrupt Mode Change, Two-Way Resync, CKL Transfer, Secure Dial, or Attention processing. The priority scheme of the Notification Message is specified in Section 2.3.2. The remaining Call Control messages, Mode Change Request, Mode Change Response, Cryptosync, and Notification with the Action set to CKL Transfer, Secure Dial, or Attention, shall be processed on a first come first served basis.

Call Control messages use the same framed transmission/reception format as specified in Section 2.1.

2.3.1 Call Control Timelines

Examples of Call Control signaling time lines are shown in Figures 2.3-1(a), 2.3-1(b), 2.3-1(c), 2.3-1(d), and 2.3-1(e). Call Control Messages are sent as framed traffic and may interrupt full bandwidth traffic. Note that these figures are presented from the Message Layer only, thus ESCAPEs, REPORTs, SOMs, and EOMs are not shown. Refer to Figure 2.1-1(a) for framed and Figure 2.1-1(b) for full bandwidth Transport Layer operations. Processing of Call Control Messages will result in terminals going to either framed or full bandwidth formats. If a terminal is required to enter application traffic, the processing of Section 3 applies. Note that re-entering a full bandwidth application without the benefit of Cryptosync means that the terminals will not transmit FILLER.

Examples of Notification Message signaling time lines are shown in Figures 2.3-1(a), 2.3-1(b), and 2.3-1(c). The examples depicted do not contain any errors, thus require no retransmissions. Figure 2.3-1(a) is the case where transmitting/receiving the Notification Message (specifically for the Actions set to Connection Terminate, Native Clear Voice, Secure Update, or Connection Idle) from full bandwidth traffic results in both terminals going to framed operation.



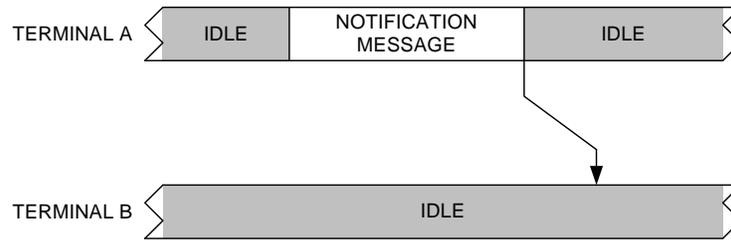
2734
2735

Figure 2.3-1(a) Notification Message Signaling Time Line (Full Bandwidth to Framed)

2737
2738

Figure 2.3-1(b) is the case where transmitting/receiving the Notification Message (for any of the Actions) from framed operation does not cause the terminal to transition from framed operation.

2739
2740
2741
2742



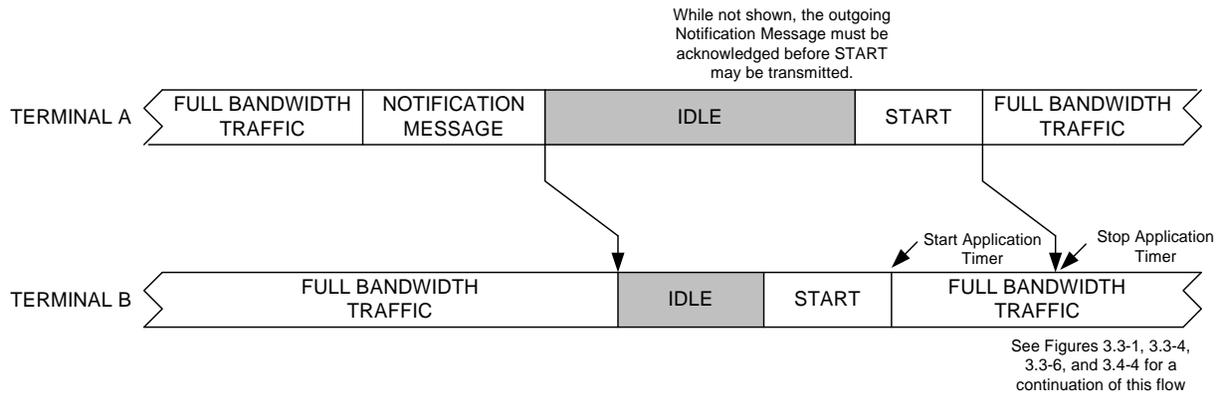
2743
2744

Figure 2.3-1(b) Notification Message Signaling Time Line (Framed to Framed)

2746
2747

Figure 2.3-1(c) is the case where after transmitting/receiving the Notification Message (specifically for the Actions set to CKL Transfer, Secure Dial, or Attention) from full bandwidth traffic and transitioning to framed operation, the terminals are required to return to full bandwidth traffic.

2748
2749
2750
2751
2752
2753

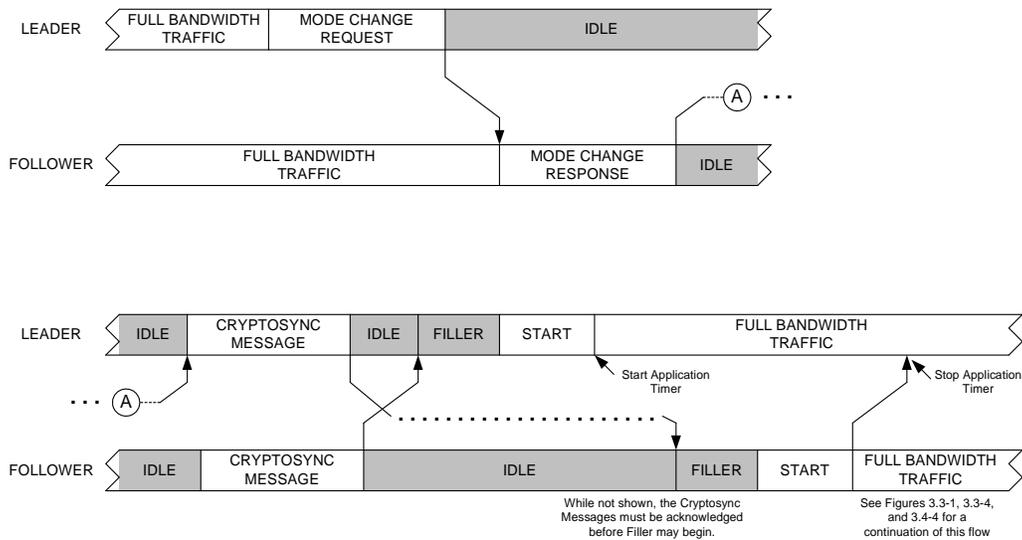


2754
2755

Figure 2.3-1(c) Notification Message Signaling Time Line (Full Bandwidth to Full Bandwidth)

2757
2758
2759

2760 An example Mode Change signaling time line is shown in Figure 2.3-1(d). The case depicted is
2761 from full bandwidth traffic and does not contain any errors, thus requires no retransmissions. A
2762 Cryptosync Exchange follows the Mode Change Request/ Response Exchange and will bring
2763 both terminals back to traffic. For full bandwidth traffic, FILLER and a Start will precede
2764 traffic. In the case of framed traffic, application frames can begin as soon as Cryptosync
2765 Exchange and verification are complete.
2766
2767



2768

2769

2770 **Figure 2.3-1(d) Mode Change Signaling Time Line**

2771

2772

2773

2774

2775

2776

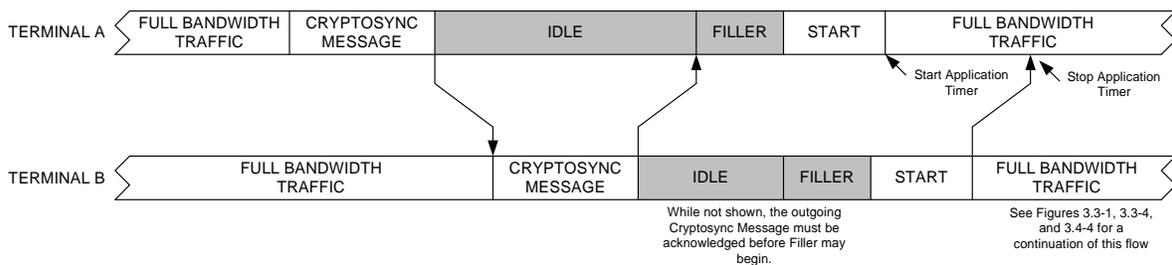
2777

2778

2779

2780

An example Two-Way Resync signaling time line is shown in Figure 2.3-1(e). The case depicted is from full bandwidth traffic and does not contain any errors, thus requires no retransmissions. Following the Cryptosync Exchange both terminals will be brought back to application traffic. For full bandwidth traffic, FILLER and a Start will precede traffic. In the case of framed traffic, application frames can begin as soon as Cryptosync Exchange and verification are complete.



2781

2782

2783 **Figure 2.3-1(e) Two-Way Resync Signaling Time Line**

2784

2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832

2.3.2 Notification Message Processing

This section specifies the processing associated with the Notification Message. The Notification Message serves several functions and has seven associated Actions to perform these functions: Connection Terminate, Native Clear Voice, Secure Update, Connection Idle, CKL Transfer, Secure Dial, and Attention. Notification Messages containing any of these Actions are sent in the clear. All Notification Messages, except for CKL Transfer, Secure Update, and Secure Dial, can be sent at any time during call setup. Additionally, all Notification Messages, except for CKL Transfer, can be sent at any time while a SCIP application is executing. Since sending Secure Dial requires having a key negotiated and verified, it can only be sent after Cryptosync Exchange and verification. A terminal requested to perform one of these functions will generate a local indication for a Notification Message to be formatted and sent to the far end. See Section 2.3.2.1 for the message format.

Section 2.3.2.2 specifies Notification (Connection Terminate) that is used to terminate the data channel. Section 2.3.2.3 specifies Notification (Native Clear Voice/Connection Idle) processing, which allows the terminal to revert to clear voice either when an error occurs or when the user selects “Nonsecure”, or to enter the Connection Idle state if neither Native Clear Voice nor Clear MELP Voice is available. It also allows a terminal to enter the Connection Idle state when the user selects “Secure” (from Clear MELP Voice) or when a terminal executes a Secure Restart. Section 2.3.2.4 specifies Notification (CKL Transfer), which allows a terminal with a later CKL version to transmit it to a terminal with an earlier one. Section 2.3.2.5 specifies Notification (Secure Dial), which allows a terminal to transmit encrypted keypad or other dialing data to the far-end terminal. Section 2.3.2.6 specifies Notification (Attention), which allows a terminal to alert the far-end user by requesting that the far-end terminal perform a vendor elective action (e.g., emitting an audible tone, blinking the display, etc.). Section 2.3.2.7 specifies Notification (Secure Update), which allows terminals executing a secure application to update the current PPK and return to the same secure application using the updated PPK.

SCIP signaling involves three priority levels.

- The transmission and reception of the Notification (Connection Terminate) Message (Section 2.3.2.2) is the highest priority process and shall interrupt every other process.
- The transmission and reception of the Notification Messages related to Failed Call (Section 2.3.2.3.1), user selection of Nonsecure (Section 2.3.2.3.2), user selection of Secure (Section 2.3.2.3.3), Secure Restart (Section 2.3.2.3.4), and Secure Update (Section 2.3.2.7) are the next highest priority processes and shall interrupt all processes of the lowest priority. Note that in these cases, as with Connection Terminate, control is not returned to the interrupted process.
- The processes related to the transmission and reception of all other SCIP call setup and control signaling messages are of lowest priority. Once such a process (e.g., Mode Change) is started, except for the transmission of Notification Messages, the process continues to completion before another process may begin. Lowest priority Notification Messages (i.e., CKL Transfer, Secure Dial, and Attention) may be

2833
2834
2835

transmitted during a “WAIT” state and will return control to that “WAIT” state, at which point the “waiting” process may continue.

Editor’s Note: These priority levels do not apply to the Transport Layer, which is first in - first out.

2836
2837
2838
2839
2840
2841
2842
2843

2.3.2.1 Notification Message Definition

The format of the Notification Message is shown in Table 2.3-1.

Table 2.3-1 Notification Message Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
Source ID								
0	0	0	0	1	1	1	0-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
Action Field								
X-msb	X	X	X	X	X	X	X	6
Source ID								
X	X	X	X	X	X	X	X-lsb	7
Information Length								
X-msb	X	X	X	X	X	X	X	8
X	X	X	X	X	X	X	X-lsb	9
Information Field (Optional)								
X	X	X	X	X	X	X	X	10
•••								
X	X	X	X	X	X	X	X	
First Information Field Entry								
•••								
X	X	X	X	X	X	X	X	
•••								
X	X	X	X	X	X	X	X	9+L
Last Information Field Entry								

2844 L = Length of Information Field.

2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871

- For the Notification Message, the value of the MID is 0x000E.
- The Message Length contains the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field will be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the Notification Message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The Action Field defines the action when a Notification Message is sent. The high order 5 bits of the first octet constitute a source for the Action Field definition. Currently identified sources are defined in Section 2.5.1. The next 11 bits constitute an Action ID. The high order bit of the Action Field is placed in bit 8 of the first octet of the field and the low order bit is placed in bit 1 of the second octet of the field. Standard values used for the Action Field are defined in Table 2.3-2.
- The Information Length field contains the actual length of the Information Field (plus the length of the Information Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 8 and the low order bit being bit 1 of octet 9.
- The Information Field is variable length and contains entries of the form shown in Table 2.3-3. The Information Field can be sent in any Notification Message, and is optional for all Action Field values except those for CKL Transfer, Secure Update, and Secure Dial. Notification Messages used for CKL Transfer, Secure Update, or Secure Dial shall contain only one Information Field Entry.

Table 2.3-2 SCIP Standard Action Field Values

Action Field Value	Action Definition
0x0002	Connection Terminate
0x0004	Native Clear Voice
0x0008	Connection Idle
0x0010	CKL Transfer
0x0020	Secure Dial
0x0040	Attention
0x0080	Secure Update

2872
2873
2874
2875
2876
2877

If a Notification Message is intended to carry only an “Action”, the Action Field is set to the desired value defined in Table 2.3-2, the Information Length field is set to 0x0002, and the optional Information Field Entries are not transmitted.

2878 If an optional Information Field Entry is present, its format shall be as shown in Table 2.3-3.
2879 Specifically, the Information Code field is set to one of the values in Table 2.3-4 (or to an
2880 implementer defined value with an appropriate Source ID). If the Information Code field is set
2881 to any of the entries in Table 2.3-4 other than 0x07FF, the optional Information Text field is not
2882 required and, if it is not present, the Information Text Length field is set to 0x0002. If the
2883 Information Code field is set to 0x07FF, the Information Text field is required. The only
2884 Notification Messages currently defined that require use of the optional Information Text field
2885 are CKL Transfer, Secure Update, and Secure Dial.
2886
2887
2888
2889

Table 2.3-3 Information Field Entry Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Information Code								
0-msb	0	0	0	0	X	X	X	1
		Source ID						
X	X	X	X	X	X	X	X-lsb	2
Information Text Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Information Text (Optional)								
X	X	X	X	X	X	X	X	5
X	X	X	X	...	X	X	X	4+L

2890 L = Length of Information Text.
2891

- 2892 • The Information Code field is set to one of the values in Table 2.3-4 (or to an
2893 implementer defined value with an appropriate Source ID). The high order 5 bits of
2894 the first octet constitute a source for the Information Code definition. Currently
2895 identified sources are defined in Section 2.5.1. The next 11 bits constitute an
2896 Information ID. The high order bit of the Information Code is placed in bit 8 of the
2897 first octet of the field, and the low order bit is placed in bit 1 of the second octet of
2898 the field. Standard values for Information Codes are defined in Table 2.3-4. Vendor
2899 specific values may also be used here. Notification Messages that have the Action set
2900 to either Secure Dial, Secure Update, or CKL Transfer shall set this to 0x07FF.
2901 Information Code 0x07FF may also be used in conjunction with any other
2902 Notification Message Action to convey additional information pertaining to the
2903 Notification that is not specifically identified by one of the predefined Information
2904 Codes. When a Notification Message, other than CKL Transfer, Secure Update, and
2905 Secure Dial, containing an Information Code of 0x07FF is received, the terminal shall
2906 recognize that the Information Text field contains additional information pertaining to
2907 the Notification; however, there are no requirements for the terminal to process this

2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929

information. A terminal shall not fail the call if an unrecognized Information Code is received.

- The Information Text Length contains the actual length of the Information Text field (plus the length of the Information Text Length field itself), in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4. If the optional Information Text field is not present, the Information Text Length field is set to 0x0002. **[Deviation Notice:** *When using the Notification Message for CKL Transfer and the first octet of the ID Information field of the Capabilities Message transmitted by the far-end terminal contains the value 0x28 (see Section 2.2.2.1), the Information Text Length shall be set to two octets less than the actual length of the Information Text field only, i.e., four octets less than the combined length of the two fields (see also Section 2.3.2.4).]*
- The Information Text is of variable length. This field is mandatory for Notification Messages that have the Action set to CKL Transfer (see Table 2.3-5 for format), Secure Dial (see Table 2.3-7 for format), or Secure Update (see Table 2.3-8 for format). In other cases this field, when present, shall carry 8-bit ASCII characters (bit 8 is the msb) that the transmitter would like the receiver to display (though there is no implied requirement that the receiver must actually do so).

Table 2.3-4 SCIP Standard Information Code Definitions

Information Code	Definition	Occurrences
0x0000	No initiator defined	Section 2.2.2.3 - Capabilities Message Reception
0x0003	No common operational modes	Section 2.2.2.3 - Capabilities Message Reception
0x0005	SCIP response not received	Section 2.2.1.2 - First Message Timeout
0x0006	No compatible keysets	Section 2.2.2.3 - Capabilities Message Reception
0x0009	Sync message verification failure	Sections 2.2.5.3 - Cryptosync Message Reception; 2.3.3.1 - Mode Change Request Message; 2.3.3.2 - Mode Change Response Message
0x000A	Seed key held	Section 2.2.2.3 - Capabilities Message Reception
0x000C	No matching parameters	Sections 2.2.2.3 - Capabilities Message Reception; 2.2.3.3 - Parameters/- Certificate Message Reception
0x000F	Security incompatibility	Sections 2.2.2.3 - Capabilities Message Reception; 2.2.3.3 - Parameters/- Certificate Message Reception

2930

2931
2932
2933

Table 2.3-4 SCIP Standard Information Code Definitions (Cont.)

Information Code	Definition	Occurrences
0x0011	Certificate verification failure	Section 2.2.3.3 - Parameters/Certificate Message Reception
0x0012	Certificate expired	Section 2.2.3.3 - Parameters/Certificate Message Reception
0x0014	Access Control failure	Sections 2.2.2.3 - Capabilities Message Reception; 2.2.3.3 -Parameters/-Certificate Message Reception
0x0017	Rekey Message CRC failure	Section 4.3 - Adaptation Layer
0x0018	Local CSE key expired	Section 2.2.3.2 - Parameters/Certificate Message Transmission
0x0041	Cryptosync/Mode Change glare	Section 2.3.3.1 - Mode Change Request Message; Section 2.3.4 - Two-Way Resync Processing
0x0042	Secure Restart	Section 2.3.2.3.4 - Secure Restart
0x07FF	Defined by Information Text field(s)	Sections 2.3.2.4, 2.3.2.5, and 2.3.2.7 - CKL Transfer, Secure Dial, and Secure Update, respectively; Section 2.3.2.1 for implementer defined display data

2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949

2.3.2.2 Notification (Connection Terminate)

Connection Terminate shall be available from any state during a call. Note that Connection Terminate is a state native to the underlying network in which the data channel is terminated. As such, it is outside the scope of SCIP-210 to specify how the network will terminate the data channel. Thus, Connection Terminate will only bring a terminal to the Connection Idle state with the provision that the network takes care of terminating the data channel. It is invoked when a terminal receives a local indication to terminate the connection. The Connection Terminate processing is shown in Figure 2.3-2.

Upon receipt of an indication to terminate the connection, the terminal shall format a Notification Message as shown in Table 2.3-1, with the Action set to Connection Terminate. The terminal will transmit this Notification Message to the far end and immediately enter the Connection Idle state.

2950

Editor's Note: For the transport layer, this implies that the terminal need not actually transmit the Notification Message nor wait to receive the REPORT acknowledging the transmitted message before entering the Connection Idle state. Conversely, if the far-end terminal does not receive the Notification Message with the Connection Terminate Action, it is assumed that there will be network indication that would bring the terminal to the Connection Terminate state.

2951

2952

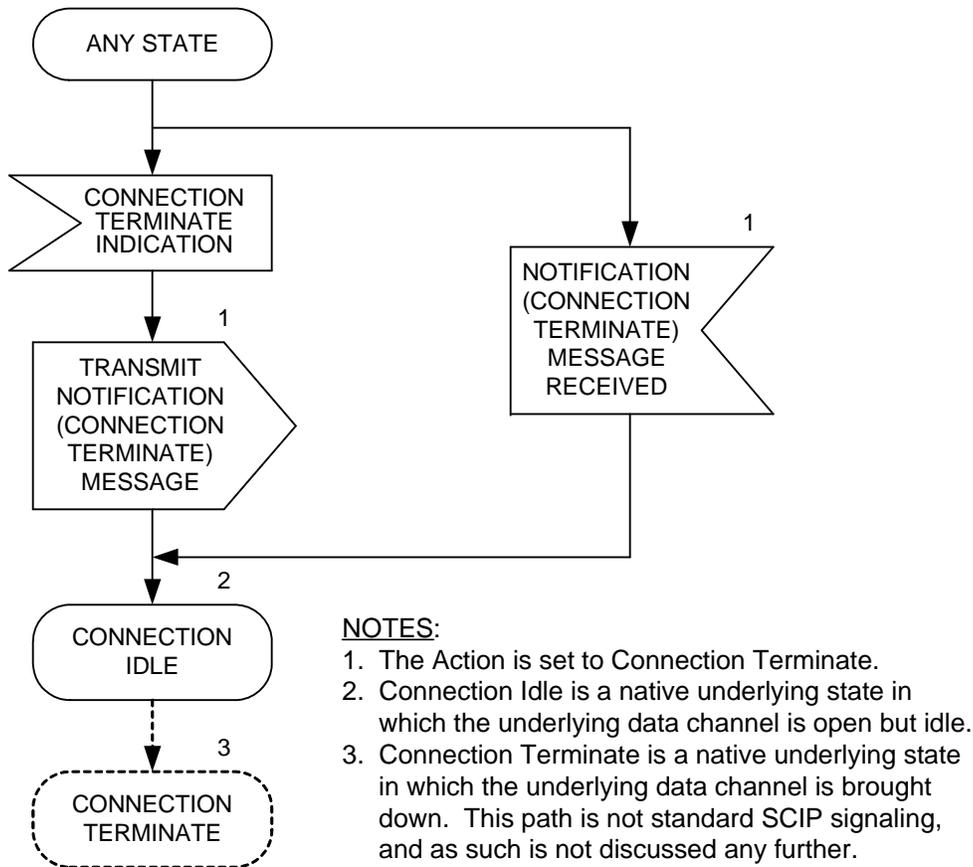
2953

2954

2955

2956

Upon receipt of a Notification Message with the Action set to Connection Terminate, the terminal shall immediately enter the Connection Idle state and then transition to Connection Terminate as shown in Figure 2.3-2.



2957

2958

2959

2960

Figure 2.3-2 Notification Message Processing (Connection Terminate)

2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974

2.3.2.3 Notification (Native Clear Voice/Connection Idle)

Notification Messages with Actions for Native Clear Voice and Connection Idle are used to perform four functions: Failed Call, Nonsecure Selected, Secure Selected, and Secure Restart. These functions are described in Section 2.3.2.3.1, Section 2.3.2.3.2, Section 2.3.2.3.3, and Section 2.3.2.3.4, respectively. Notification (Native Clear Voice/Connection Idle) receive processing is described in Section 2.3.2.3.5. The Actions for Native Clear Voice and Connection Idle shall be available from the indicated states during a call, except when the terminal is already processing a Notification (Connection Terminate) or a Notification (Native Clear Voice/-Connection Idle). The Notification (Native Clear Voice/Connection Idle) processing is shown in Figure 2.3-3.

2975

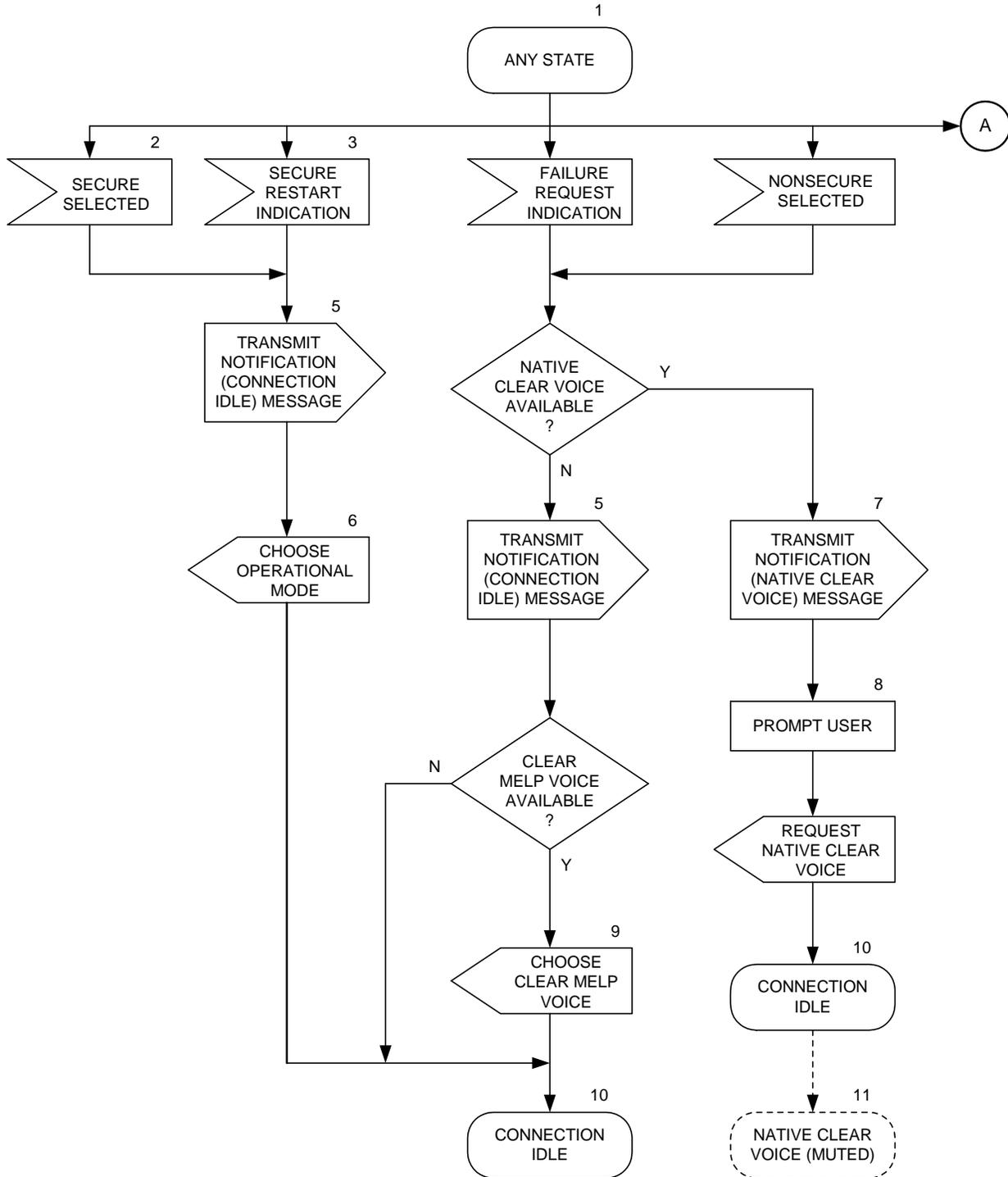


Figure 2.2-2

2976
2977
2978
2979

Figure 2.3-3(a) Notification Message Processing (Native Clear Voice/Connection Idle)

2980

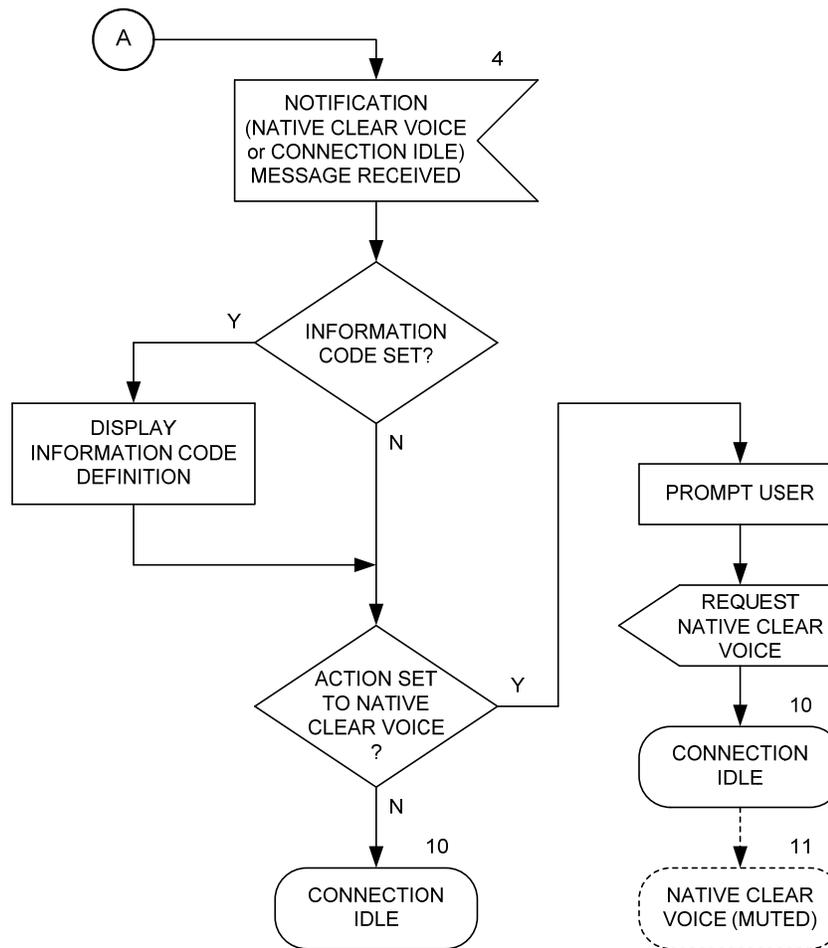


Figure 2.2-2

NOTES:

1. Can be entered from any state, except when the terminal is already processing another Notification (Native Clear Voice/Connection Idle) or a Notification (Connection Terminate).
2. Secure Selected followed by the transmission of a Notification Message occurs only from Clear MELP Voice.
3. Secure Restart occurs only from a secure application (not including Electronic Rekey).
4. The Action can be set to either Native Clear Voice or Connection Idle.
5. The Action is set to Connection Idle; for Secure Restart, an Information Code of *Secure Restart* is also included.
6. For Secure Selected, the selected secure application will appear as the first Entry in the Operational Modes List of the Initiator's Capabilities Message; for Secure Restart, the secure application just exited will be the first Entry.
7. The Action is set to Native Clear Voice.
8. If Native Clear Voice processing was initiated through user action (e.g., the user selected "Nonsecure"), the user need not be prompted again.
9. Clear MELP Voice will appear as the only Entry in the Operational Modes List of the Initiator's Capabilities message.
10. Connection Idle is a native underlying state in which the underlying data channel is alive but idle. See Section 2.2.2 for transitioning into other SCIP states from Connection Idle.
11. Native Clear Voice is an application native to the underlying network. This path is not standard SCIP signaling, and as such is not discussed any further.

2981
2982
2983
2984
2985

**Figure 2.3-3(b) Notification Message Processing (Native Clear Voice/Connection Idle)
(Cont.)**

2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008

2.3.2.3.1 Failed Call

Failed Call uses the Action of either Native Clear Voice, if available, or Connection Idle, otherwise. Native Clear Voice is an application native to the underlying network providing the data channel. As such, it is outside the scope of this Signaling Plan to specify how the network will handle transitions into it. Thus, Native Clear Voice will only bring a terminal to the Connection Idle state with the provision that the network takes care of transitioning into a clear voice application native to it. Connection Idle is a state native to the underlying network in which the data channel is alive but idle. Failed Call is invoked when a terminal receives a local Failure Request indication (e.g., as a result of internal error detection).

Upon receipt of a local Failure Request indication, the terminal shall format a Notification Message as shown in Table 2.3-1, with the Action set to either Native Clear Voice or Connection Idle. From the Capabilities Message Exchange of call setup (see Section 2.2.2), a terminal knows which clear applications it has in common with the far end. This information will be retained from call setup and made available for Failed Call processing.

- If both the local and remote terminals support Native Clear Voice, the local terminal shall format a Notification Message with the Action set to Native Clear Voice, transmit it to the far end, prompt the user, generate a local request to enter Native Clear Voice, and immediately enter the Connection Idle state.

Editor's Note: For Native Clear Voice, at the Transport Layer the terminal need not wait to receive the acknowledgment for the transmitted message before entering the Connection Idle state. Conversely, if the far-end terminal does not receive the Notification Message with the Native Clear Voice Action, it is assumed that there will be a network indication which would bring the terminal to Native Clear Voice. Note that even in the case where a terminal enters Native Clear Voice as a result of a network indication, the user must first acknowledge the transition.

3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022

- If Native Clear Voice is not available, the local terminal shall format a Notification Message with the Action set to Connection Idle and transmit it to the far end.
 - ◇ If both the local and remote terminals support Clear MELP Voice, the terminal shall generate a local request to enter Clear MELP Voice and go to the Connection Idle state. From the Connection Idle state, the terminal will request Clear MELP Voice by transmitting a Capabilities Message, with Clear MELP Voice as the only Operational Mode offered, in accordance with the signaling specified in Section 2.2.2. Clear MELP Voice is described in Section 3.3.1.3.
 - ◇ If the local and remote terminals have no clear voice application in common, the local terminal shall go to the Connection Idle state.

3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068

2.3.2.3.2 Nonsecure Selected

Nonsecure Selected shall be identical to Failed Call except that it is invoked when a terminal receives a local Nonsecure Selected indication (e.g., as the result of the user selecting “Nonsecure”). Additionally, the terminal receiving the local Nonsecure Selected indication will not prompt the user prior to entering the Connection Idle state.

2.3.2.3.3 Secure Selected

Secure Selected uses only the Action of Connection Idle. Secure Selected is invoked from Clear MELP Voice when a terminal receives a local Secure Selected indication (e.g., as the result of the user selecting “Secure”).

Upon receipt of a local Secure Selected indication, the terminal shall format a Notification Message as shown in Table 2.3-1, with the Action set to Connection Idle, and transmit it to the far end. The terminal shall then generate a local request to enter an Operational Mode with the selected mode as the preferred mode, and go to the Connection Idle state. From the Connection Idle state, the terminal will then enter secure call setup by transmitting a Capabilities Message in accordance with the signaling specified in Section 2.2.2.

2.3.2.3.4 Secure Restart

Secure Restart provides the capability for terminals executing a secure application to generate a new traffic encryption key using the FIREFLY or ECMQV Key Exchange and return to the same secure application. Secure Restart is invoked when a terminal in a secure application, other than Electronic Rekey, receives a local Secure Restart indication (e.g., for the case described in SCIP-230, Section 3.3.1, SCIP-231, Section 3.1.4.1, or SCIP-232, Section 3.4.1).

Upon receipt of a local Secure Restart indication, the (Leader) terminal shall format a Notification Message as shown in Table 2.3-1, with the Action set to Connection Idle and an Information Code of *Secure Restart*, and transmit it to the far end. The Leader terminal shall then generate a local request to enter an Operational Mode with the mode and parameter option just exited as the preferred mode and option, and go to the Connection Idle state. From the Connection Idle state, the Leader terminal shall enter secure call setup by transmitting a Capabilities Message with the I/R bits set to Initiator in accordance with the signaling specified in Section 2.2.2. In this Capabilities Message, the Leader terminal shall offer only FIREFLY or NATO ECMQV/AES Keysets with the same KMID, or the ECMQV/AES Keyset that was in use prior to the Secure Restart.

The Secure Restart Follower terminal, after receiving the Notification (Connection Idle) Message waits in the Connection Idle state until it receives the Capabilities Message transmitted by the Leader. Secure call setup then proceeds in the same manner as for any secure call with the I/R bits set to Responder in the Capabilities Message transmitted by the Follower terminal.

3069
3070 If the classification is changed in a Secure Restart, both the Leader and Follower terminals shall
3071 prompt the user and wait for an acknowledgment before transmitting secure traffic. Secure
3072 Restart places no other special requirements on the Follower terminal.
3073

3074 **2.3.2.3.5 Notification (Native Clear Voice/Connection Idle) Receive Processing**

3075
3076 Upon receipt of a Notification (Native Clear Voice/Connection Idle) Message, the terminal shall
3077 determine whether an Information Code is included. If an Information Code is included, the
3078 terminal will display a text message locally associated with the value contained in the
3079 Information Code field. (Implementers are permitted to associate different locally defined
3080 display texts with the Standard Information Codes contained in Table 2.3-4 so long as the text
3081 conveys the intended meaning of the code to the user.) The terminal will also determine whether
3082 Information Text is included. Information Text received in a Notification Message is text that
3083 the transmitter intends be displayed to the user (though this Signaling Plan levies no requirement
3084 on the recipient terminal to actually display this text). The terminal shall then examine the
3085 Action field. If the Action is set to Native Clear Voice, the terminal shall prompt the user,
3086 generate a local request to enter Native Clear Voice, and immediately enter the Connection Idle
3087 state. If the Action is set to Connection Idle, the terminal shall go to the Connection Idle state.
3088 From the Connection Idle state, the terminal will wait for the receipt of a Capabilities Message
3089 and then enter SCIP call setup in accordance with the signaling specified in Section 2.2.2.
3090
3091

3092 **2.3.2.4 Notification (CKL Transfer)**

3093
3094 CKL Transfer allows a terminal to transmit its CKL to the far-end terminal. During secure call
3095 setup, the versions of the CKL held by both terminals are compared. If the local terminal's CKL
3096 version is later than that of the far-end terminal, the local terminal transmits its CKL to the far-
3097 end terminal (see Sections 2.2.5.2 and 2.2.5.3).
3098

3099
3100 Since the CKL is large, it may be segmented and transmitted in multiple Notification Messages.
3101 Of course, the entire CKL may be transmitted as a single segment in a single Notification
3102 Message. Rules for segmenting the CKL are left to the implementer since, while they may
3103 impact performance, such rules do not impact interoperability.
3104

3105 This section describes the processing of a single Notification Message containing a single CKL
3106 segment. If the CKL has been segmented for transmission, the process described below shall be
3107 performed as many times as there are segments.
3108

3109 CKL Transfer shall be available only during SCIP Call Setup after a Cryptosync Message has
3110 been received and before the locally generated Cryptosync Message has been transmitted (see
3111 Figure 2.2-10).
3112

3113 Upon the local determination that a CKL Transfer is required, the terminal shall format a
3114 Notification Message as shown in Table 2.3-1, with the Action set to CKL Transfer and the

3115 Information Text formatted as shown in Table 2.3-5, and transmit it to the far-end terminal as
3116 shown in Figure 2.2-10.

3117
3118
3119
3120

Table 2.3-5 CKL Transfer - Information Text

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
X	X	X	X	X	X	X	X	1
X	X	X	X	X	X	X	X	2
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
X	X	X	X	X	X	X	X	5
CKL Segment (First Octet of CKL Segment) •••								
X	X	X	X	X	X	X	X	4+L
(L'th Octet of CKL Segment)								

3121 L = Length of CKL Segment.

3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142

- Segment Number indicates the relative position of the current Notification Message in the sequence of Notification Messages used to transmit the CKL. This value shall be represented as an unsigned binary integer with the high order bit being bit 8 and the low order bit being bit 1. 0x00 is presently RESERVED. 0x01 is used to indicate the first Notification Message in the sequence.
- Number of Segments indicates how many Notification Messages in total are used to transmit the CKL. This value shall be represented as an unsigned binary integer with the high order bit being bit 8 and the low order bit being bit 1. Set to 0x00 if unused/unknown (e.g., if the terminal has not yet determined how it will segment the remainder of the CKL). This field and the Segment Number field shall be set to the same value in the Notification Message that carries the final segment of the CKL. The CKL Segment Length field contains the actual length of the CKL Segment (plus the CKL Segment Length field itself) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4. **[Deviation Notice:** *When the first octet of the ID Information field of the Capabilities Message transmitted by the far-end terminal contains the value 0x28 (see Section 2.2.2.1), a CKL transmitted to this terminal shall have the CKL Segment Length set to the actual length of the CKL Segment field only. The length of the CKL Segment Length field itself shall not be included. A CKL received from this terminal will also be formatted in this manner. However, there is*

- 3143 *no requirement to either transmit a CKL to this terminal or to process a CKL*
3144 *received from it.]*
- 3145 • CKL Segment Blocks (defined in SCIP-230 or SCIP-232, Section 2.1.2.1.1) shall be
3146 transmitted in order, i.e., the Block containing M11 precedes the Block containing
3147 M21, precedes the (optional) Block containing M12, precedes the (optional) Block
3148 containing M22. Within a Block the bits are ordered from high to low based on the
3149 calculation defined in SCIP-230 or SCIP-232. The high order bit of the Block
3150 containing M11 shall be placed in Bit 8 of the first octet of the first Segment
3151 transmitted, and the low order bit of the last Block shall be placed in Bit 1 of the last
3152 octet of the last Segment transmitted.

3153
3154 Upon receipt of a Notification Message with the Action set to CKL Transfer, the terminal will
3155 store the CKL segment. This process is shown in Figure 2.3-4. Note that it occurs in the 'Wait
3156 for CS Message' state shown in Figure 2.2-10.

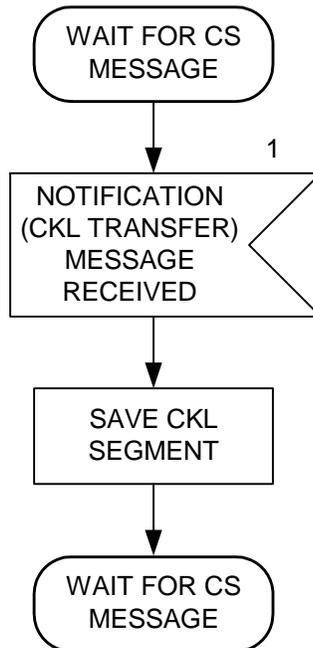
3157
3158 When the CKL has been received in its entirety, the terminal will process it in accordance with
3159 SCIP-230 or SCIP-232, Section 2.1.2.

3160

Editor's Note: No requirements are implied as to when the processing of a received CKL will occur. This may in fact occur after the call has been completed.

3161

3162



NOTES:

1. The Action is set to "CKL Transfer".

3163

3164

3165

3166

3167

3168

3169

Figure 2.3-4 Notification Message Receive Processing (CKL Transfer)

2.3.2.5 Notification (Secure Dial)

Editor's Note: The ability to transmit Secure Dial Characters is a required capability, but the ability to process received Secure Dial Characters is optional.

3170

3171

3172

3173

3174

3175

3176

3177

Secure Dial allows a terminal or gateway to transmit encrypted control panel information or other dialing data to the far end and to receive encrypted information from the far-end terminal for display on the control panel or for use in controlling a red gateway. This capability is provided to allow the local terminal to gain access to gateway and interworking equipment and to control it remotely. The characters that may be used as Secure Dial Characters are listed in Table 2.3-6.

3178
3179
3180

Table 2.3-6 Secure Dial Characters

ASCII character (8 bit format)	Definition
0 - 9	0-9
*	*
#	#
T	change to TONE dialing mode
P	change to PULSE dialing mode
,	pause
H	hookflash
A	Autovon FO
B	Autovon F
C	Autovon I
D	Autovon P
R	hookswitch reset
E	end of dialing
F	go off-hook
N	go on-hook

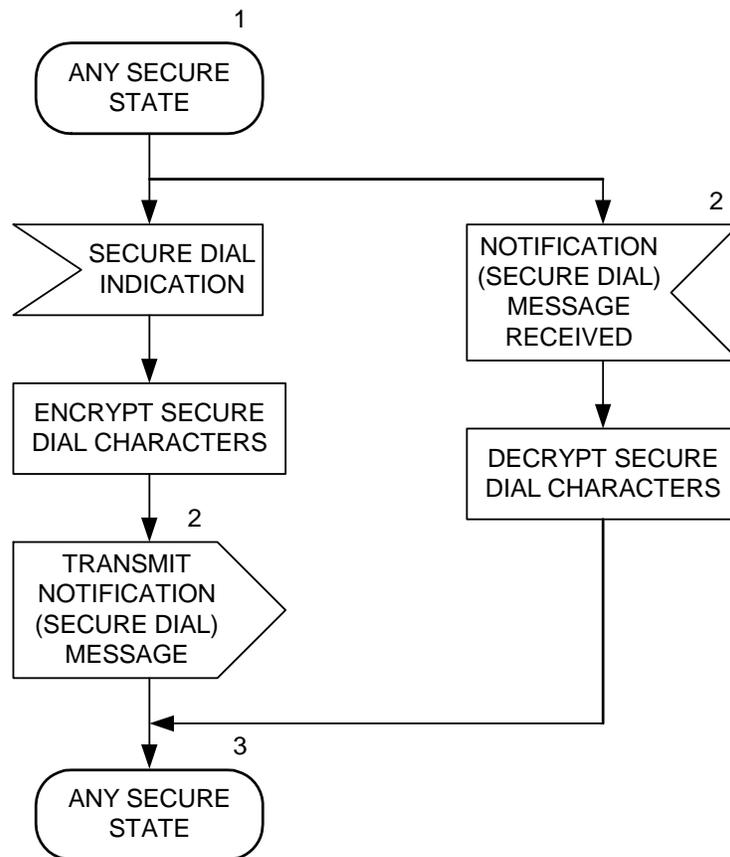
3181
3182

Editor’s Note: Use of the “end of dialing” character (see Table 2.3-6) is optional and left to the discretion of the implementer.

3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194

Secure Dial Characters may be transmitted in one or more Notification Messages with each Notification Message containing one to twelve Secure Dial Characters. The Notification Messages are transmitted in an order so that the first Characters to be displayed or to be passed to the red gateway are transmitted first. Characters may either be accumulated or may be transferred as soon as they are available.

This section describes processing of a single Notification Message. This processing is shown in Figure 2.3-5. If the Secure Dial characters are to be transmitted in multiple Notification Messages, the processing described in this section will be repeated for each Notification Message until all dialing information has been sent.



NOTES:

1. Can be entered from any secure state, once Cryptosync Messages have been exchanged and verified, until that state is exited by processing an Action for Native Clear Voice/Connection Idle or Connection Terminate.
2. The Action is set to Secure Dial.
3. If Secure Dial was entered from application traffic, the same application is re-entered. See Section 3.

3195
3196
3197
3198
3199
3200
3201
3202
3203
3204

Figure 2.3-5 Notification Message Processing (Secure Dial)

Secure Dial shall be available any time after the key has been negotiated and verified (i.e., Cryptosync Messages have been exchanged) for as long as the key remains available for use. (Note that the key is no longer available for use after a Native Clear Voice/Connection Idle or a Connection Terminate operation).

3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233

2.3.2.5.1 Encryption of Secure Dial Characters

The encryption of Secure Dial characters is specified in SCIP-230 or SCIP-231, Section 4.1.3.1; or SCIP-232, Section 4.3.1. The Secure MELP Voice encryption mode is used. In this mode two 54 bit frames are encrypted for each value of the state vector (which in Secure Dial is based on the value carried in the IV field of the Notification Message). The Secure Dial characters are ordered as they will be displayed or passed to the red gateway (e.g., character 1 in the message is to be displayed before character 2, etc.).

The Secure Dial characters to be included in a Notification Message shall be formatted into one or two six-character frames prior to encryption. If there are fewer than six characters in a frame, padding may be used to complete the frame. While the IV is updated for each Notification Message sent, for a single Notification Message both frames shall be encrypted, as specified in SCIP-230 or SCIP-231, Section 4.1.3.1; or SCIP-232, Section 4.3.1, using the same IV.

After the data has been encrypted, it is transmitted in the Information Text field of a Notification Message. Only the encrypted bits corresponding to the Secure Dial characters shall be transmitted. Encrypted padding octets (if present) shall be discarded.

2.3.2.5.2 Data Transmission and Reception

The terminal shall format a Notification Message as shown in Table 2.3-1, with the Action set to Secure Dial and the Information Text formatted as shown in Table 2.3-7, and transmit it to the far-end terminal. If the Secure Dial transmission interrupted full bandwidth application traffic, after the Notification Message has been acknowledged the terminal will re-enter the same application using the signaling specified in Section 3.2.

3234
3235
3236

Table 2.3-7 Secure Dial - Information Text

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Segment Number								
X	X	X	X	X	X	X	X	1
Number of Segments								
X	X	X	X	X	X	X	X	2
IV Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
IV								
(First Octet of IV)								
X-msb	X	X	X	X	X	X	X	5
•••								
(L'th Octet of IV)								
X	X	X	X	X	X	X	X-lsb	4+L
Secure Dial Packet Length								
X-msb	X	X	X	X	X	X	X	5+L
X	X	X	X	X	X	X	X-lsb	6+L
Secure Dial Packet								
(First Encrypted Character)								
b8 - msb	b7	b6	b5	b4	b3	b2	b1 - lsb	7+L
•••								
(M'th Encrypted Character)								
b8 - msb	b7	b6	b5	b4	b3	b2	b1 - lsb	6+L+M

3237 L = Length of IV.
3238 M = Length of Secure Dial Packet.

3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250

- The Segment Number indicates the relative position of a Notification Message in a sequence of multiple Notification Messages used to transmit the Secure Dial characters. This value shall be represented as an unsigned binary integer with the high order bit being bit 8 and the low order bit being bit 1. 0x00 is presently RESERVED. 0x01 is used to indicate the first of many Notification Messages, 0x02 the second, etc.
- Number of Segments indicates how many Notification Messages in total are used to transmit the Secure Dial characters. This value shall be represented as an unsigned binary integer with the high order bit being bit 8 and the low order bit being bit 1. Set to 0x00 if unused/unknown (e.g., if the end of the user dialing sequence is unknown until a specific character is dialed).

- 3251 • The IV Length field contains the actual length of the IV field (plus the IV Length
3252 field itself) in octets. The value of the field shall be an unsigned binary integer with
3253 the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- 3254 • The IV field shall contain the IV used to encrypt the Secure Dial Characters. Details
3255 of the length, format, and contents are found in SCIP-230, Section 3.5.1, SCIP-231,
3256 Section 3.3.1, or SCIP-232, Section 3.6.1. The msb of the IV (as defined in SCIP-
3257 23x) is placed in bit 8 of octet 5.
- 3258 • Secure Dial Packet Length field contains the actual length of the Secure Dial Packet
3259 (plus the Secure Dial Packet Length field itself) in octets. The high order bit of the
3260 Secure Dial Packet Length is placed in bit 8 of the first octet of the field and the low
3261 order bit is placed in the second octet of the field.
- 3262 • Secure Dial Packet. Contains one to twelve encrypted Secure Dial Characters.

3263
3264 Upon receipt of a Notification Message with the Action set to Secure Dial, the terminal shall
3265 decrypt the Secure Dial characters and make them available either for display or for use in
3266 controlling a red gateway. If the Secure Dial transmission interrupted full bandwidth application
3267 traffic, after the Notification Message has been correctly received and acknowledged the
3268 terminal will re-enter the same application using the signaling specified in Section 3.2.

3269

Editor's Note: If additional secure Notifications are added to the Signaling Plan, the intent is to follow the general structure shown in Table 2.3-7, i.e., the Information Text field will carry an IV followed by encrypted data.

3270

3271

3272 2.3.2.6 Notification (Attention)

3273

Editor's Note: Implementation of Attention is optional. This means that a terminal does not have to implement the capability to transmit it nor to process it (i.e., alerting the user). However, a terminal receiving an Attention is required to acknowledge it in the same manner as with all other standard SCIP messages. If it is implemented, terminals with this capability will behave in accordance to the specifications outlined in this section.

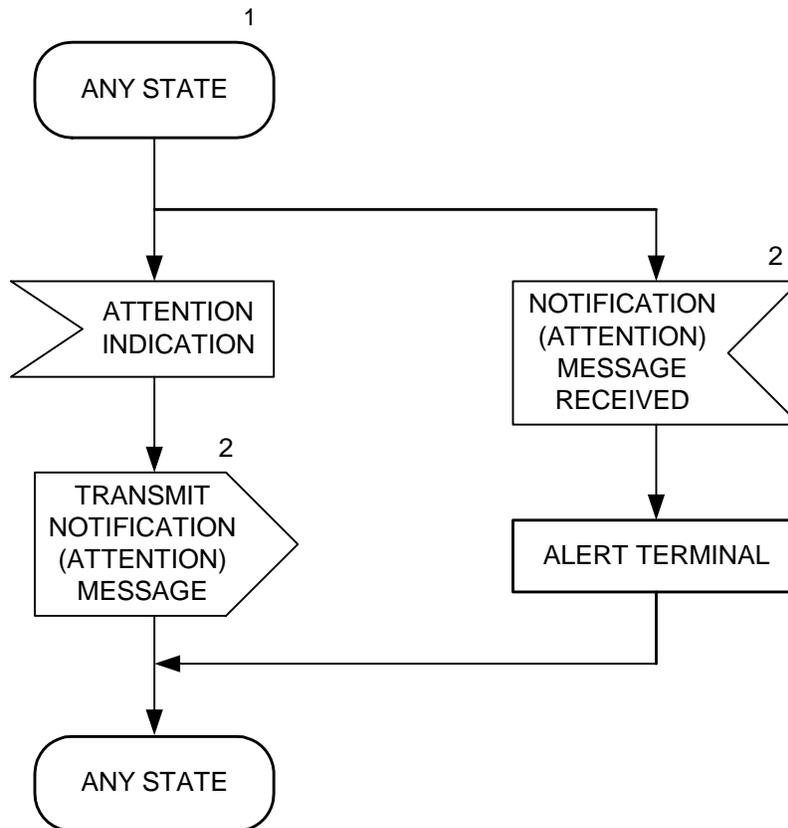
3274

3275 Attention shall be available from any state during a call except when the terminal is already
3276 processing a Native Clear Voice/Connection Idle or Connection Terminate. It is invoked when a
3277 terminal receives a local indication to send an Attention to the far end. When received, the
3278 Notification Message (containing the Attention option) alerts the terminal to warn the user by
3279 performing a vendor elected action (e.g., emitting an audible tone, blinking the display, etc.).
3280 This processing is shown in Figure 2.3-6.

3281

3282 Upon receipt of an Attention indication, the terminal shall format a Notification Message as
3283 specified in Table 2.3-1, with the Action set to Attention, and transmit it to the far-end terminal.
3284 If the entry to Attention processing was from an application, the terminal shall re-enter the same
3285 application via processing as specified in the subsection of Section 3 that describes that
3286 application.

3287 Upon receipt of a Notification Message with the Action set to Attention, the terminal shall alert
3288 the user. If the entry to Attention processing was from an application (either clear or secure), the
3289 terminal shall re-enter the same application via processing as specified in the subsection of
3290 Section 3 that describes that application.
3291
3292



NOTES:

1. Can be entered from any state except when the terminal is already processing an Action for Native Clear Voice/Connection Idle or Connection Terminate.
2. The Action is set to Attention.
3. If this process is entered from application traffic, the same application is re-entered. See Section 3.

3293
3294
3295
3296

Figure 2.3-6 Notification Message Processing (Attention)

3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334

2.3.2.7 Notification (Secure Update)

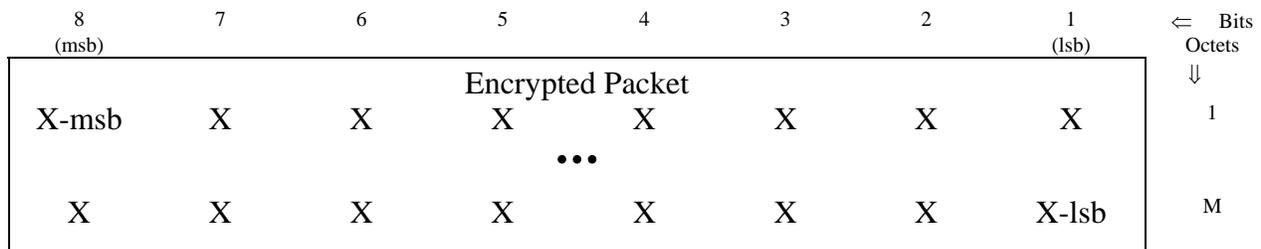
Secure Update provides the capability for terminals in a secure call using a PPK to update the PPK, switch from the currently active PPK to the updated PPK, and return to the same secure application. Secure Update is analogous to Secure Restart (Section 2.3.2.3.4), which is used to generate a new traffic encryption key using the FIREFLY or ECMQV Key Exchange during a secure call and return to the same secure application.

During a Secure Update, the Leader terminal always updates the currently active PPK. If the Follower terminal is configured for automatic updates, it updates the currently active PPK automatically. If both terminals successfully update the currently active PPK, the updated PPK is negotiated. Otherwise, the currently active PPK is renegotiated.

Secure Update shall be available for use only when a PPK is in use and both terminals transmitted Message Version 1 or higher Capabilities Messages during secure call setup. Secure Update is invoked when a terminal in a secure application, other than Electronic Rekey, receives a local Secure Update indication (e.g., for the cases described in SCIP-230, Sections 2.1.1.8.3 and 3.3.1.3, or SCIP-232, Sections 2.1.1.8.3 and 3.4.1.3).

Upon receipt of a local Secure Update indication, the Leader terminal shall generate an Encrypted Packet, as specified in SCIP-230, Section 3.4.2.4, or SCIP-232, Section 3.5.2.4, to be verified by the Follower terminal. The Leader terminal shall then format a Notification Message as shown in Table 2.3-1, with the Action set to Secure Update and the Information Text formatted as shown in Table 2.3-8, and transmit it to the far end. The Leader terminal shall then generate a local request to enter an Operational Mode with the mode and parameter option just exited as the preferred mode and option, and go to the Connection Idle state. The Leader terminal shall then update the currently active PPK. Only the update of the currently active PPK and the PPK in use prior to the Secure Update shall be offered in the subsequent secure call setup (except for the case specified in SCIP-230, Section 3.3.1.3, or SCIP-232, Section 3.4.1.3, where only the update of the currently active PPK is offered). From the Connection Idle state, the Leader terminal shall enter secure call setup by transmitting a Capabilities Message with the I/R bits set to Initiator in accordance with the signaling specified in Section 2.2.2. The Keysets shall be ordered as specified in SCIP-230 or SCIP-232, Section 2.1.1.8.3.

Table 2.3-8 Secure Update - Information Text



M = Length of Encrypted Packet.

3335
3336

- Inclusion of the Encrypted Packet is mandatory in the Secure Update Notification Message. The msb of the Encrypted Packet (as defined in SCIP-230 or SCIP-232) is placed in Bit 8 of the first octet of the Encrypted Packet field. The length, the encryption algorithm and mode to be used, and the content and format of the plaintext data to be encrypted are defined in SCIP-230, Section 3.4, or SCIP-232, Section 3.5.

If configured for automatic updates, the Secure Update Follower terminal, after receiving the Notification (Secure Update) Message, shall verify the Encrypted Packet contained in the Secure Update Notification Message as specified in SCIP-230, Section 3.4.2.4, or SCIP-232, Section 3.5.2.4. If the Follower terminal is not configured for automatic updates, it shall offer only the last negotiated PPK in the subsequent secure call setup.

If the Encrypted Packet verifies, the Follower terminal shall automatically update the currently active PPK and offer this update and the last negotiated PPK in the subsequent secure call setup.

If the Encrypted Packet does not verify, the Follower terminal shall offer only the last negotiated PPK in the subsequent secure call setup.

The Follower terminal then waits in the Connection Idle state until it receives the Capabilities Message transmitted by the Leader. Secure call setup then proceeds in the same manner as for any secure call with the I/R bits set to Responder in the Capabilities Message transmitted by the Follower terminal. This processing is shown in Figure 2.3-7.

If the PPK in use prior to the Secure Update is negotiated and the terminals are configured for attended operation, both the Leader and Follower terminals shall prompt the user to acknowledge that the key update operation did not occur successfully and that the PPK in use prior to the Secure Update will be used to continue the secure call. The terminals shall wait for a user acknowledgment before transmitting secure traffic. If the user accepts the negotiated key, the call shall proceed to secure traffic using the negotiated key. If the user does not accept the key, the terminal shall execute Failed Call processing as specified in Section 2.3.2.3.1. The user prompt may be disabled for terminals configured for unattended operation (see SCIP-230 or SCIP-232, Section 2.1.1.8.3).

3370

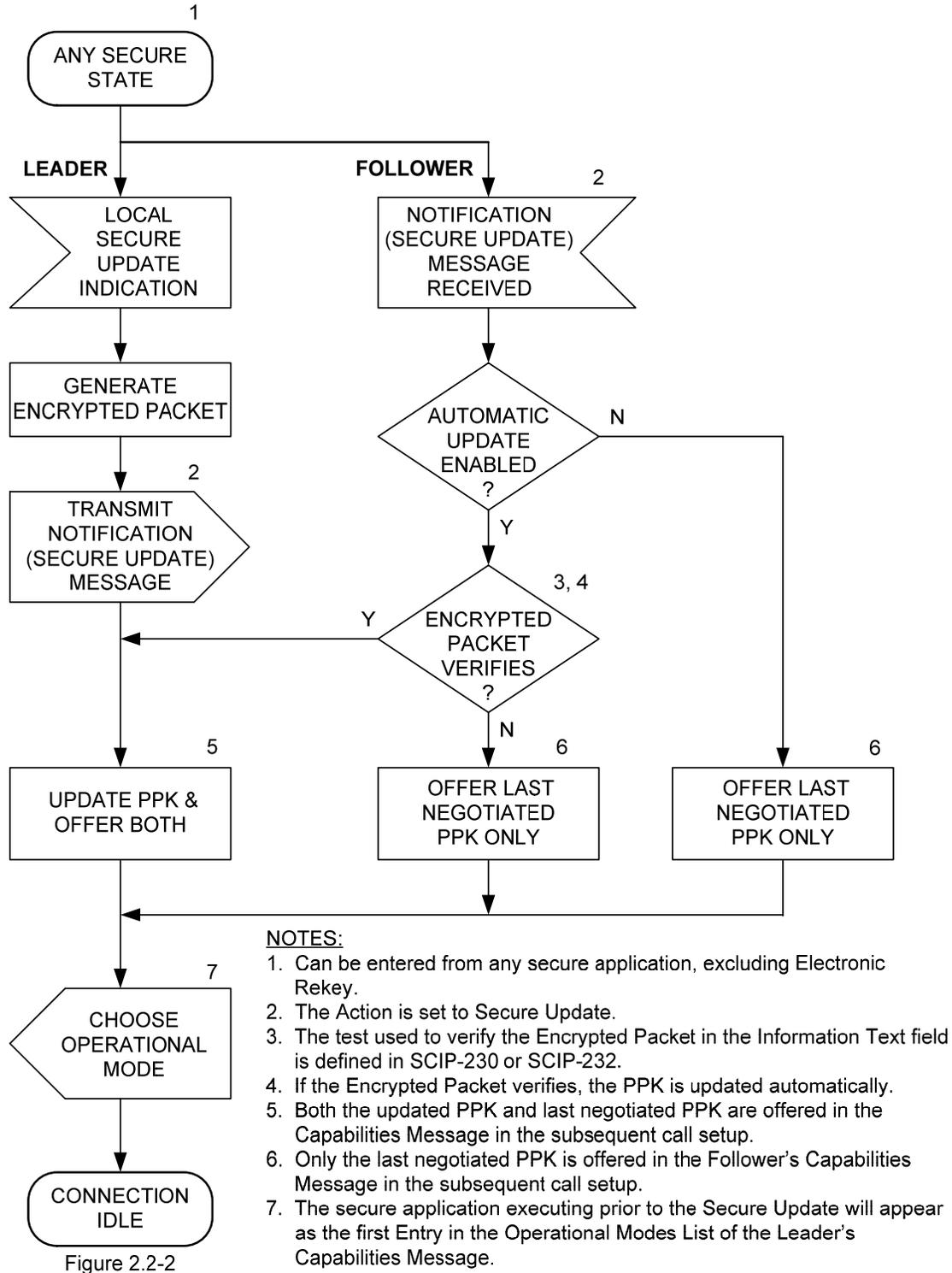


Figure 2.2-2

Figure 2.3-7 Notification Message Processing (Secure Update)

3371

3372

3373

3374

3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387

2.3.3 Mode Change Processing

This section specifies the signaling associated with Mode Change processing. Mode Change processing shall be available for use only when both terminals transmitted Message Version 1 or higher Capabilities Messages during SCIP secure call setup, and will be entered only when both terminals are in secure application traffic. The use of Mode Change shall be limited to changing from one secure application to a different secure application, or to the same secure application with different parameters, using the same key and the same traffic encryption algorithm. Two messages are involved: Mode Change Request and Mode Change Response. Section 2.3.3.1 specifies the Mode Change Request Message, and Section 2.3.3.2 specifies the Mode Change Response Message. The signaling is shown in Figure 2.3-8.

Editor's Note: Currently, the only standard SCIP clear application defined is Clear MELP Voice. In the event that other standard SCIP clear applications are defined, Mode Change may need to be updated to include changing from one clear application to another one.

3388

3389

2.3.3.1 Mode Change Request Message

3390

3391

3392

3393

3394

3395

3396

3397

3398

3399

3400

3401

3402

3403

3404

3405

3406

3407

3408

3409

3410

3411

3412

3413

3414

Mode Change is invoked when a terminal in a secure application receives a local Mode Change indication. The terminal will ensure that the requested Operational Mode is one common to both terminals and that it is allowed by the ACL, if the ACL has been activated for the chosen Operational Mode (See SCIP-230 or SCIP-232, Section 2.1.3.1.2) and the chosen Keypset Type is supported by the ACL, before proceeding. If the ACL has not been activated for the chosen Operational Mode and/or the chosen Keypset Type is not supported by the ACL, the ACL check is skipped. Upon receipt of a local Mode Change indication, the terminal shall assume the role of Leader, format a Mode Change Request Message, and transmit it to the far end. The format of the Mode Change Request Message shall be as specified in Table 2.3-9. The Leader shall then wait for a Mode Change Response Message from the far-end terminal.

Upon receipt of a Mode Change Response Message, the Leader shall format a Cryptosync Message as specified in Section 2.2.5, transmit it to the far-end terminal, and wait for a Cryptosync Message. Upon receipt of a Cryptosync Message, the terminal shall verify the Encrypted Packet contained in the Cryptosync Message as specified in SCIP-230, Section 3.4.2, SCIP-231, Section 3.2.2, or SCIP-232, Section 3.5.2. If this check does not pass, the terminal shall execute Failed Call processing, defined in Section 2.3.2.3.1, with an Information Code of *sync message verification failure*. If the Encrypted Packet check passes and there was no classification change as a result of the Mode Change, the terminal shall initiate the indicated Operational Mode as specified in Section 3. If the classification was changed during the Mode Change, the user shall be prompted, and an acknowledgment is required prior to initiating the indicated Operational Mode.

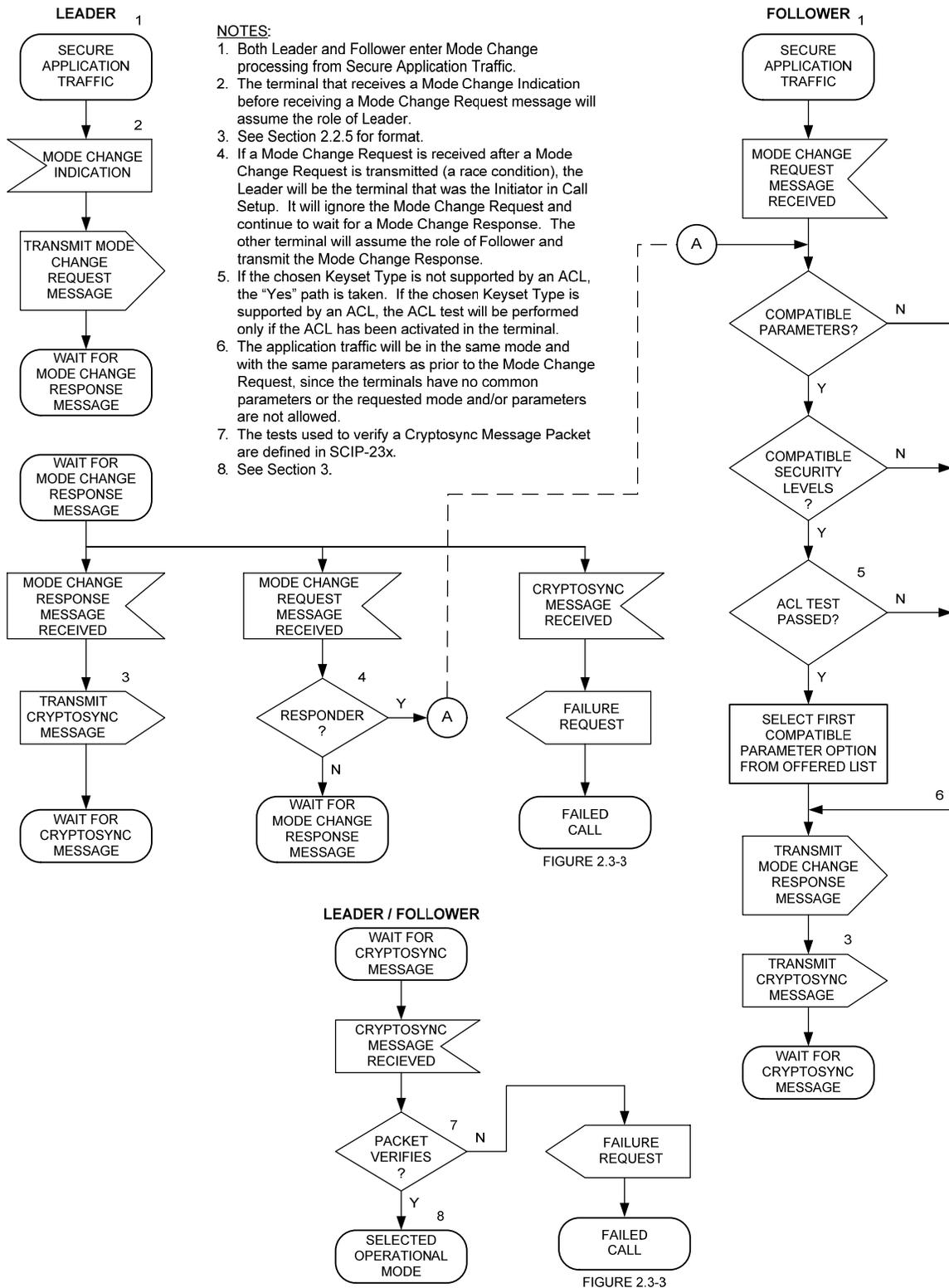


Figure 2.3-8 Mode Change Processing

3415
3416
3417

3418 The following signaling will take place in the event of a race condition, i.e., both terminals
3419 receive local Mode Change indications and transmit Mode Change Request Messages. The
3420 terminal that was determined to be the Responder in call setup shall assume the role of Follower,
3421 and the other terminal shall assume the role of Leader. The Leader, upon receipt of a Mode
3422 Change Request Message, shall ignore it, wait for the Mode Change Response Message, and
3423 continue in the manner described above when it is received. The Follower, upon receipt of a
3424 Mode Change Request Message shall proceed in the manner described in Section 2.3.3.2.

3425 In the event of a glare condition, i.e., instead of receiving the expected Mode Change Response
3426 Message, the Leader receives a Cryptosync Message, the following signaling shall take place.
3427 Upon receipt of the Cryptosync Message, the terminal shall initiate Failed Call processing as
3428 specified in Section 2.3.2.3 with the Information Code set to *Cryptosync/Mode Change glare*.
3429

3430
3431
3432 **Table 2.3-9 Mode Change Request Message Format**

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								1
0-msb	0	0	0	0	0	0	0	
Source ID								2
0	0	0	1	1	0	1	0-lsb	
Message Length								3
X-msb	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X-lsb	4
Message Version								5
0	0	0	0	0	0	0	0	
Operational Mode								6
X-msb	X	X	X	X	X	X	X	
Source ID								7
X	X	X	X	X	X	X	X-lsb	
Parameters Length								8
X-msb	X	X	X	X	X	X	X	
X	X	X	X	X	X	X	X-lsb	9
Operational Mode Parameters (Optional)								10
X	X	X	X	...	X	X	X	
X	X	X	X	...	X	X	X	9+L

3434 L = Length of Operational Mode Parameters.

3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480

- For the Mode Change Request Message, the value of the MID is 0x001A.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the Mode Change Request Message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The Operational Mode field shall contain the ID of the selected Operational Mode. For the format and values of these IDs, see the definition of Operational Mode IDs in Section 2.2.2.1. The high order bit of the Operational Mode field is placed in bit 8 of octet 6 and the low order bit is placed in bit 1 of octet 7. Note that this selected Operational Mode will be one supported by both terminals.
- The Parameters Length field contains the actual length of the Operational Mode Parameters field (plus the length of the Parameters Length itself), in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of the first octet of the field and the low order bit being bit 1 of the second octet of the field.
- The Operational Mode Parameters field shall contain parameters for the selected Operational Mode. The length, format, and contents of the Operational Mode Parameters are unique to each Operational Mode and are defined in Section 2.2.6 for each standard Operational Mode having a Parameters/Certificate Exchange. This field is optional and is not present unless Parameters are defined for a given Operational Mode.

2.3.3.2 Mode Change Response Message

Upon receipt of a Mode Change Request Message while in a secure application, the terminal shall assume the role of Follower. It shall then check for compatible parameters and security levels for the offered Operational Mode. If the ACL has been activated for the chosen Operational Mode and the chosen Keypset Type is supported by the ACL, the terminal shall also perform the ACL test as specified in SCIP-230 or SCIP-232, Section 2.1.3.1.2. If there are compatible parameters and compatible security levels and the ACL test passes, the Follower shall accept the offered Operational Mode. If there are no compatible parameters or no compatible security levels, or if the ACL test fails, the terminals shall continue executing the current Operational Mode. If the ACL has not been activated for the chosen Operational Mode and/or the chosen Keypset Type is not supported by the ACL, the ACL check is skipped.

The Follower shall transmit to the far end a Mode Change Response Message formatted as specified in Table 2.3-10 indicating the selected Operational Mode and Operational Mode Parameters Option. It shall then format a Cryptosync Message as specified in Section 2.2.5, transmit it to the far-end terminal, and wait for a Cryptosync Message. Upon receipt of a Cryptosync Message, the terminal shall verify the Encrypted Packet contained in the Cryptosync Message as specified in SCIP-230, Section 3.4.2.3, SCIP-231, Section 3.2.2, or SCIP-232, Section 3.5.2.3. If this check does not pass, the terminal shall execute Failed Call processing,

3481 defined in Section 2.3.2.3.1, with an Information Code of *sync message verification failure*. If
3482 the Encrypted Packet check passes and there was no classification change as a result of the Mode
3483 Change, the terminal shall initiate the indicated Operational Mode as specified in Section 3. If
3484 the classification was changed during the Mode Change, the user shall be prompted, and an
3485 acknowledgment is required prior to initiating the indicated Operational Mode.

3487 **Table 2.3-10 Mode Change Response Message Format**

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
0-msb	0	0	0	0	0	0	0	1
MID								
Source ID								
0	0	0	1	1	1	0	0-lsb	2
X-msb	X	X	X	X	X	X	X	3
Message Length								
X	X	X	X	X	X	X	X-lsb	4
0	0	0	0	0	0	0	0	5
Message Version								
X-msb	X	X	X	X	X	X	X	6
Operational Mode								
Source ID								
X	X	X	X	X	X	X	X-lsb	7
X-msb	X	X	X	X	X	X	X	8
Parameters Length								
X	X	X	X	X	X	X	X-lsb	9
X	X	X	X	X	X	X	X	10
Operational Mode Parameters (Optional)								
•••								
X	X	X	X	X	X	X	X	9+L

3490 L = Length of Operational Mode Parameters.

- 3491
- 3492
- For the Mode Change Response Message, the value of the MID is 0x001C.
 - The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- 3493
- 3494
- 3495
- 3496

- 3497 • For the version of the Mode Change Response Message defined in this version of the
3498 Signaling Plan, the value of the Message Version field is 0x00.
- 3499 • The Operational Mode field shall contain the ID of the selected Operational Mode.
3500 For the format and values of these IDs, see the definition of Operational Mode IDs in
3501 Section 2.2.2.1. The high order bit of the Operational Mode field is placed in bit 8 of
3502 octet 6, and the low order bit is placed in bit 1 of octet 7. Note that Operational Mode
3503 shall be the mode offered in the Mode Change Request Message, unless either the
3504 terminal cannot support at least one of the offered Operational Mode Parameters
3505 Options, if included, or the ACL test fails. If the terminal does not support any of the
3506 Options offered or if the ACL test fails, the Operational Mode shall be the mode the
3507 terminal was executing when the Mode Change Request Message was received.
- 3508 • The Parameters Length field contains the actual length of the Operational Mode
3509 Parameters field (plus the length of the Parameters Length field itself), in octets. The
3510 value of the field shall be an unsigned binary integer with the high order bit being bit
3511 8 of the first octet of the field and the low order bit being bit 1 of the second octet of
3512 the field.
- 3513 • The Operational Mode Parameters shall contain the first Option on the Leader's
3514 Options List that is also supported by the Follower for the selected Operational Mode.
3515 The length, format, and contents of the Operational Mode Parameters are unique to
3516 each Operational Mode and are defined in Section 2.2.6 for each standard Operational
3517 Mode having a Parameters/Certificate Exchange. This field is optional and is not
3518 present unless Parameters are defined for the selected Operational Mode. If the
3519 terminal does not support any of the Options offered or if the ACL test fails, the
3520 Operational Mode Parameters shall contain the Option the terminal was executing
3521 when the Mode Change Request Message was received.

3522
3523

3524 **2.3.4 Two-Way Resync Processing**

3525
3526
3527
3528
3529
3530
3531
3532

This section specifies the signaling associated with Two-Way Resync processing. Only the Cryptosync Message is involved. The processing is shown in Figure 2.3-9. Two-Way Resync processing is invoked when a terminal in secure application traffic receives a local Two-Way Resync indication. A local Two-Way Resync indication is generated when a terminal detects that it has lost cryptographic synchronization with the far end or when selected manually by the user (e.g., by selecting "Secure" during secure application traffic).

3533

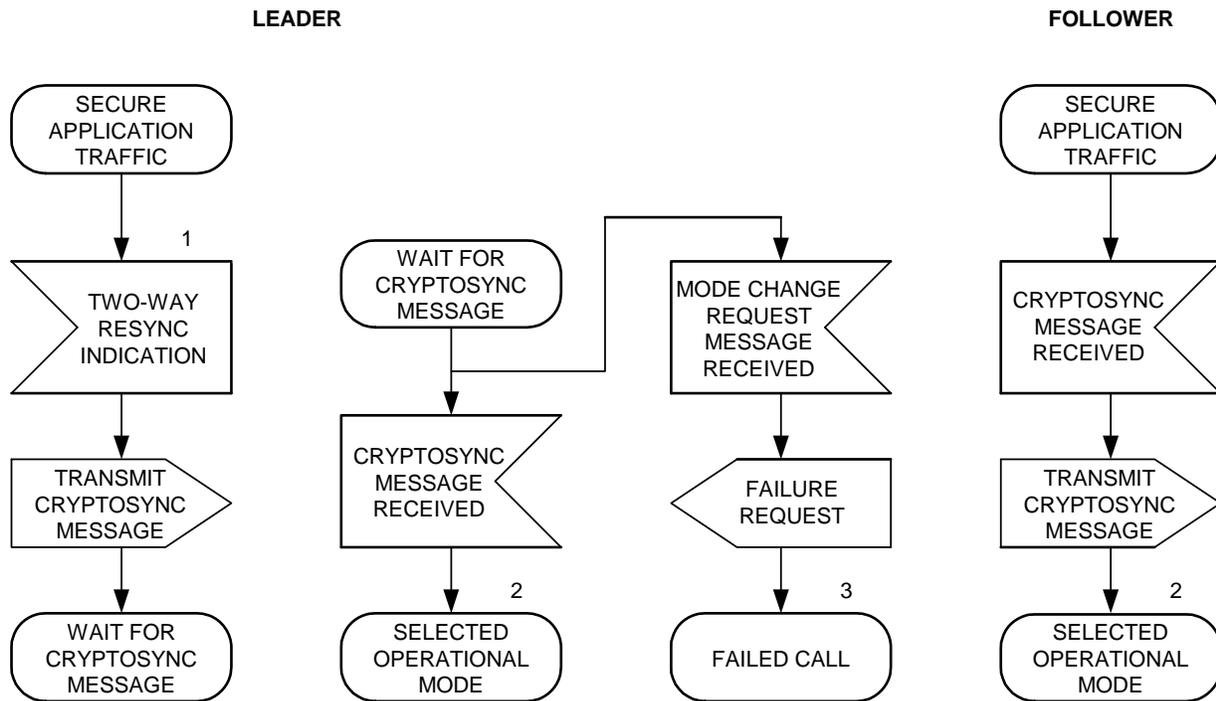


Figure 2.3-3

NOTES:

1. A Two-way Resync indication is generated when a terminal in a secure application, such as Secure MELP Voice, detects that it is cryptographically out-of-sync with the far end or when the user selects "Secure".
2. The terminal re-enters the same secure application from which it entered Two-Way Resync.
3. See Section 2.3.2.3.

3534

3535

Figure 2.3-9 Two-Way Resync Processing

3536

3537

3538

3539 During the secure call setup processing of Cryptosync Messages, Application IVs are exchanged
3540 together with Encrypted Packets that verify call setup negotiations were performed correctly.
3541 Since Two-Way Resync is not initiated until after secure call setup has been completed, i.e., both
3542 terminals have received Cryptosync Messages, the verification process is not repeated.

3543

3544 Upon receipt of a local Two-Way Resync indication, the terminal shall assume the role of
3545 Leader, format a Cryptosync Message as specified in Section 2.2.5, except that the optional
3546 Encrypted Packet shall not be included (i.e., the Packet Length contained in the Cryptosync
3547 Message is set to 0x0002, and the optional Encrypted Packet field is not present), transmit it to
3548 the far end, and wait for a Cryptosync Message. Upon receipt of the Cryptosync Message, the
3549 Leader shall initiate the secure application as specified in Section 3.

3550

3551 Upon receipt of a Cryptosync Message, a terminal that has not transmitted a Cryptosync
3552 Message shall assume the role of Follower, format a Cryptosync Message as specified in Section
3553 2.2.5 (but without the optional Encrypted Packet), transmit it to the far-end terminal, and initiate
3554 the secure application as specified in Section 3.

3555
3556 In the event of a glare condition, i.e., instead of receiving the expected Cryptosync Message, the
3557 Leader receives a Mode Change Request Message, the following signaling shall take place.
3558 Upon receipt of the Mode Change Request Message, the terminal shall initiate Failed Call
3559 processing as specified in Section 2.3.2.3 with the Information Code set to *Cryptosync/Mode*
3560 *Change glare*.

3561

3562
3563
3564
3565
3566
3567
3568
3569
3570
3571

2.4 SCIP Signaling Timeouts

Table 2.4-1 identifies the timeouts that have been defined for SCIP Signaling. It identifies the conditions under which each timeout occurs and the action to be taken. The initial values to be used for the timers are suggested values and are not requirements.

Table 2.4-1 SCIP Signaling Timeouts

Timeout (Identification and Conditions)	Starting the Timer	Stopping the Timer	Action to be Performed on Timeout
Transport Layer Timers			
<p>Retransmit Timeout</p> <p>Occurs when neither a REPORT ACK'ing all outstanding frames nor a REPORT NAK'ing some specific frames has been received. See Section 2.1.</p>	<p>Except after Transport Layer control messages (REPORT), this timer is started at the point where the EOM has been transmitted. It is initialized to 3 seconds at the beginning of initial call setup. It may then be adapted based on measured channel delay. See Section 2.1</p>	<p>This timer is stopped when a REPORT ACK'ing all outstanding frames is received. It is stopped/restarted when a frame group is (re)transmitted. See Section 2.1.</p>	<p>Upon expiration of this timer, frames that have not been acknowledged are retransmitted. See Section 2.1.</p>
Message Layer Timer			
<p>First Message Timeout</p> <p>Occurs when a recognizable SCIP message is not received from the far end. See Section 2.2.</p>	<p>This timer is started when the Capabilities Message is transmitted by an Initiator. It is set to 30 seconds to facilitate multiple retransmissions. See Section 2.2.</p>	<p>This timer is stopped when a valid Capabilities or Notification Message is received from the far end. See Section 2.2.2.</p>	<p>Upon expiration of this timer, the Failed Call Processing logic defined in Section 2.3.2.3 is executed.</p>

3572

3573
3574
3575

Table 2.4-1 SCIP Signaling Timeouts (Cont.)

Timeout (Identification and Conditions)	Starting the Timer	Stopping the Timer	Action to be Performed on Timeout
Application Timer (Note - This timer applies to full bandwidth applications. Such applications are not required to be implemented in a layered manner. If they are implemented in a layered manner, the location of the timer depends on the implementer's layering.)			
<p>Application Timeout</p> <p>Occurs when a START is not received from the far-end terminal. See Section 3.2.1.1, Application Timeout.</p>	<p>This timer is started when a terminal transmits the START pattern prior to receiving the pattern from the far end. It is initialized to 6 seconds at initial call setup. It may then be adapted based on measured channel delay. See Section 3.2.1.1.</p>	<p>This timer is stopped when a START is received. See Section 3.2.1.1.</p>	<p>Upon expiration of this timer, the START is retransmitted, preceded by an ESCAPE, and the Timer is restarted. See Section 3.2.1.1.</p>

3576

3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588

2.5 SCIP Signaling Constants

2.5.1 Source Definitions

Several fields include a Source ID in the upper 5 bits of the field. The high order bit of the Source ID maps to bit 8 of the first octet of the field, and the low order bit of the Source ID maps to bit 4 of the first octet. For all such fields the Source IDs defined in Table 2.5-1 shall be used.

Table 2.5-1 Source Definitions

Source ID	Source Definition
0x00	Defined in this Signaling Plan.
0x01	Defined by France.
0x03	Defined by General Dynamics.
0x05	Defined by L-3.
0x09	Defined by QUALCOMM.
0x12	Defined by the UK.

3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603

2.5.2 MIDs

The definitions of the standard MIDs are scattered throughout this document. They are gathered in Table 2.5-2 for convenience. Should a difference be found between this table and the other sections of the document, the other sections govern.

MIDs are 2 octets in length. The high order 5 bits of the first octet constitute a source for the MID. Currently identified sources are defined in Table 2.5-1. The next 11 bits constitute an MID number.

Table 2.5-2 MIDs

MID Values	MID Definition
0x0001	Reserved.
0x0002	Capabilities Message.
0x0003	Extended Keysets List Message.
0x0004	F(R) Message.
0x0008	Cryptosync Message.
0x0009	Multipoint Cryptosync Message.
0x000E	Notification Message.
0x0010	Parameters/Certificate Message.

3604
3605
3606

Table 2.5-2 MIDs (Cont.)

MID Values	MID Definition
0x001A	Mode Change Request Message.
0x001C	Mode Change Response Message.
0x0020	REPORT.
0x0023	Reserved for Tactical IWF (CONNECT).
0x0025	Reserved for Tactical IWF (DISCONNECT).
0x0040	Secure Reliable Transport Asynchronous Data Message.
0x0080	Reserved for compatibility with legacy terminals.
0x00E0	SCIP Rekey Message.

3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634

2.5.3 Miscellaneous SCIP Signaling Constants

Table 2.5-3 identifies the constants that have been defined for SCIP Signaling. It identifies both the values and uses of each constant.

2.5.3.1 ESCAPE

The ESCAPE sequence consists of two concatenated copies of the output of a 7-stage maximum length linear sequence generator padded with the first two bits of the sequence to give 256 bits. The coefficients of the generator polynomial are 203 (octal)¹.

2.5.3.2 Start of Message (SOM) and End of Message (EOM)

The SOM sequence is the output of a 6-stage maximum length linear sequence generator augmented with a leading zero-bit. The coefficients of the generator polynomial are 103 (octal). The EOM is the bit by bit complement of the SOM.

2.5.3.3 START and FILLER

The START sequence is the output of a 6-stage maximum length linear sequence generator augmented with a leading zero-bit. The coefficients of the generator polynomial are 141 (octal). FILLER is the bit by bit complement of the START sequence.

¹ The polynomial with coefficients 203 (octal) is $x^7 + x + 1$.

3635
3636
3637
3638
3639
3640
3641
3642
3643

2.5.3.4 Headers

The Header sequences are the outputs of 4-stage maximum length linear sequence generators, each augmented with a leading zero-bit. The coefficients of the generator polynomials are 31 (octal) for the Sync Management (SM) frame Header and 23 (octal) for the G.729D Encrypted Speech (ES) frame Header PN sequence. Note that the ES frame Header PN sequence is formed by rotating the generator output two bit positions to the right prior to adding the leading zero.

3644
3645
3646

Table 2.5-3 Miscellaneous SCIP Signaling Constants

Constant	Value(s)	Use(s) for the Constant
ESCAPE (256 bits long)	Generated PN Sequence: 0x FE041851E459D4FA 1C49B5BD8D2EE655 FC0830A3C8B3A9F4 38936B7B1A5DCCAB	For point-to-point operation, the transmitter uses the ESCAPE to break the receiver out of full bandwidth application traffic. If a terminal is in an application when the ESCAPE is received, it stops the application traffic and starts framing. (Details are in Section 2.1.)
	Message Table Value: [7F 20 18 8A 27 9A 2B 5F 38 92 AD BD B1 74 67 AA 3F 10 0C C5 13 CD 95 2F 1C C9 D6 DE 58 BA 33 D5]	
EOT (256 bits long)	Generated PN Sequence: 0x FE041851E459D4FA 1C49B5BD8D2EE655 FC0830A3C8B3A9F4 38936B7B1A5DCCAB	For multipoint operation, the transmitter uses the EOT, which is the same pattern as the ESCAPE, to indicate the end of multipoint traffic transmission. (Details are in Sections 5.1.5 and 5.2.)
	Message Table Value: [7F 20 18 8A 27 9A 2B 5F 38 92 AD BD B1 74 67 AA 3F 10 0C C5 13 CD 95 2F 1C C9 D6 DE 58 BA 33 D5]	
SOM (64 bits long)	Generated PN Sequence: 0x7E08629E8E4B766A	The message as a whole is delimited by a Start of Message (SOM) and an End of Message (EOM). Upon receipt of an SOM, the receiver will start message processing. (Sections 2.1 and 5.1)
	Message Table Value: [7E 10 46 79 71 D2 6E 56]	
EOM (64 bits long)	Generated PN Sequence: 0x81F79D6171B48995	The message as a whole is delimited by a Start of Message (SOM) and an End of Message (EOM). (Sections 2.1 and 5.1) (Note that this is the bit by bit complement of the SOM.)
	Message Table Value: [81 EF B9 86 8E 2D 91 A9]	

3647

3648
3649
3650

Table 2.5-3 Miscellaneous SCIP Signaling Constants (Cont.)

Constant	Value(s)	Use(s) for the Constant
START (64 bits long)	Generated PN Sequence: 0x7EACDDA4E2F28C20	Used to detect the start of full bandwidth application traffic. (Sections 3 and 5.1.4)
	Message Table Value: [7E 35 BB 25 47 4F 31 04]	
SM Header (16 bits long)	Generated PN Sequence: 0x7AC8	The first 16 bits of the Sync Management (SM) frame which is sent during full bandwidth application traffic. (Section 3)
	Message Table Value: [5E 13]	
Bit Complement of SM Header (16 bits long)	Generated PN Sequence: 0x8537	Replaces Header in Sync Management frames containing the first segment of the Partial Long Term Component. (Section 3)
	Message Table Value: [A1 EC]	
ES Header PN Sequence (16 bits long)	Generated PN Sequence: 0x5E26	The first 16 bits of the G.729D Encrypted Speech frame Header which is sent during Secure G.729D Voice application traffic. (Section 3.3)
	Message Table Value: [7A 64]	
FILLER (64 bits long)	Generated PN Sequence: 0x8153225B1D0D73DF	An integer number of 64-bit pattern repetitions are transmitted following the Cryptosync and Multipoint Cryptosync messages. (Sections 3 and 5.1.3) (Note that this is the bit by bit complement of the START.)
	Message Table Value: [81 CA 44 DA B8 B0 CE FB]	

3651
3652
3653
3654
3655
3656
3657

NOTES:

1. The Generated PN Sequence (read left to right and top to bottom if multiple lines) is specified in the bit order that the PN generator outputs the serial bit stream.
2. The Message Table Value is specified in the bit order that the PN sequences are represented (as octets) in the message format tables in this document.
3. The Message Table Value is passed, in ascending octet order, to the lower layers for transmission.

3658
3659 **3.0 SCIP USER APPLICATION SIGNALING – Point-to-Point Operation**
3660

3661 **3.1 SCIP User Applications**
3662

3663 Six user applications are currently defined for SCIP point-to-point implementations. They are:
3664 1) Secure 2400 bps MELP Voice - Blank and Burst, 2) Secure MELP Voice - Burst w/o Blank,
3665 3) Clear 2400 bps MELP Voice, 4) Secure G.729D Voice - Burst w/o Blank, 5) Secure Reliable
3666 Transport Asynchronous Data, and 6) Secure Best Effort Transport (BET) Asynchronous Data.
3667 [Note that a “reliable message transport” mechanism is also specified for the SCIP Electronic
3668 Rekey application – see Section 4.1.1.] The MELP Voice, Secure G.729D Voice, and Secure
3669 Best Effort Transport Asynchronous Data applications are full-bandwidth applications, since
3670 they are designed for use on connections where the information rate is equal, or approximately
3671 equal, to the available channel rate. Note that any of these applications may be used with a
3672 bearer service, such as IP, where the available transmission bandwidth may considerably exceed
3673 that of the application. It is not required to implement both the voice and data applications in all
3674 terminals. Voice-only and data-only products are allowed. Clear 2400 bps MELP Voice is
3675 optional in all cases. Detailed transmission formats for the SCIP user applications are specified
3676 in Sections 3.3 (voice) and 3.4 (data). The signaling in this Section is shown octet aligned.
3677 However it may be carried on networks that do not preserve octet alignment. Therefore, the
3678 SCIP receiver shall be capable of recovering and processing the SCIP signaling shown herein
3679 even if it is not octet aligned when it is received.
3680

3681 **3.2 Application Start-up/Restart Signaling**
3682

3683 **3.2.1 Full Bandwidth Applications**
3684

3685 Full-bandwidth applications (e.g., Secure MELP Voice, Secure G.729D Voice, and Secure Best
3686 Effort Transport Asynchronous Data) are required to bypass the Transport Layer functionality
3687 when they are invoked. This shall be accomplished as specified in the following paragraphs.
3688

3689 There are three cases of full-bandwidth application start-up/restart signaling: the case where a
3690 Cryptosync message exchange has occurred, the case of clear voice start-up, and the case of a
3691 restart following an interruption where only Notification and REPORT messages have been
3692 exchanged. In addition, there is a full bandwidth Application Timeout (specified in Section
3693 3.2.1.1) associated with all three cases.
3694

3695 For start-up/restart after a Cryptosync exchange, a terminal is ready to transition to full
3696 bandwidth application signaling when acknowledgments have been transmitted and received for
3697 all frames of the Cryptosync messages. For this case, the terminal shall transmit an integer
3698 number of repetitions of FILLER not less than 100 milliseconds duration. The START pattern
3699 shall follow FILLER as soon as any frames queued at the Transport Layer are transmitted and,
3700 except for REPORT messages, are acknowledged. For start-up of the Clear MELP Voice
3701 application, a terminal is ready to transition to full bandwidth application signaling when the
3702 final call setup message has been transmitted and acknowledged and the user has enabled
3703

3704 nonsecure operation. For this case, the terminal shall transmit the START pattern, without
3705 FILLER, as soon as any frames queued at the Transport Layer are transmitted and, except for
3706 REPORT messages, are acknowledged. For restart after full bandwidth traffic has been
3707 interrupted by the transmission of a Notification (Secure Dial, or Attention) or a REPORT
3708 message, the START pattern shall be transmitted, without FILLER, as soon as all frames queued
3709 at the Transport Layer are transmitted and, except for REPORT messages, are acknowledged.

3710
3711 Transmission of the START pattern shall be followed by full bandwidth traffic when it is
3712 available. The START pattern shall be transmitted even if no full bandwidth traffic is available
3713 for transmission (because of the Application Timeout – see Section 3.2.1.1). Upon receipt of a
3714 START pattern, a terminal shall begin looking for full bandwidth traffic. The format and length
3715 of FILLER and START patterns are defined in Section 2.5.3. Full bandwidth traffic
3716 transmission formats include Secure MELP Voice (both Blank and Burst and Burst w/o Blank –
3717 Sections 3.3.1.1 and 3.3.1.2, respectively), Clear MELP Voice (Section 3.3.1.3), Secure G.729D
3718 Voice (Section 3.3.2), and Secure Best Effort Transport Asynchronous Data (Section 3.4.2).

3719
3720 Note that start-up/restart applies to secure call setup, Two-Way Resync and Mode Change, and
3721 includes the case where both Cryptosync and Notification Messages are transmitted during a
3722 break in full bandwidth traffic. This case is illustrated in Figure 2.2-1 for secure call setup, in
3723 Figure 2.3-1(d) for Mode Change, and in Figure 2.3-1(e) for Two-Way Resync. Restart after
3724 transmission of a Notification or a REPORT message is illustrated in Figure 2.1-1(c) at the
3725 Transport Layer and in Figure 2.3-1(c) at the Message Layer.

3726
3727

3728 **3.2.1.1 Application Timeout**

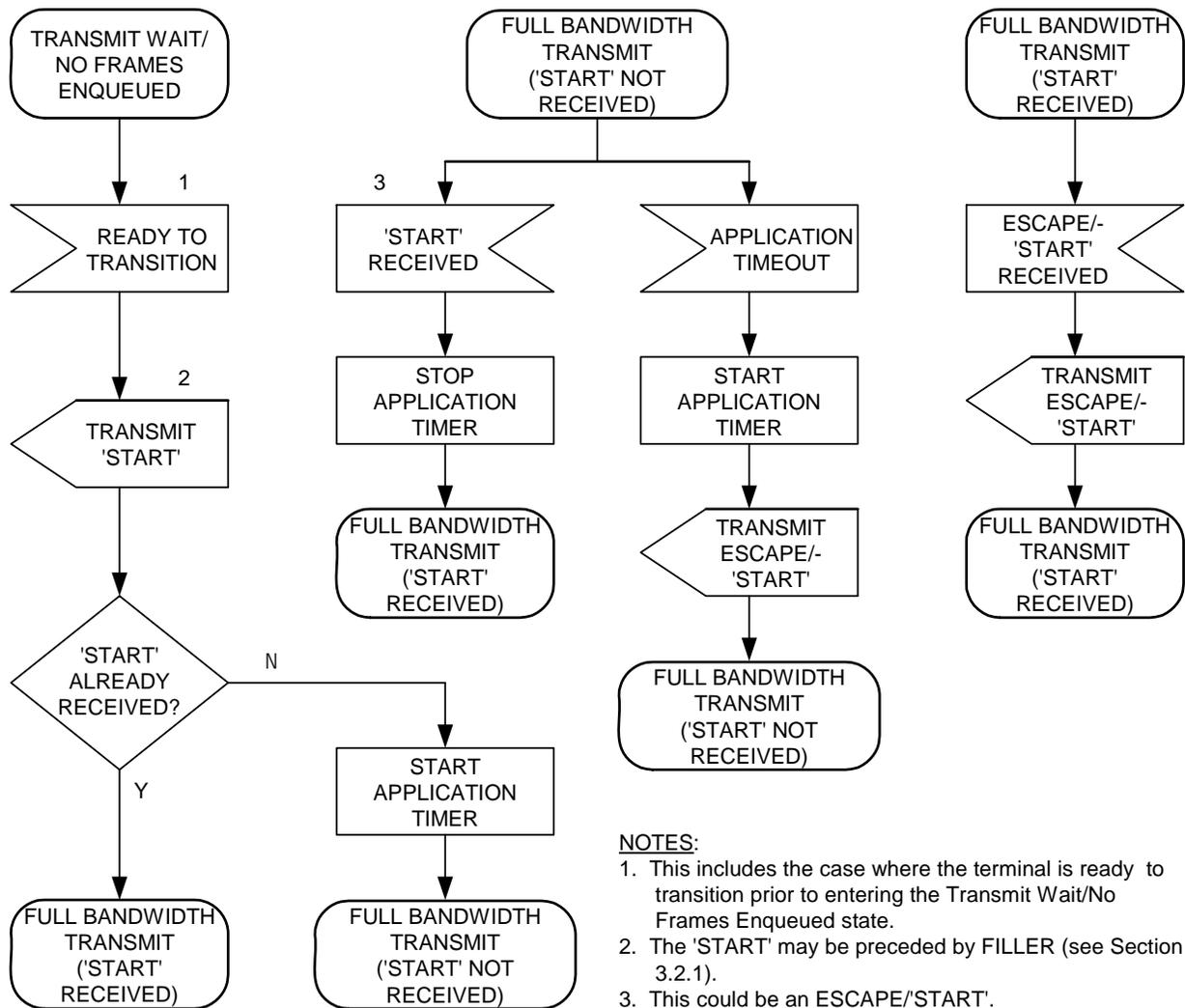
3729

3730 As specified above, a terminal transmits the START pattern when it is ready to start or restart
3731 full bandwidth traffic. An Application Timeout shall be utilized to ensure that both terminals
3732 transition to full bandwidth traffic. A suggested initial value for the Application Timeout is
3733 given in Table 2.4-1. Processing associated with the Application Timeout is shown in Figure
3734 3.2-1.

3735

3736

3737



3738

3739

3740

3741

3742

3743

3744

3745

3746

3747

3748

3749

3750

3751

3752

3753

3754

Figure 3.2-1 Application Timeout Processing

Two processing substates are associated with the Application Timeout. These are the full bandwidth transmit (START not received) substate and the full bandwidth transmit (START received) substate.

When a terminal has completed transmitting a START pattern, if a START pattern has not yet been received from the far end, it shall start an Application Timer and enter the full bandwidth transmit (START not received) substate. If the terminal has already received a START pattern, it shall enter the full bandwidth transmit (START received) substate without starting the Application Timer.

When a START pattern is received while the Application Timer is running, the terminal shall stop the Timer and enter the full bandwidth transmit (START received) substate.

3755
3756 If an Application Timeout occurs before a START pattern is received, the terminal shall transmit
3757 the ESCAPE/START (the ESCAPE pattern followed immediately by the START pattern),
3758 restart the Application Timer, and remain in the full bandwidth transmit (START not received)
3759 substate.

3760
3761 When the Application Timer has been stopped, a terminal may, under exception conditions,
3762 receive another ESCAPE/START. If this occurs, the terminal shall transmit another
3763 ESCAPE/START and continue in the full bandwidth transmit (START received) substate. The
3764 Application Timer is not restarted.
3765

Editor's Note: It is important that the Application Timeout value always be greater than the round-trip path delay; otherwise, the terminals may fall into continuous 'ping-pong' retransmissions of the ESCAPE/START patterns. The recommended initial value for the Application Timeout in Table 2.4-1 should be adequate for most, if not all, connections.

3766

3767

3768 **3.2.2 Reliable Transport Applications**

3769

3770 Applications defined as reliable transport (e.g., Secure Reliable Transport Asynchronous Data)
3771 are required to retain the Transport Layer functionality after completing call setup or Mode
3772 Change signaling. This is accomplished as follows.

3773

3774 When a reliable transport application has been chosen in the initial SCIP call setup or Mode
3775 Change signaling, the application shall begin when the following conditions have been met:

3776

3777 (a) the final SCIP call setup or control message has been transmitted,

3778

3779 (b) there are no remaining outstanding Transport Layer frames in the final SCIP call setup or
3780 control message received from the far end terminal, and

3781

3782 (c) all outstanding Transport Layer frames have been acknowledged for the last message
3783 transmitted.

3784

3785 When these conditions have been met, the application shall begin transmitting reliable transport
3786 application messages (e.g., Secure Reliable Transport Asynchronous Data messages), when they
3787 are available, without terminating the Transport Layer. FILLER and START patterns shall not
3788 be sent following call setup or control signaling, since the Transport Layer functionality is not
3789 being bypassed. Therefore, there is no Application Timeout. Likewise, ESCAPE shall not be
3790 sent when transitioning back to call control signaling.

3791

3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807

3.3 Secure Voice Applications

3.3.1 Secure MELP Voice

Two variants of Secure MELP Voice are defined. These are 2400 bps Blank and Burst and Burst w/o Blank. Both variants utilize a superframe structure consisting of a Sync Management frame followed by MELP frames. In 2400 bps Blank and Burst the first MELP frame of a superframe is replaced with a Sync Management frame. In Burst w/o Blank, a Sync Management frame is inserted prior to the first MELP frame. (Thus in Blank and Burst, the superframe is 24 frames in length, while in Burst w/o Blank, the superframe is 25 frames in length.) All instances of the term MELP in this document may refer to either MELP as defined in MIL-STD-3005 or 2400 bps MELPe as defined in NATO STANAG 4591. Although MELPe is the preferred voice coder, the bit streams for both specifications are identical; therefore, full compatibility is maintained.

Editor's Note: Burst w/o Blank MELP requires a channel capacity greater than 2400 bps.

3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824

3.3.1.1 Secure 2400 bps MELP Voice – Blank and Burst

For Secure 2400 bps Blank and Burst MELP Voice, a Sync Management frame is substituted periodically for a vocoder frame. The vocoder frame that would normally have been transmitted during the Sync Management frame transmission interval is discarded. The Sync Management frame contains information that allows late-entry cryptographic synchronization as well as cryptographic synchronization maintenance.

Secure 2400 bps Blank and Burst MELP Voice shall be transmitted in a "superframe" consisting of a 54-bit Sync Management frame followed by 23 54-bit MELP vocoder frames, except when shortened by DTX action (see Section 3.3.1.4) or by the transmission of an ESCAPE to return to framed operation. To provide octet alignment on networks that require it, two, four, or six zero bits of padding may be postpended if the length of a shortened superframe is not a multiple of eight bits.

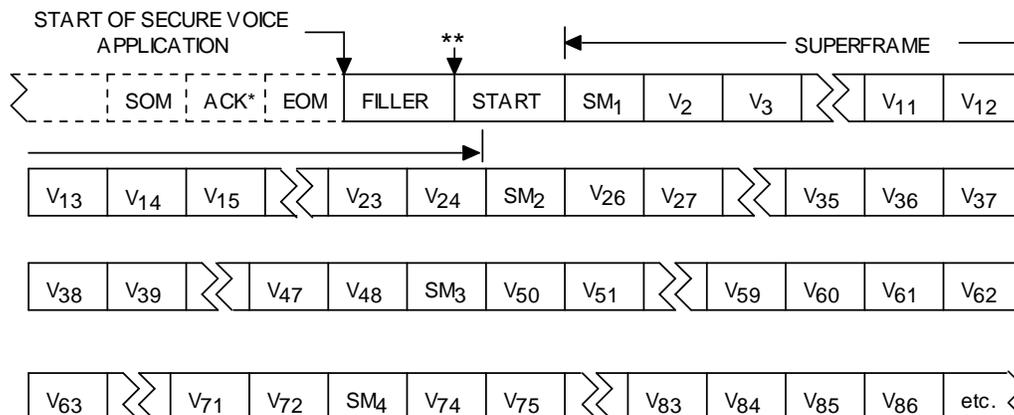
Editor's Note: While the superframe size is currently defined to be 24 frames for Blank and Burst, if problems are found during field testing this may be changed and/or may be made negotiable.

3825
3826
3827
3828
3829
3830
3831
3832
3833

An example of Secure 2400 bps Blank and Burst MELP Voice transmission is shown in Figure 3.3-1. Note that the superframe always begins with a Sync Management frame to facilitate vocoder frame synchronization following a silence interval in implementations utilizing Voice Activity Detection. Note also that except for the first superframe following a gap in speech, the first vocoder frame shall be discarded (blanked) and replaced by the Sync Management frame. (See Appendix B.6 for the case of the first superframe following a gap in speech.) In all cases, however, the first MELP frame actually transmitted in a superframe is encrypted using the second half of the first state vector value for that superframe.

3834
3835
3836
3837
3838
3839
3840

The contents of the 54-bit MELP vocoder frame, representing 22.5 msec. of speech, shall be as specified in MIL-STD-3005 or NATO STANAG 4591. In particular, bit 1 is as defined therein. The alternating 1/0 sync bit in the first MELP vocoder frame transmitted may have either value, and the receiver must be prepared to accept either value.



NOTES:
SM = Sync Management Frame
V = MELP Vocoder Frame
* = ACKed via Report Message
** = Application re-entry point after Notification Message processing

3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852

Figure 3.3-1 Secure MELP Voice Transmission Format – Blank and Burst

3.3.1.1.1 Sync Management Frame

The Sync Management frame shall be transmitted as the first frame of each Secure 2400 bps Blank and Burst MELP Voice superframe. Its format shall be as shown in Figure 3.3-2. The Sync Management frame is not encrypted.



3853
3854
3855
3856
3857
3858
3859
3860
3861

Figure 3.3-2 Sync Management Frame Format – Blank and Burst

The contents of the Secure 2400 bps Blank and Burst MELP Voice Sync Management frame are shown in Table 3.3-1. The fixed 16-bit Header shall be the 16-bit PN sequence defined in Section 2.5.3. The bits of the Header shall be inverted in a Sync Management frame that contains the first segment of the Long Component. The Partial Long Component and the Short

3862 Component refer to encryption parameters that are specified in SCIP-23x. The bit transmission
3863 order for the Sync Management frame is shown in Table 3.3-2.

3864
3865 The CRC is an 8-bit frame check sequence that protects the Partial Long Component and Short
3866 Component fields. Its generator polynomial is $P(x) = x^8 + x^6 + x + 1$. The CRC shall be
3867 computed as follows. Let $S(x)$ be the polynomial representing the 30 bits of the Sync
3868 Management frame beginning with the most significant bit of the Partial Long Component and
3869 extending, in the order that the bits are transmitted, through the least significant bit of the Short
3870 Component. The most significant bit of the Partial Long Component is the coefficient of the
3871 highest degree term of $S(x)$. The transmitted CRC checksum, $F(x)$, shall be the ones complement
3872 of the remainder of $(x^8S(x) + x^{30}(x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1))/P(x)$. Note that multiplying
3873 $S(x)$ by x^8 is equivalent to shifting $S(x)$ eight places to provide the space for the 8-bit CRC
3874 checksum, and adding $x^{30}(x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$ to $x^8S(x)$ is equivalent to inverting
3875 the first eight bits of $S(x)$. $F(x)$ is then added to $x^8S(x)$ forming the 38-bit Sync Management
3876 frame, exclusive of the Header. The coefficient of the x^7 term of $F(x)$ shall be transmitted
3877 immediately following the least significant bit of the Short Component (see Table 3.3-2).

3878
Editor's Note: Inverting the first eight bits of $S(x)$ can also be accomplished in a shift register implementation by setting the register to all “ones” initially. This permits the receiver to detect erroneous addition or deletion of zero bits at the leading end of $S(x)$. Complementing the remainder permits the receiver to detect addition or deletion of trailing zeros that may appear as a result of errors. At the receiver, the shift register is again set to all “ones” initially, and the CRC is computed over the received $S(x)$. If the computed and received CRC are the same value, there are no errors.

3879
3880
3881
3882

Table 3.3-1 Sync Management Frame Contents – Blank and Burst

Field	Length (bits)
Header (PN Sequence)	16
Partial Long Component	16
Short Component	14
CRC	8

3883
3884

Editor's Note: The SCIP cryptography and the use of the Long and Short Components transmitted in the Sync Management frame are defined in SCIP-23x.

3885

3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902

3.3.1.1.2 Encryption and Transmission Ordering

MELP vocoder data is generated in 54-bit frames. Vocoder frames may be padded to 56 bits (to provide octet alignment if required) prior to encryption, but only the output bits corresponding to the first 54 bits of the input shall be transmitted. Data ordering for encrypting MELP vocoder data is specified in SCIP-230 or SCIP-231, Section 4.1.1.1; or SCIP-232, Section 4.1.1. Note that the vocoder frames that are deleted to allow for insertion of Sync Management frames shall be deleted following encryption.

Encrypted MELP vocoder frames shall have Sync Management frames inserted (in place of the deleted vocoder frames) and shall be formatted into superframes as shown in Table 3.3-2. The superframes shall then be passed, in ascending octet order beginning with the first octet of the Header, to the lower layers for transmission. If the length of a shortened superframe is not a multiple of eight bits, sufficient padding bits may be added to make the resulting padded superframe a multiple of eight bits, if the underlying transport service requires octet alignment.

3903
3904
3905

Table 3.3-2 Secure MELP Transmission Bit Ordering – Blank and Burst

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Header (PN Sequence)								
0	1	0	1	1	1	1	0	1
0	0	0	1	0	0	1	1	2
Partial Long Component								
b8	b9	b10	b11	b12	b13	b14	b15-msb	3
b0-lsb	b1	b2	b3	b4	b5	b6	b7	4
Short Component								
b6	b7	b8	b9	b10	b11	b12	b13-msb	5
CRC								
b6	b7-msb	b0-lsb	b1	b2	b3	b4	b5	6
MELP Frame 2								
b2	b1	b0-lsb	b1	b2	b3	b4	b5	7
b10	b9	b8	b7	b6	b5	b4	b3	8
...								
MELP Frame 3								
b4	b3	b2	b1	b54	b53	b52	b51	14
...								
...								
...								
MELP Frame 24								
b6	b5	b4	b3	b2	b1	b54	b53	156
...								
b54	b53	b52	b51	b50	b49	b48	b47	162

3906

3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918

3.3.1.2 Secure MELP Voice – Burst w/o Blank

For Secure Burst w/o Blank MELP Voice, a Sync Management frame is inserted periodically between vocoder frames. The Sync Management frame contains information that allows late-entry cryptographic synchronization as well as cryptographic synchronization maintenance.

Secure Burst w/o Blank MELP Voice shall be transmitted in a "superframe" consisting of a 56-bit Sync Management frame followed by 24 56-bit MELP frames (54 MELP vocoder bits plus two padding bits), except when shortened by DTX action (see Section 3.3.1.4) or by the transmission of an ESCAPE to return to framed operation.

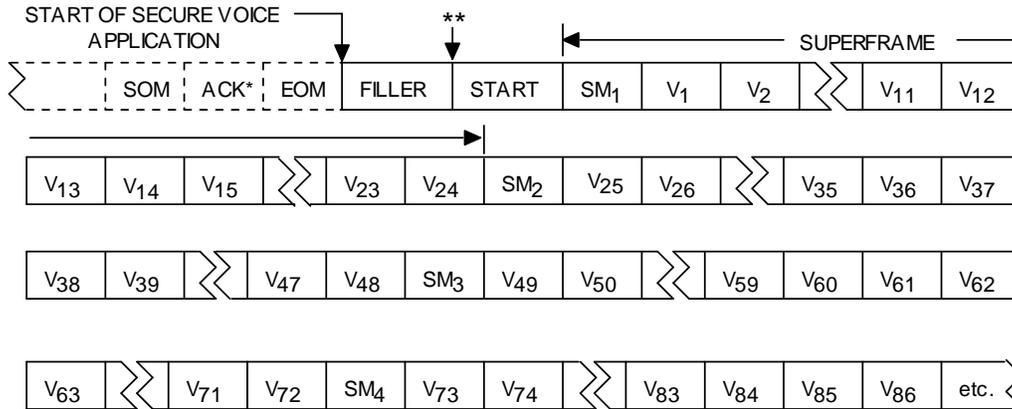
Editor's Note: While the superframe size is currently defined to be 25 frames for Burst w/o Blank, if problems are found during field testing this may be changed and/or may be made negotiable.

3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931

An example of Secure Burst w/o Blank MELP Voice transmission is shown in Figure 3.3-3. Note that the superframe always begins with a Sync Management frame to facilitate vocoder frame synchronization following a silence interval in implementations utilizing Voice Activity Detection.

The contents of the 54-bit MELP vocoder output frame, representing 22.5 msec. of speech, shall be as specified in MIL-STD-3005 or NATO STANAG 4591. In particular, bit 1 is as defined therein. The alternating 1/0 sync bit in the first MELP frame transmitted may have either value, and the receiver must be prepared to accept either value. Two padding bits shall be added to the vocoder output as specified in SCIP-230 or SCIP-231, Section 4.1.1.1.1; or SCIP-232, Section 4.1.1.1.

3932



NOTES:

- SM = Sync Management Frame
- V = MELP Vocoder Frame
- * = ACKed via Report Message
- ** = Application re-entry point after Notification Message processing.

3933

3934

Figure 3.3-3 Secure MELP Voice Transmission Format – Burst w/o Blank

3936

3.3.1.2.1 Sync Management Frame

3937

3938

The Sync Management frame shall be transmitted as the first frame of the Secure Burst w/o Blank MELP Voice superframe. Its format shall be as shown in Figure 3.3-4.

3940

3941

3942

3943



3944

3945

Figure 3.3-4 Sync Management Frame Format – Burst w/o Blank

3946

3947

3948

The contents of the Secure Burst w/o Blank MELP Voice Sync Management frame are shown in Table 3.3-3. The Header, Partial Long Component, and Short Component shall be the same as that specified for Secure Blank and Burst MELP Voice (see Section 3.3.1.1.1). The "PLC Index" field consists of two bits that shall be set as defined in SCIP-23x.

3949

3950

3951

3952

3953

The CRC protects the Partial Long Component, Short Component, and PLC Index fields. It shall be computed over the 32 bits of the Sync Management frame beginning with the most significant bit of the Partial Long Component and extending, in the order the bits are transmitted, through the least significant bit of the PLC Index. Except for the additional field covered (PLC Index), the transmitted CRC checksum shall be calculated and transmitted as defined in Section 3.3.1.1.1.

3954

3955

3956

3957

3958

3959

3960
3961
3962
3963
3964
3965

The bit transmission order for the Sync Management frame is shown in Table 3.3-4.

Table 3.3-3 Sync Management Frame Contents – Burst w/o Blank

Field	Length (bits)
Header (PN Sequence)	16
Partial Long Component	16
Short Component	14
PLC Index	2
CRC	8

3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977

3.3.1.2.2 Encryption and Transmission Ordering

MELP vocoder data is generated in 54-bit frames. Vocoder output frames shall be padded to 56 bits (to provide octet alignment). (The padding bits shall be removed from received MELP frames prior to passing to the vocoder.) Data ordering for encrypting MELP vocoder data is specified in SCIP-230 or SCIP-231, Section 4.1.1.1.1; or SCIP-232, Section 4.1.1.1.

Encrypted MELP frames shall have Sync Management frames inserted and shall be formatted into superframes as shown in Table 3.3-4. The superframes shall then be passed, in ascending octet order beginning with the first octet of the Header, to the lower layers for transmission.

3978
3979
3980

Table 3.3-4 Secure MELP Transmission Bit Ordering – Burst w/o Blank

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Header (PN Sequence)								
0	1	0	1	1	1	1	0	1
0	0	0	1	0	0	1	1	2
Partial Long Component								
b8	b9	b10	b11	b12	b13	b14	b15-msb	3
b0-lsb	b1	b2	b3	b4	b5	b6	b7	4
Short Component								
b6	b7	b8	b9	b10	b11	b12	b13-msb	5
PLC Index								
b0-lsb	b1-msb	b0-lsb	b1	b2	b3	b4	b5	6
CRC								
b0-lsb	b1	b2	b3	b4	b5	b6	b7-msb	7
MELP Frame 1								
b8	b7	b6	b5	b4	b3	b2	b1	8
...								
Padding Bits								
X	X	b54	b53	b52	b51	b50	b49	14
MELP Frame 2								
b8	b7	b6	b5	b4	b3	b2	b1	15
...								
Padding Bits								
X	X	b54	b53	b52	b51	b50	b49	21
...								
...								
MELP Frame 24								
b8	b7	b6	b5	b4	b3	b2	b1	169
...								
Padding Bits								
X	X	b54	b53	b52	b51	b50	b49	175

3981

3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995

3.3.1.3 Clear MELP Voice – Blank and Burst

Signaling for Clear MELP Voice, when it is supported, will be in a "Blank and Burst" format. This means that a Sync Management frame is substituted periodically for a vocoder frame. The vocoder frame that would normally have been transmitted during the Sync Management frame transmission interval is discarded.

Clear MELP Voice shall be transmitted in a "superframe" consisting of a 54-bit Sync Management frame followed by 23 54-bit MELP vocoder frames, except when shortened by DTX action (see Section 3.3.1.4) or by the transmission of an ESCAPE to return to framed operation. To provide octet alignment on networks that require it, two, four, or six zero bits of padding may be postpended if the length of a shortened superframe is not a multiple of eight bits.

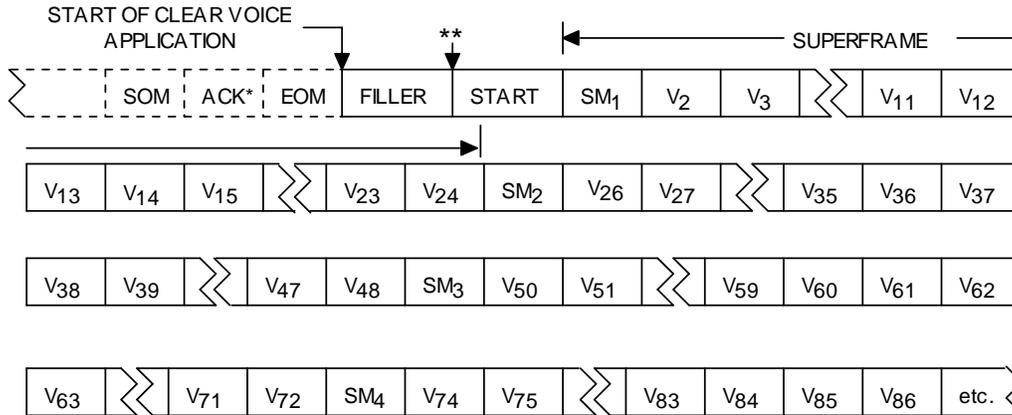
<p>Editor's Note: While the superframe size is currently defined to be 24 frames for Clear MELP Voice, if problems are found during field testing this may be changed and/or may be made negotiable.</p>

3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008

An example of Clear MELP Voice transmission is shown in Figure 3.3-5. Note that the superframe begins with a Sync Management frame to facilitate vocoder frame synchronization following a silence interval. Note also that except for the first superframe following a gap in speech, the first vocoder frame shall be discarded (blanked) and replaced by the Sync Management frame. (See Appendix B.6 for the case of the first superframe following a gap in speech.)

The contents of the 54-bit MELP vocoder frame, representing 22.5 msec. of speech, shall be as specified in MIL-STD-3005 or NATO STANAG 4591. In particular, bit 1 is as defined therein. The alternating 1/0 sync bit in the first MELP vocoder frame transmitted may have either value, and the receiver must be prepared to accept either value.

4009



NOTES:
SM = Sync Management Frame
V = MELP Vocoder Frame
* = ACKed via Report Message
** = Application re-entry point after Notification Message processing

4010

4011

Figure 3.3-5 Clear MELP Voice Transmission Format

4012

4013

4014

3.3.1.3.1 Sync Management Frame

4015

4016

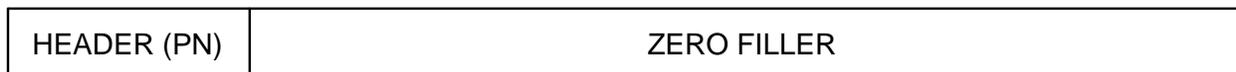
The Sync Management frame shall be transmitted as the first frame of each Clear MELP Voice superframe. Its format shall be as shown in Figure 3.3-6.

4017

4018

4019

4020



4021

4022

Figure 3.3-6 Clear MELP Voice Sync Management Frame Format

4023

4024

4025

The contents of the Clear MELP Voice Sync Management frame are shown in Table 3.3-5. The fixed 16-bit Header shall be the 16-bit PN sequence defined in Table 2.5-3. Following the Header will be 38 bits of filler, each of which is set to zero. The bit transmission order for the Sync Management frame is shown in Table 3.3-6.

4026

4027

4028

4029

4030

4031
4032
4033

Table 3.3-5 Clear MELP Voice Sync Management Frame Contents

Field	Length (bits)
Header (PN Sequence)	16
Zero Filler	38

4034
4035
4036
4037

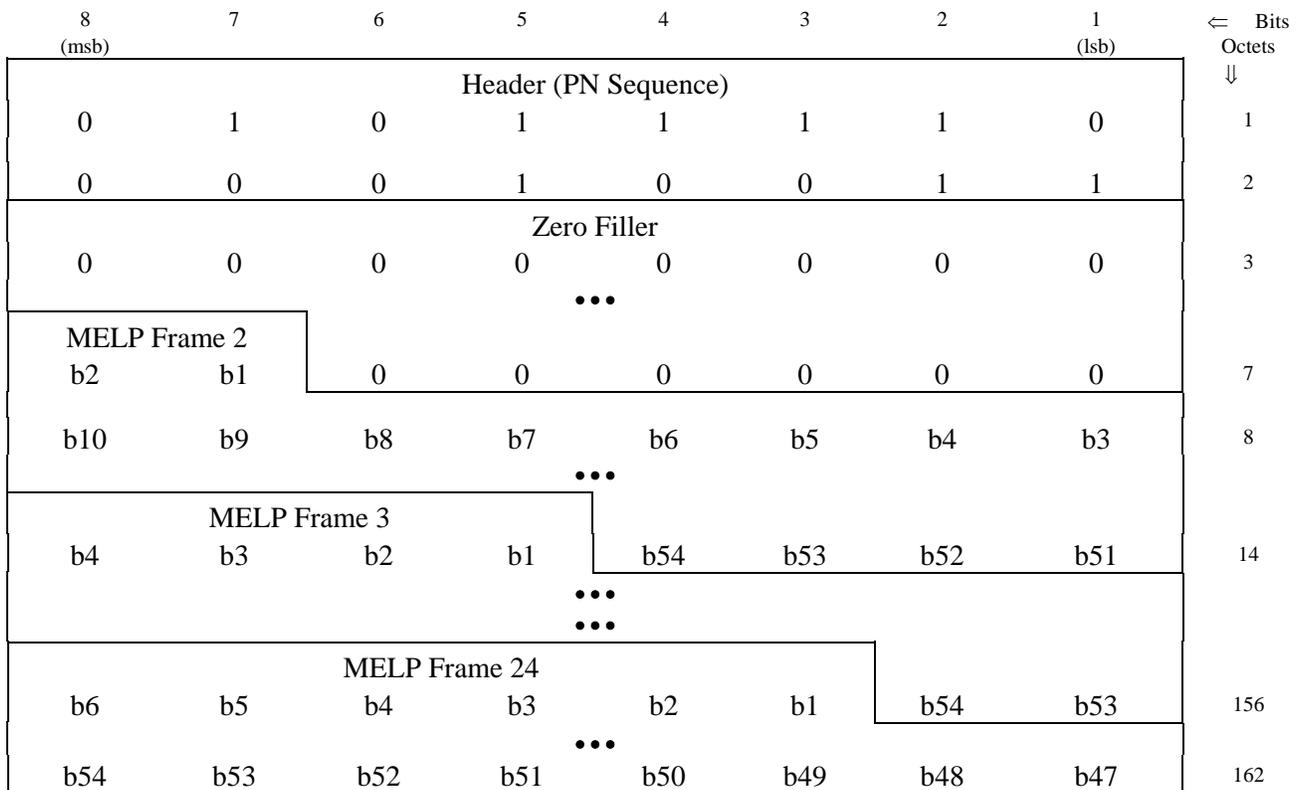
3.3.1.3.2 Transmission Ordering

4038
4039
4040
4041
4042
4043
4044
4045
4046

MELP vocoder data is generated in 54-bit frames. Clear MELP Voice vocoder frames shall have Sync Management frames inserted in place of the deleted vocoder frames, and shall be formatted into superframes as shown in Table 3.3-6. The superframes shall then be passed, in ascending octet order beginning with the first octet of the Header, to the lower layers for transmission. If the length of a shortened superframe is not a multiple of eight bits, sufficient padding bits may be added to make the resulting padded superframe a multiple of eight bits, if the underlying transport service requires octet alignment.

4047
4048

Table 3.3-6 Clear MELP Voice Transmission Bit Ordering – Blank and Burst



4049

4050
4051
4052
4053
4054
4055
4056
4057

3.3.1.4 Voice Activity Factor Processing

Two options are defined for SCIP MELP Voice. These are Discontinuous Voice Transmission (DTX) and Force Continuous Transmission (FCT), as described below. Where terminals have implemented both DTX and FCT, Secure Voice Options are provided in Section 2.2.6.2 for negotiating which one to use.

Editor's Note: A cellular phone may support Discontinuous Voice Transmission so that it will cease transmitting when the user stops speaking. However, for security reasons, it may be required that the phone be set for Force Continuous Transmission. All current (as of 7/02) Secure MELP Voice implementations are FCT only.

4058
4059
4060
4061

3.3.1.4.1 Discontinuous Voice Transmission

Discontinuous voice transmission (DTX) is specified in Appendix B. This Section only addresses the signaling associated with it.

4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080

During DTX operation, when voice activity is initially detected a superframe shall be formatted and transmitted. For as long as voice is detected, full length superframes (defined in the sections that define the individual applications) shall be continuously transmitted. When silence is detected, two or more Grace Period frames (defined in Appendix B.3) shall be transmitted in place of the corresponding number of MELP frames. Transmission shall then cease for at least n MELP frames, where n is the Minimum Blank Period (defined in Appendix B.4). The final superframe before transmission ceases may be shorter than a full length superframe; it contains a Sync Management frame, zero or more MELP frames, and one or more Grace Period frames. After a period of silence, when voice is again detected, transmission of MELP frames is restarted. The first frame transmitted following a gap shall be a Sync Management frame, regardless of the duration of the gap. This Sync Management frame shall contain the next value in the cyclic rotation of Partial Long Term Components that would normally follow the value transmitted in the last Sync Management frame prior to the gap. The crypto is not flywheeled during gaps in voice transmission. MELP frame transmission continues with full length superframes until silence is again detected.

4081
4082

3.3.1.4.2 Force Continuous Transmission

4083
4084
4085
4086

Force Continuous Transmission (FCT) applies to both Secure MELP Voice and Clear MELP Voice applications.

4087
4088
4089
4090

During FCT operation, full length superframes shall be transmitted continuously between the START (see Section 3.2) and the ESCAPE (see Section 2.1.4). The MELP vocoder is run continuously, and all frames that are generated (excluding blanked frames) are transmitted.

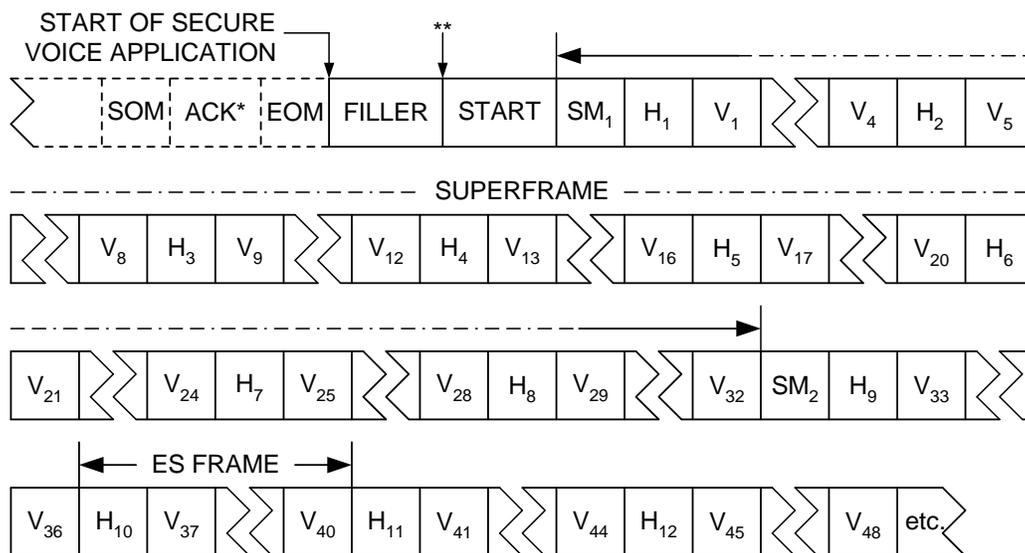
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111

3.3.2 Secure G.729D Voice

Secure G.729D Voice is transmitted in a Burst w/o Blank superframe structure with a Sync Management (SM) frame and Encrypted Speech (ES) frame Headers inserted periodically between vocoder frames. The Sync Management frame contains information that allows late-entry cryptographic synchronization as well as cryptographic synchronization maintenance. The Encrypted Speech frame Headers allow resynchronization between Sync Management frames. The 6400 bps vocoder output plus the framing requires a channel capacity of at least 7200 bps.

Secure G.729D Voice shall be transmitted in a "superframe" consisting of a 64-bit Sync Management frame followed by up to eight Encrypted Speech frames. Each Encrypted Speech frame shall consist of a 24-bit Header followed by four 64-bit G.729D Voice frames.

An example of Secure G.729D Voice transmission highlighting the superframe and Encrypted Speech frame structure is shown in Figure 3.3-7. Detailed breakouts of a superframe and an Encrypted Speech frame are shown in Figure 3.3-8. Note that the superframe always begins with a Sync Management frame to facilitate frame synchronization following an interruption, e.g., a period of framed operation.



- NOTES:**
SM = Sync Management Frame
H = Encrypted Speech frame Header
V = G.729D Vocoder Frame
* = ACKed via Report Message
** = Application re-entry point after Notification Message processing

4112
4113
4114
4115
4116

Figure 3.3-7 Secure G.729D Voice Transmission

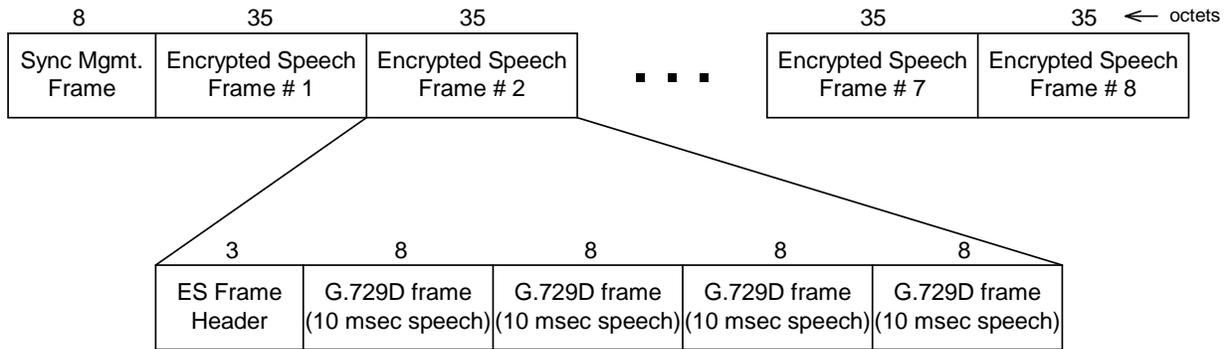


Figure 3.3-8 Secure G.729D Voice Superframe Details

Full-length superframes shall be transmitted except when shortened by DTX action (see Section 3.3.2.4) or by the transmission of an ESCAPE to return to framed operation. A shortened Secure G.729D Voice superframe (resulting from an interruption) shall be terminated only at the end of an Encrypted Speech frame. An example of a Secure G.729D Voice superframe interruption and the subsequent restart of Secure G.729D Voice transmission is shown in Figure 3.3-9. Note that this figure shows the case where the interruption does not include the transmission of a Cryptosync message. If a Cryptosync message is transmitted, e.g., in a Two-Way Resync, the START is preceded by FILLER, and the Partial Short Component (PSC) is set as specified in SCIP-230 or SCIP-231, Section 4.1.1.2.1; or SCIP-232, Section 4.1.2.1.

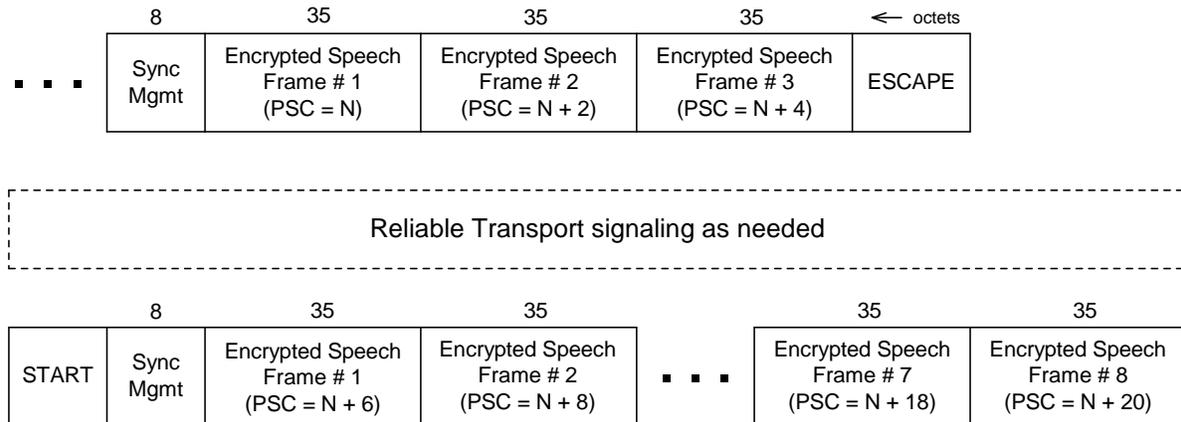


Figure 3.3-9 Secure G.729D Voice Escape and Return Example (No Cryptosync)

4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148

3.3.2.1 Secure G.729D Voice Frame

The 64-bit G.729D vocoder output frame, representing 10 msec. of speech, shall be as specified in ITU-T Recommendation G.729 Annex D. The frame parameters and their lengths, as specified in Table 8/G.729 (Recommendation G.729), but with the specific parameters and lengths as modified by Table D.1/G.729 (Recommendation G.729 Annex D), are shown in Table 3.3-7.

Table 3.3-7 Secure G.729D Voice Frame Parameters

Parameter	Number of Bits
L0	1
L1	7
L2	5
L3	5
P1	8
C1	9
S1	2
GA1	3
GB1	3
P2	4
C2	9
S2	2
GA2	3
GB2	3

4149
4150
4151
4152
4153
4154
4155
4156
4157

3.3.2.1.1 Secure G.729D Voice Transmission Format

The transmission format for Secure G.729D Voice is based on the bit transmission ordering specified in the ITU standards, specifically in Table 8/G.729. That is, the frame parameters shall be transmitted in the order shown in Table 3.3-7. Also, the individual parameters shall be transmitted most significant bit first. The frame transmission ordering is shown in Table 3.3-8.

4158
4159
4160

Table 3.3-8 G.729D Vocoder Frame Bit Transmission Order

Frame Bit #	Parameter -[bit #]						
1	L0-[0]	17	L3-[1]	33	C1-[2]	49	C2-[7]
2	L1-[6]	18	L3-[0]	34	C1-[1]	50	C2-[6]
3	L1-[5]	19	P1-[7]	35	C1-[0]	51	C2-[5]
4	L1-[4]	20	P1-[6]	36	S1-[1]	52	C2-[4]
5	L1-[3]	21	P1-[5]	37	S1-[0]	53	C2-[3]
6	L1-[2]	22	P1-[4]	38	GA1-[2]	54	C2-[2]
7	L1-[1]	23	P1-[3]	39	GA1-[1]	55	C2-[1]
8	L1-[0]	24	P1-[2]	40	GA1-[0]	56	C2-[0]
9	L2-[4]	25	P1-[1]	41	GB1-[2]	57	S2-[1]
10	L2-[3]	26	P1-[0]	42	GB1-[1]	58	S2-[0]
11	L2-[2]	27	C1-[8]	43	GB1-[0]	59	GA2-[2]
12	L2-[1]	28	C1-[7]	44	P2-[3]	60	GA2-[1]
13	L2-[0]	29	C1-[6]	45	P2-[2]	61	GA2-[0]
14	L3-[4]	30	C1-[5]	46	P2-[1]	62	GB2-[2]
15	L3-[3]	31	C1-[4]	47	P2-[0]	63	GB2-[1]
16	L3-[2]	32	C1-[3]	48	C2-[8]	64	GB2-[0]

4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171

NOTES:

1. Bit 0 of a G.729D parameter is the least significant bit.

3.3.2.2 Sync Management Frame

The Sync Management frame shall be transmitted as the first frame of each Secure G.729D Voice superframe. Its format shall be as shown in Figure 3.3-10.

SM HEADER (PN)	PARTIAL LONG COMPONENT	SHORT COMPONENT	PLC INDEX	CRC - 8	PADDING
----------------	------------------------	-----------------	-----------	---------	---------

4172
4173
4174
4175
4176
4177
4178
4179

Figure 3.3-10 Secure G.729D Voice Sync Management Frame Format

The contents of the Secure G.729D Voice Sync Management frame are shown in Table 3.3-9. The SM Header, Partial Long Component, Short Component, PLC Index, and CRC shall be the same as that specified for Secure Burst w/o Blank MELP Voice (see Section 3.3.1.2.1). The

4180 "Padding" field consists of eight bits that shall be set to zero. The bit transmission order for the
4181 Sync Management frame is shown in Table 3.3-11, octets 1 - 8.

4182
4183
4184
4185

Table 3.3-9 Secure G.729D Voice Sync Management Frame Contents

Field	Length (bits)
SM Header (PN Sequence)	16
Partial Long Component	16
Short Component	14
PLC Index	2
CRC	8
(Padding)	8

4186
4187
4188
4189
4190
4191
4192
4193

3.3.2.3 Encrypted Speech Frame Header

The Encrypted Speech frame Header shall be transmitted at the beginning of each Encrypted
Speech frame. Its format shall be as shown in Figure 3.3-11.



4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207

Figure 3.3-11 Secure G.729D Voice Encrypted Speech Frame Header

The contents of the Encrypted Speech frame Header are shown in Table 3.3-10. The fixed 16-bit
ES Header PN sequence (different than the Sync Management frame Header) shall be as defined
in Section 2.5.3. The Partial Short Component refers to an encryption parameter that is specified
in SCIP-230 or SCIP-231, Section 4.1.1.2.1; or SCIP-232, Section 4.1.2.1. The bit transmission
order for the Encrypted Speech frame Header is shown in Table 3.3-11, octets 9 - 11.

Table 3.3-10 Secure G.729D Voice Encrypted Speech Frame Header Contents

Field	Length (bits)
ES PN Sequence	16
Partial Short Component	8

4208

4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223

3.3.2.4 Encryption and Transmission Ordering

G.729D vocoder data is generated in 64-bit frames and formatted in the G.729D Vocoder Frame Bit Transmission Order (see Table 3.3-8). Encryption shall be as specified in SCIP-230 or SCIP-231, Section 4.1.1.2; or SCIP-232, Section 4.1.2.

Encrypted G.729D Voice frames shall have Sync Management frames and Encrypted Speech frame Headers inserted and shall be formatted into superframes as shown in Table 3.3-11. The superframes shall then be passed, in ascending octet order beginning with the first octet of the Sync Management frame Header, to the lower layers for transmission.

Table 3.3-11(a) Secure G.729D Voice Transmission Bit Ordering (Octets 1 - 8)

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
SM Header (PN Sequence)								
0	1	0	1	1	1	1	0	1
0	0	0	1	0	0	1	1	2
Partial Long Component								
b8	b9	b10	b11	b12	b13	b14	b15-msb	3
b0-lsb	b1	b2	b3	b4	b5	b6	b7	4
Short Component								
b6	b7	b8	b9	b10	b11	b12	b13-msb	5
PLC Index								
b0-lsb	b1-msb	b0-lsb	b1	b2	b3	b4	b5	6
CRC								
b0-lsb	b1	b2	b3	b4	b5	b6	b7-msb	7
Padding								
0	0	0	0	0	0	0	0	8

4224

4225
4226
4227

Table 3.3-11(b) Secure G.729D Voice Transmission Bit Ordering (Octets 9 - 288)

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
----- Encrypted Speech Frame 1 -----								
ES PN Sequence								
0	1	1	1	1	0	1	0	9
0	1	1	0	0	1	0	0	10
Partial Short Component								
b0-lsb	b1	b2	b3	b4	b5	b6	b7-msb	11
G.729D Frame 1								
b8	b7	b6	b5	b4	b3	b2	b1	12
...								
b64	b63	b62	b61	b60	b59	b58	b57	19
...								
G.729D Frame 4								
b8	b7	b6	b5	b4	b3	b2	b1	36
...								
b64	b63	b62	b61	b60	b59	b58	b57	43
...								
----- Encrypted Speech Frame 8 -----								
ES PN Sequence								
0	1	1	1	1	0	1	0	254
0	1	1	0	0	1	0	0	255
Partial Short Component								
b0-lsb	b1	b2	b3	b4	b5	b6	b7-msb	256
G.729D Frame 1								
b8	b7	b6	b5	b4	b3	b2	b1	257
...								
b64	b63	b62	b61	b60	b59	b58	b57	264
...								
G.729D Frame 4								
b8	b7	b6	b5	b4	b3	b2	b1	281
...								
b64	b63	b62	b61	b60	b59	b58	b57	288

4228

4229
4230
4231
4232
4233

3.3.2.5 Discontinuous Voice Transmission

TBSL.

Editor's Note: It is expected that the SCIP approach to DTX will be compatible with G.729 Annex F; however, the details remain to be determined.

4234
4235
4236
4237
4238
4239
4240
4241
4242
4243

3.3.2.6 Force Continuous Transmission

During Force Continuous Transmission (FCT) operation, full-length superframes shall be transmitted continuously following the START (see Section 3.2) unless interrupted by the ESCAPE (see Section 2.1.4). The G.729D Active Voice Encoder is run continuously, and all frames that are generated are transmitted.

4244
4245
4246
4247
4248
4249
4250

3.4 Secure Data Applications

Two secure asynchronous data applications are specified herein: Secure Reliable Transport Asynchronous Data (the SCIP MER data application defined in Section 3.4.1) and Secure Best Effort Transport Asynchronous Data (an optional SCIP data application defined in Section 3.4.2). Asynchronous Data Options may be defined in the future for additional data applications such as Fax or Chat.

4251
4252
4253
4254
4255
4256
4257
4258
4259

3.4.1 Secure Reliable Transport (RT) Asynchronous Data

The Secure Reliable Transport Asynchronous Data application utilizes the same transport mechanisms as are used for secure call setup messages to deliver user data reliably. Framing, forward error correction, and residual error detection reduce the maximum throughput for this application to less than 65% of the channel rate.

Editor's Note: The reliable transport application specified in this section may be applicable to synchronous data transmission as well, although issues such as indicating valid bits within an octet need to be addressed for synchronous data transmission.

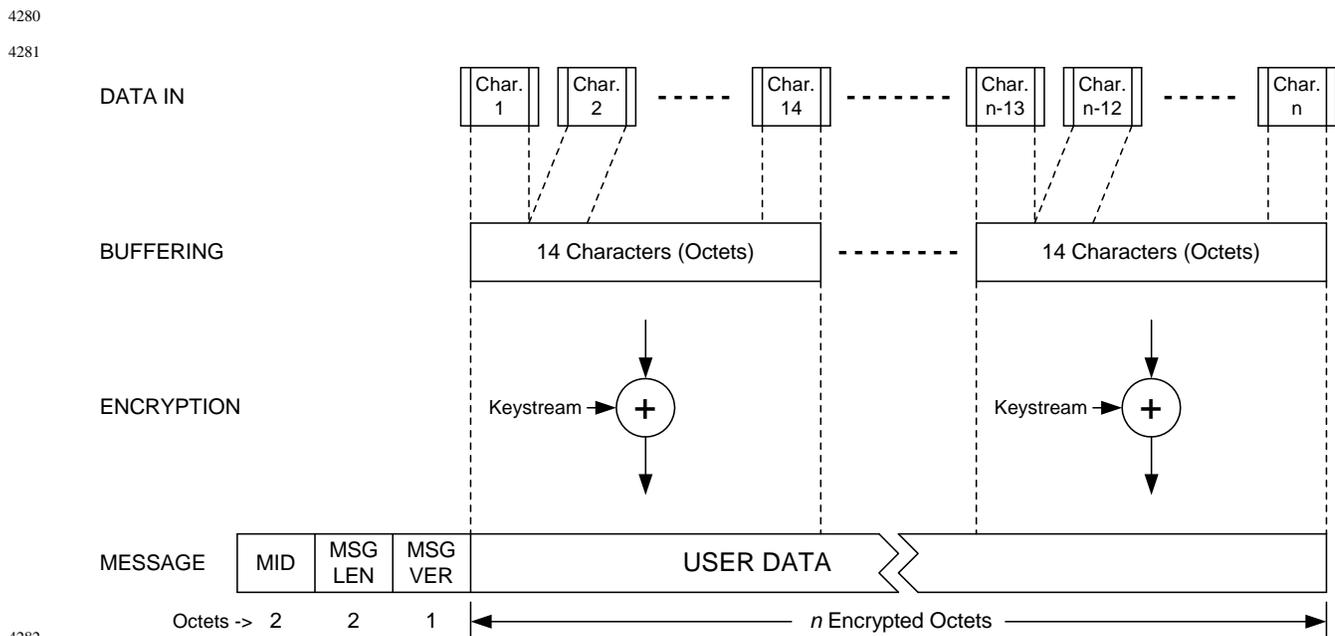
4260
4261
4262
4263
4264
4265
4266
4267
4268

The Secure Reliable Transport (RT) Asynchronous Data application uses the SCIP message transport mechanisms specified in Section 2.1 to provide reliable delivery of user data to a receiving terminal. Following initial call setup signaling or Mode Change signaling where the Secure RT Asynchronous Data application is chosen, the Transport Layer protocol remains in place and transports Secure RT Asynchronous Data messages. Secure RT Asynchronous Data messages contain a variable number of user data octets that have been input from the terminal's data port and encrypted prior to being placed into the User Data fields of these messages.

4269 Since a reliable transport mechanism is used, all transmitted data will arrive at the receiver under
4270 most channel conditions. There is no opportunity for cryptosync to be lost, and late entry is not
4271 an issue for a reliable transport application, which is by definition point-to-point. Therefore,
4272 sync maintenance is not required in the Secure RT Asynchronous Data application.

4273
4274 Note that the reliable transport application specified in this section results in stateless data
4275 handling at the Transport Layer. That is, the Transport Layer does not require knowledge of the
4276 current terminal state (signaling vs. traffic).

4277
4278 Figure 3.4-1 illustrates the processing required for preparation of the Secure RT Asynchronous
4279 Data message.



4285 **Figure 3.4-1 Secure RT Asynchronous Data Message Preparation**

4288 **3.4.1.1 Secure RT Asynchronous Data Transmission**

4289
4290 Once the transition to the Secure RT Asynchronous Data application is complete, the terminal
4291 shall begin accepting plaintext asynchronous data characters at the user data port. Start and stop
4292 bits shall be removed prior to encryption and reinserted at the receiver following decryption.
4293 Plaintext octets (asynchronous data characters with start and stop bits removed) shall be
4294 encrypted and buffered until a sufficient number have been collected to create a Secure RT
4295 Asynchronous Data message. This message format is the same as that for other SCIP signaling
4296 messages, that is, it begins with a two-octet MID followed by a two-octet Message Length field
4297 and a one-octet Message Version field. These fields are not encrypted. The number of
4298 encrypted octets that are placed in each Secure RT Asynchronous Data message is left as an
4299 implementation option. This determination may be based on factors such as the user data port

4300 character rate, desired latency, and the cryptographic block size. A Secure RT Asynchronous
4301 Data message may contain as few as one user data octet or as many as 65,532 (the maximum
4302 allowed by the 16-bit Message Length field).
4303

4304
4305

3.4.1.1.1 Encryption and Transmission Ordering

4306

4307 V.14 asynchronous data is input at the DTE interface as shown in Figure 3.4-2. The start and
4308 stop bits shall be removed and discarded. The 8-bit user data characters shall then be formatted
4309 into 14-octet blocks prior to encryption. If there are fewer than 14 octets to be transmitted, or if
4310 there are fewer than 14 octets remaining for the final block of a Secure RT Asynchronous Data
4311 message, zero padding may be used to complete a 14-octet block. However, the first octet of the
4312 following Secure RT Asynchronous Data message shall be encrypted using a new state vector.
4313 That is, the encryption of all Secure RT Asynchronous Data messages shall begin with a new
4314 state vector. Asynchronous DTE I/O formats other than V.14 may also be supported (e.g., a
4315 USB interface). In any DTE I/O format, the user data characters (octets) shall be extracted from
4316 the DTE I/O format and formatted into 14-octet blocks prior to encryption as illustrated for V.14.
4317 Details pertaining to Secure RT Asynchronous Data encryption may be found in SCIP-230,
4318 Sections 4.1.2.1 and 4.1.2.2, SCIP-231, Section 4.1.2.1, or SCIP-232, Sections 4.2.1 and 4.2.2.
4319

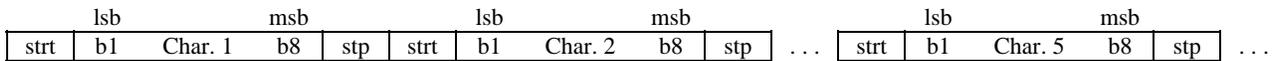
4320

DTE Data in:

4321

4322

time ->



4323

Figure 3.4-2 V.14 Asynchronous Data Input Ordering

4324

4325

4326

4327 After the data has been encrypted, it shall be formatted into the Encrypted User Data Octets field
4328 of the Secure RT Asynchronous Data message as shown in Table 3.4-1. Encrypted padding
4329 octets (if present) shall be discarded. The Secure RT Asynchronous Data message shall then be
4330 passed to the Transport Layer for transmission.
4331

4331

4332
4333
4334

Table 3.4-1 Secure RT Asynchronous Data Message Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
Source ID								
0	1	0	0	0	0	0	0-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
----- Encrypted User Data								
Octet 1								
b8	b7	b6	b5	b4	b3	b2	b1	6
•••								
•••								
•••								
Octet n								
b8	b7	b6	b5	b4	b3	b2	b1	5+n

4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347

n = number of encrypted user data octets

- For the Secure RT Asynchronous Data message the value of the MID is 0x0040.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- For the version of the Secure RT Asynchronous Data message defined in this version of the Signaling Plan, the value of the Message Version field is 0x00.
- The Encrypted User Data Octets field contains a variable number of octets originally obtained from the user data port and encrypted before being placed into this field.

4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393

3.4.1.1.2 Message Transmission

Each complete Secure RT Asynchronous Data message shall be passed to the Transport Layer where the processes specified in Section 2.1, including SOM/EOM framing, frame counter, CRC, FEC, and ACK/NAK using REPORT messages, shall be used to provide reliable transmission to the far-end terminal.

Appropriate flow control procedures (e.g. RTS/CTS or XON/XOFF) shall be implemented at the user data port to prevent loss of data in the event data arrives faster than it can be transmitted over the communications channel. Note that a terminal may also flow control the communications channel receive data rate if necessary by delaying the normal frame acknowledgement at the Transport Layer.

If the DTE lowers Request to Send (RTS) while the Secure RT Asynchronous Data application is active, the terminal shall build and transmit a Secure RT Asynchronous Data message containing any buffered data, and then cease transmitting. When RTS is again activated, the terminal shall again start accepting octets from the user data port and building Secure RT Asynchronous Data messages. The crypto is not flywheeled during periods when the terminal is not transmitting.

3.4.1.2 Secure RT Asynchronous Data Message Reception

Following the transition from signaling to the Secure RT Asynchronous Data application, the receiving terminal's Transport Layer shall continue to monitor the communications channel searching for incoming SOM patterns and the associated transport frames as described in Section 2.1.7. Payload data from the transport frames shall be transferred to the message layer, where Secure RT Asynchronous Data messages shall be verified and interpreted.

Ciphertext data extracted from the Encrypted User Data Octets field of each Secure RT Asynchronous Data message (see Table 3.4-1) shall be decrypted in the order shown in Figure 3.4-3. Start and stop bits shall then be reinserted, and the 10-bit data characters shall be forwarded to the user data port in the order shown in Figure 3.4-2 for V.14 asynchronous data. For asynchronous DTE I/O formats other than V.14 (e.g., a USB interface) received data shall be reformatted to the receiving DTE I/O format, which may be different than the sending DTE I/O format.

3.4.2 Secure Best Effort Transport (BET) Asynchronous Data

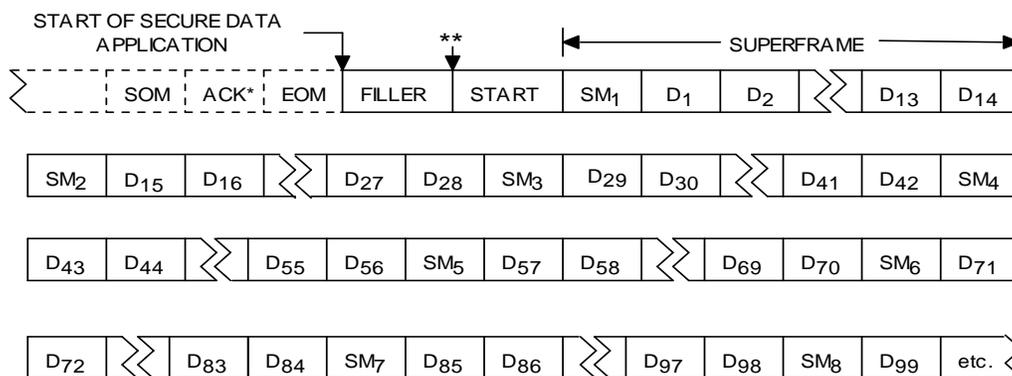
The Best Effort Transport (BET) Asynchronous Data application utilizes the channel capacity efficiently by extracting the asynchronous character octets from their I/O format (e.g., V.14 Start/Stop bits or USB Framing). The channel capacity saved by not transmitting the character framing allows Sync Management frames to be inserted into the transmitted data at periodic intervals.

Editor's Note: This data application was originally conceived to allow 2400 bps V.14 asynchronous data to be compressed sufficiently to allow SCIP overhead to be inserted in a 2400 bps channel. The data application scales to any data rate and is usable with asynchronous DTE I/O formats other than V.14.

4394 This Section defines a secure data application that may optionally be supported by SCIP
4395 implementations.
4396
4397

4398 Signaling for Secure BET Asynchronous Data shall be in a "Burst w/o Blank" format. The Sync
4399 Management frame contains information that allows cryptographic synchronization maintenance.
4400 No user data is discarded; the Sync Management frame is inserted between consecutive frames
4401 of user data. The discarding of start and stop bits ensures sufficient capacity to permit the
4402 transmission of the Sync Management frame.
4403

4404 Secure BET Asynchronous Data shall be transmitted in 162-octet "superframes" consisting of a
4405 64-bit Sync Management frame followed by fourteen 11 octet asynchronous data frames (64/8
4406 +14*11 = 162 octets). An example of this is shown in Figure 3.4-3. A more detailed picture of a
4407 single superframe is shown in Figure 3.4-4. The superframe shall begin with a Sync
4408 Management frame, and each asynchronous data frame shall be composed of 0 to 10 user data
4409 characters, followed by 0 to 10 filler octets having the value 0x00, followed by a one octet
4410 Validity Count that consists of a 4-bit character Count field and a 4-bit Count Check field. Filler
4411 octets shall be used if no data is available from the DTE. The number of user data characters
4412 plus the number of filler octets in a data frame shall sum to ten. The value of the character Count
4413 field shall be set to the number of user data characters in the frame. Start and stop bits for the
4414 V.14 asynchronous data characters shall not be transmitted. No user data is discarded; the Sync
4415 Management frame shall precede the first 11-octet frame of user data and shall be inserted before
4416 the first 11-octet user data frame of each subsequent superframe.
4417
4418

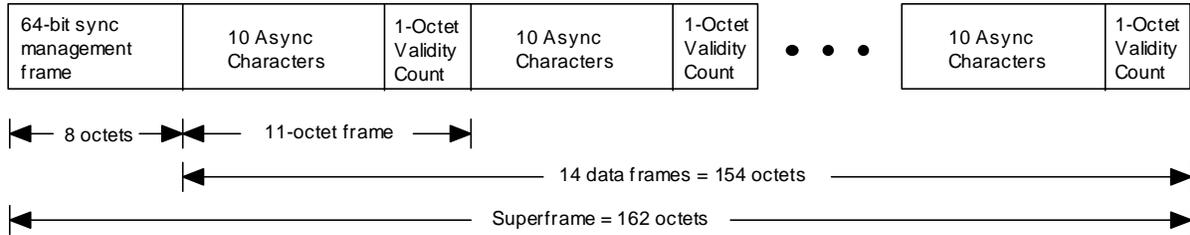


NOTES:
SM = Sync Management Frame
D = 11-octet Async Data Frame
* = ACKed via Report Message
** = Application re-entry point after Notification Message processing

4419
4420
4421

Figure 3.4-3 Secure BET Asynchronous Data Transmission Format

4422
4423



SVSE0003

4424
4425
4426
4427
4428
4429
4430
4431

Figure 3.4-4 Secure BET Asynchronous Data Superframe Structure

The value of the Count Check field is set based on the value of the character Count field. The correspondence is given in Table 3.4-2. Note that this combination of eight bits provides a Hamming distance of four and thus a capability to detect two bit errors and correct one.

Editor's Note: Bit b_1 is set to provide even parity in bits b_1 , b_5 , b_6 , and b_7 . Bit b_2 is set to provide even parity in bits b_2 , b_5 , b_6 , and b_8 . Bit b_3 is set to provide even parity in bits b_3 , b_5 , b_7 , and b_8 . Bit b_4 is set to provide even parity in all bits in the octet. This corresponds to encoding the Count field with a Hamming (7, 3) code augmented with an overall parity bit.

4432
4433
4434
4435

Table 3.4-2 Validity Count Field Values

b8 (msb)	Count	b5 (lsb)	b4 (msb)	Count Check	b1 (lsb)
	0x0			0x0	
	0x1			0x7	
	0x2			0xB	
	0x3			0xC	
	0x4			0xD	
	0x5			0xA	
	0x6			0x6	
	0x7			0x1	
	0x8			0xE	
	0x9			0x9	
	0xA			0x5	

4436

4437
4438
4439
4440
4441
4442
4443

3.4.2.1 Sync Management Frame

The Sync Management frame shall be transmitted as the first frame of each Secure BET Asynchronous Data superframe. Its format shall be as shown in Figure 3.4-5.

HEADER (PN)	PARTIAL LONG COMPONENT	SHORT COMPONENT	PLC INDEX	CRC - 8	PADDING
-------------	------------------------	-----------------	-----------	---------	---------

4444
4445
4446
4447
4448

Figure 3.4-5 Sync Management Frame Format

The contents of the secure asynchronous data Sync Management frame are shown in Table 3.4-3. The Header, Partial Long Component, Short Component, PLC Index and CRC shall be the same as that specified for Secure Burst w/o Blank MELP Voice (see Section 3.3.1.2.1). The "Padding" field consists of eight bits that shall be set to zero. The bit transmission order for the Sync Management frame is shown in Table 3.4-4.

4454
4455
4456
4457

Table 3.4-3 Sync Management Frame Contents

Field	Length (bits)
Header (PN Sequence)	16
Partial Long Component	16
Short Component	14
PLC Index	2
CRC	8
(Padding)	8

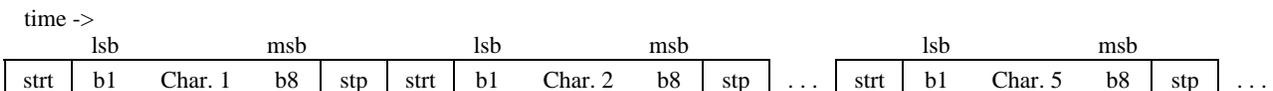
4458
4459
4460
4461

3.4.2.2 Encryption and Transmission Ordering

V.14 asynchronous data is input at the DTE interface as shown in Figure 3.4-6. The start and stop bits shall be removed, and the 8-bit user data characters shall be encrypted. Asynchronous DTE I/O formats other than V.14 may also be supported (e.g., a USB interface). In any DTE I/O format, the user data characters (octets) shall be extracted from the DTE I/O format prior to encryption as illustrated for V.14.

4467
4468
4469
4470

DTE Data in:



4471
4472

Figure 3.4-6 V.14 Asynchronous Data Input Ordering

4473
4474
4475
4476
4477
4478
4479
4480

When the data has been encrypted, it shall be formatted into superframes as shown in Table 3.4-4. The superframes shall then be passed, in ascending octet order beginning with the first octet of the Header, to the lower layers for transmission.

Table 3.4-4 Secure BET Asynchronous Data Transmission Bit Ordering

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Header (PN Sequence)								
0	1	0	1	1	1	1	0	1
0	0	0	1	0	0	1	1	2
Partial Long Component								
b8	b9	b10	b11	b12	b13	b14	b15-msb	3
b0-lsb	b1	b2	b3	b4	b5	b6	b7	4
Short Component								
b6	b7	b8	b9	b10	b11	b12	b13-msb	5
PLC Index								
b0-lsb	b1-msb	b0-lsb	b1	b2	b3	b4	b5	6
CRC								
b0-lsb	b1	b2	b3	b4	b5	b6	b7-msb	7
Zero Filler								
0	0	0	0	0	0	0	0	8
Data Frame 1								
Char 1								
b8	b7	b6	b5	b4	b3	b2	b1	9
•••								
Char 10								
b8	b7	b6	b5	b4	b3	b2	b1	18
Count				b5-lsb	Count Check		b1-lsb	19
b8-msb	b7	b6	b5-lsb	b4-msb	b3	b2	b1-lsb	
•••								
Data Frame 14								
Char 1								
b8	b7	b6	b5	b4	b3	b2	b1	152
•••								
Char 10								
b8	b7	b6	b5	b4	b3	b2	b1	161
Count				b5-lsb	Count Check		b1-lsb	162
b8-msb	b7	b6	b5-lsb	b4-msb	b3	b2	b1-lsb	

4481
4482

4483 If the DTE lowers Request to Send (RTS) while the Secure BET Asynchronous Data application
4484 is active, the terminal shall complete transmission of the current superframe, filling data frame
4485 octets with 0x00 as required. If RTS remains low at the end of the superframe, the terminal shall
4486 cease transmitting. When RTS is again activated, asynchronous data transmission shall begin
4487 with a Sync Management frame followed by asynchronous data frames.
4488

4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502

4.0 SCIP ELECTRONIC REKEY SIGNALING

This section specifies the signaling required to electronically rekey the FIREFLY or ECMQV key material in SCIP terminals. In addition to providing new key material, the rekey data authenticates the SCIP terminal, and is customized to furnish organizational identification information. Compromised Key List (CKL) management is also provided as part of the Electronic Rekey function. The design approach for Electronic Rekey utilizes the Generic Rekey Front End (GRFE) to interface to the Key Processing Facility (KPF). The telephone network interface for the GRFE is provided by the SCIP Line Interface Terminal (SCIP-LIT), which establishes a secure call with the calling SCIP terminal and provides encryption for the Rekey Application Protocol Data Units (APDUs). Electronic Rekey is an independent Operational Mode in the SCIP terminal that is negotiated automatically on calls to the SCIP-LIT.

Editor's Note: The electronic rekey facility for ECMQV key material may be different than the one described herein for FIREFLY key material.

4503
4504
4505
4506
4507
4508
4509

Section 4.1 describes the SCIP Electronic Rekey protocol architecture and communication paths. Section 4.2 specifies the SCIP Message Transport layer. This is followed by an overview of the Adaptation layer and Generic Rekey Application layer in Sections 4.3 and 4.4, respectively. Detailed Electronic Rekey processing requirements are specified in SCIP-230, Section 6, or SCIP-232, Appendix E.

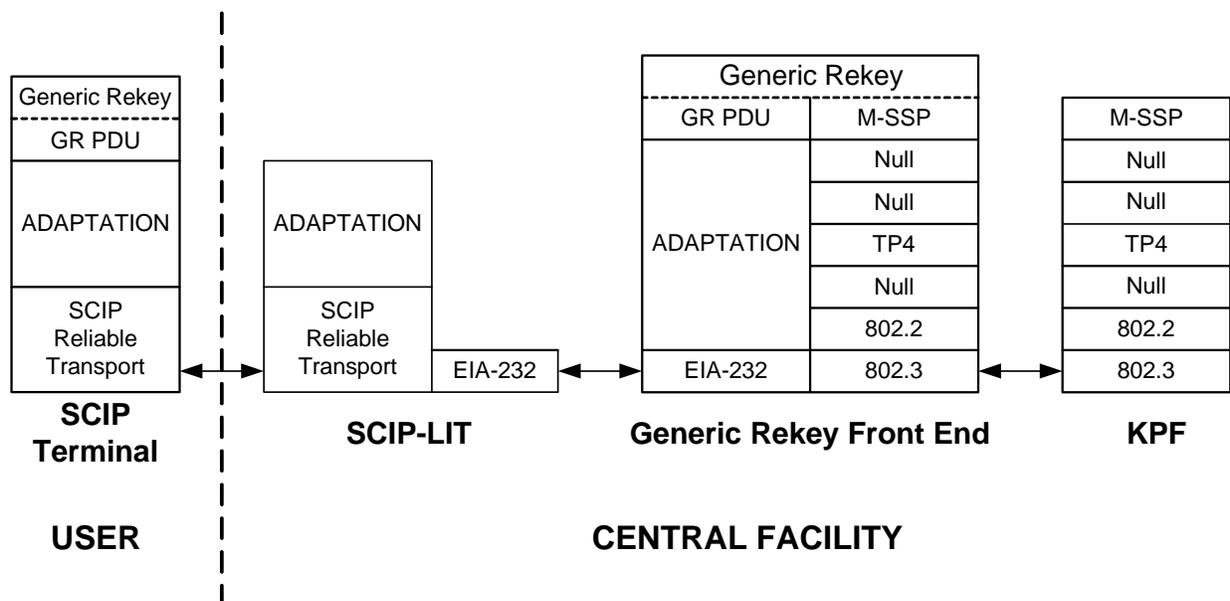
Editor's Note: The user interface for initiating rekey is left to the implementer. Options include a "Rekey" button on the terminal, programming an available speed-dialing button to dial the rekey telephone number, and simply manually dialing the rekey number. When the SCIP-LIT answers the call, it will initiate secure call setup automatically.

4510

4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522

4.1 Electronic Rekey Protocol Architecture and Communication Paths

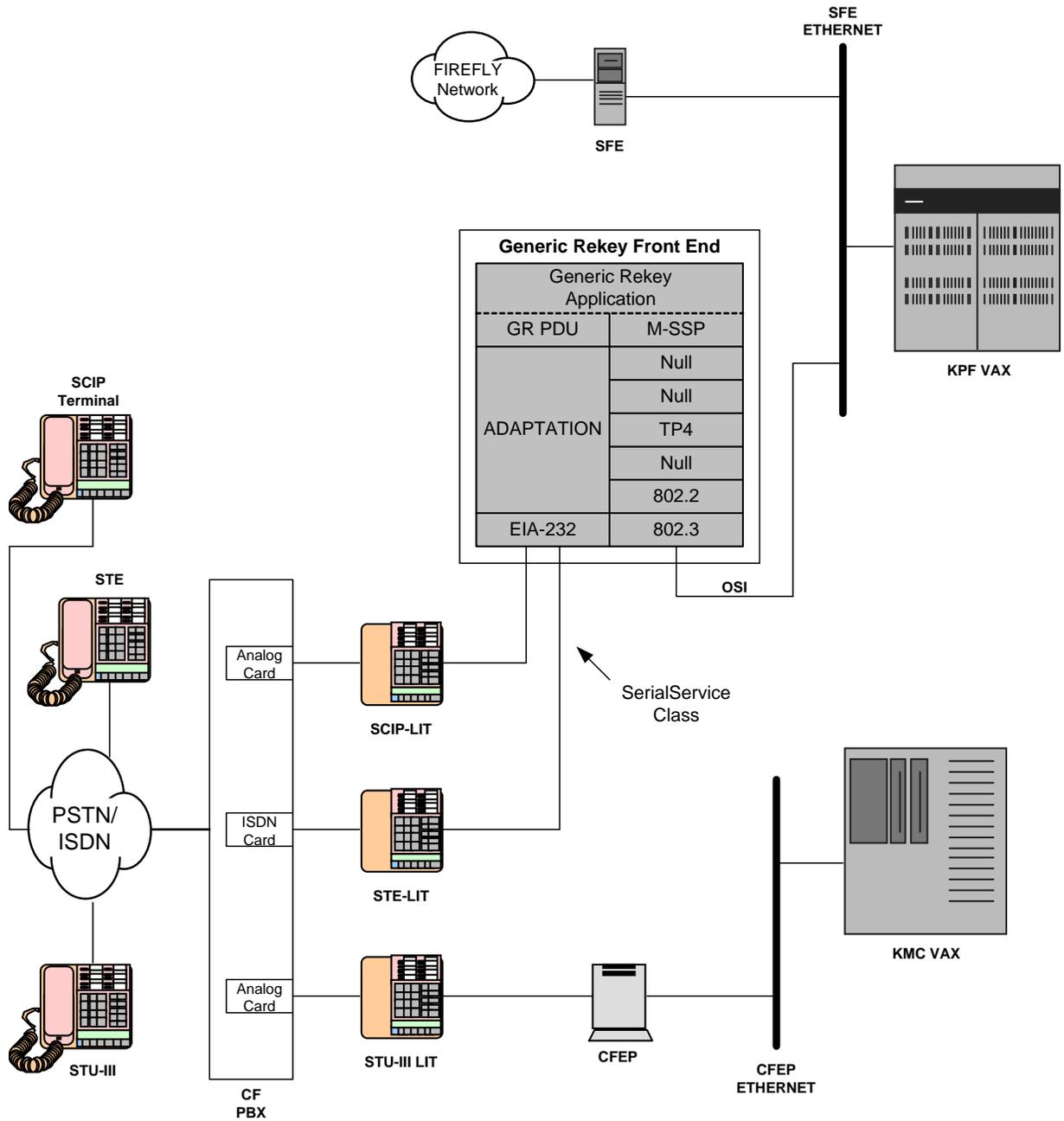
The protocol layer descriptions in this section identify the subset of the OSI seven-layer model that is used in the SCIP terminal to perform Electronic Rekey through the GRFE. The GRFE provides protocol conversion and performs limited authentication between the SCIP terminals and the KPF. Figure 4.1-1 illustrates the rekey protocol stacks for each device in the SCIP Electronic Rekey communication path. As shown, the terminal must implement the Generic Rekey Protocol (GRP) at the application layer, an Adaptation layer function that reassembles GRPDUs into APDUs, and the SCIP Message Transport protocol specified in Section 2.1.



4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535

Figure 4.1-1 Rekey Protocol Conversion Using the GRFE

Figure 4.1-2 illustrates the communication devices and paths employed by the rekey system infrastructure. The GRFE has serial interfaces that connect to the SCIP-LIT and the STE-LIT. The GRFE communicates with the Central Facility (CF) KPFs over an Ethernet interface and interfaces to the CF's Digital PBX via the LITs. Analog cards are installed at the PBX for connections to the SCIP-LITs and the STU-III LITs. (The STU-III rekey path is shown for completeness only.)



NOTES:

1. There are generally more than one of each type of LIT; however, only one is shown here to illustrate the architecture.

4536
4537
4538
4539

Figure 4.1-2 Electronic Rekey System Infrastructure

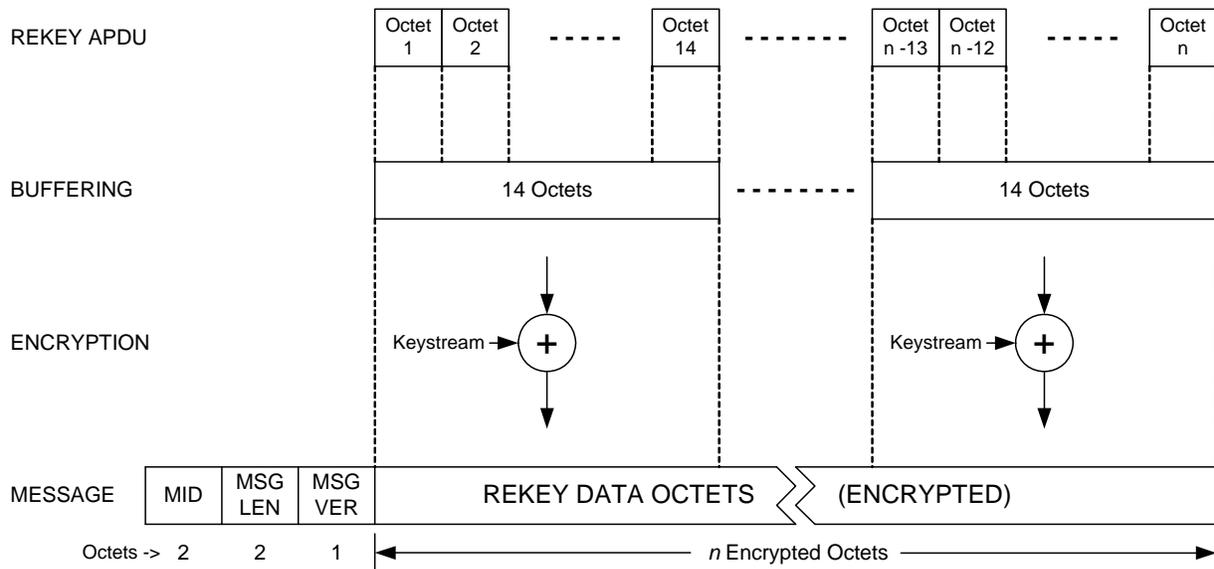
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558

4.2 SCIP Electronic Rekey Message Transport

The SCIP Electronic Rekey application uses the SCIP message transport mechanisms specified in Section 2.1 to assure that all rekey data sent from the GRFE arrives at the receiving terminal error-free under most channel conditions and in exactly the same order it was originally sent. Following initial SCIP call setup signaling, where the SCIP Electronic Rekey application (the only SCIP application supported by the SCIP-LIT) is negotiated, the transport layer protocol remains in place and transports SCIP Rekey Messages. SCIP Rekey Messages carry the variable-length Rekey APDUs as their payloads.

Figure 4.2-1 illustrates how the Rekey APDUs, specified in SCIP-230, Section 6.2, or SCIP-232, Appendix E.2, are encapsulated in SCIP Rekey Messages.

Note that a Rekey APDU may be transmitted either in a single Rekey Message or in multiple Rekey Messages. The SCIP-LIT typically transmits a Rekey Response APDU in multiple Rekey Messages.



4559
4560
4561
4562

Figure 4.2-1 SCIP Rekey Message Preparation

4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587

4.2.1 Encryption and Transmission Ordering

The Rekey APDU octets are formatted into 14-octet blocks prior to encryption. If there are fewer than 14 octets remaining in the final block of an APDU, zero padding may be used to complete a 14-octet block. Rekey octets are encrypted in the order they appear in the Rekey APDUs beginning with the high order Adaptation layer octet (see SCIP-230, Section 6.2.1, or SCIP-232, Appendix E.2.1). Detailed requirements for Rekey APDU encryption are specified in SCIP-230, Sections 6.2.1 and 6.2.2, or SCIP-232, Appendices E.2.1 and E.2.2.

After the Rekey APDU octets have been encrypted, they shall be formatted into the Encrypted Rekey APDU field of the SCIP Rekey Message as shown in Table 4.2-1. Encrypted padding octets (if present) shall be discarded. The SCIP Rekey Message shall then be passed to the Transport layer for transmission.

Table 4.2-1 SCIP Rekey Message Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
MID								
0-msb	0	0	0	0	0	0	0	1
Source ID								
1	1	1	0	0	0	0	0-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
----- Encrypted Rekey Data								
Octet 1								
b8	b7	b6	b5	b4	b3	b2	b1	6
•••								
Octet n								
b8	b7	b6	b5	b4	b3	b2	b1	5 + n

n = number of Encrypted Rekey Data octets.

- For the SCIP Rekey Message the value of the MID is 0x00E0.
- The Message Length shall contain the actual length of the message body (including the length of the Message Length field itself but not including the length of the MID field) in octets. The value of the field shall be an unsigned binary integer with the high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.

- 4588 • For the version of the SCIP Rekey Message defined in this version of the Signaling
4589 Plan, the value of the Message Version field is 0x00.
- 4590 • The Encrypted Rekey Data field contains a variable number of octets that correspond
4591 to an encrypted Rekey APDU or partial APDU. The msb of the high order
4592 Adaptation layer octet of the encrypted APDU (as defined in SCIP-230, Section
4593 6.2.1, or SCIP-232, Appendix E.2.1) shall be placed in bit 8 of octet 1 of the first
4594 Rekey Message.

4595
4596

4.2.2 SCIP Rekey Message Transmission

4597

4598 SCIP Rekey Messages are constructed at the SCIP-LIT or the destination terminal, depending on
4599 the direction the message will travel. The SCIP-LIT or the destination terminal shall then use
4600 the processes described in Section 2.1, including SOM/EOM framing, frame counters, CRC,
4601 FEC, and ACK/NAK using REPORT Messages, to transmit the SCIP Rekey Message to the
4602 destination terminal or to the SCIP-LIT. The message shall be transmitted in ascending octet
4603 order.
4604

4605
4606

4.2.3 SCIP Rekey Message Reception

4607

4608 Following the transition from call setup signaling to SCIP Rekey, the receiving terminal's
4609 Transport layer shall continue to monitor the communications channel searching for incoming
4610 SOM patterns and the associated transport frames as described in Section 2.1.6. Payload data
4611 from the transport frames shall be transferred to the message layer, where SCIP Rekey Messages
4612 shall be verified and interpreted.
4613

4614

4615 Information extracted from the Encrypted Rekey Data field of each received SCIP Rekey
4616 Message shall be decrypted in the order shown in Figure 4.2-2 and passed to the Adaptation
4617 layer described in Section 4.3.

4618

4619 The terminal shall wait at least 10 seconds after call setup is complete before transmitting a
4620 Rekey Request Message (Grkrq) to the SCIP-LIT. This is required to accommodate the SCIP-
4621 LIT processing delay.

4622

4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646

4.3 Adaptation Layer

The Adaptation layer, which resides between the Generic Rekey Application layer and the SCIP Reliable Transport layer, is used to convey application PDU length information and to reassemble application PDUs from the received Rekey data. On the transmit side, a two-octet length field is appended to the front of each Generic Rekey PDU (GRPDU) indicating the number of octets in the PDU (not including the appended length field). If Rekey Option 0x0006 (with 32-bit CRC) was negotiated, the CRC check bits are also computed and appended to the end of the GRPDU. The resulting Rekey APDU, comprised of the GRPDU and the appended length field (and CRC, if this option was negotiated), is encrypted and then transmitted using the SCIP Reliable Transport. On the receive side, the decrypted Rekey APDU is passed to the Adaptation layer, which extracts and examines the two-octet length field to determine the number of octets in the application PDU to be reconstructed (and verifies the CRC, if this option was negotiated).

If Rekey Option 0x0006 (with 32-bit CRC) was negotiated and the CRC verification fails at the SCIP-LIT, the SCIP-LIT shall terminate the call by transmitting a Notification Message with the Action set to Connection Terminate and the Information Code set to *Rekey Message CRC failure*. If the CRC verification fails at the terminal being rekeyed, the terminal shall proceed with one of the options specified in SCIP-230, Section 6.2.1, or SCIP-232, Appendix E.2.1.

Further details of the Adaptation layer are specified in SCIP-230, Section 6.2.1, or SCIP-232, Appendix E.2.1.

4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658

4.4 Generic Rekey Application Layer

For SCIP Electronic Rekey, the terminal shall implement the Generic Rekey Protocol at the Application layer. The Generic Rekey Protocol (GRP) is used for the transmission of rekey requests and acknowledgments (by the terminal), and for the transmission of rekey/CKL data and associated error indications (by the GRFE). Table 4.4-1 lists the currently specified GRP messages or protocol data units (GRPDUs).

Table 4.4-1 Generic Rekey Protocol Data Units (GRPDUs)

GRPDU	PDU Description
Grkrq	Terminal request for a rekey or seed conversion from the KPF/GRFE.
Grkrs	KPF/GRFE response to terminal's rekey request with current and/or next keys encrypted separately.
Gerror	KPF/GRFE response indicating an error and/or to keep the communication link open.
Grkcmp	Terminal's acknowledgment of a completed rekey update with a success/failure indication.

4659
4660
4661
4662
4663
4664
4665
4666
4667

All GRPDUs are encoded using the transfer syntax Distinguished Encoding Rules (DERs), processed using the Adaptation function, and transmitted according to the procedures specified in Section 4.2. The format and use of each PDU, including syntax elements, component lengths (where applicable) and format and value restrictions, are specified in SCIP-230, Section 6.2.3, or SCIP-232, Appendix E.2.3. ASN.1 encoding definitions of the GRPDUs are provided in SCIP-230, Appendices A.2 and B.2, or SCIP-232, Appendices E.3 and E.4.

4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693

5.0 SCIP SIGNALING – Multipoint Operation

This section defines the SCIP signaling for multipoint operation, which is a one-to-many mode with one transmitter and multiple receivers. The transmitter and receivers use PPKs for encryption and decryption of secure application traffic. The specific encryption key, encryption algorithm, and secure application option to be used during multipoint operation is determined prior to the start of SCIP multipoint signaling. The transmitter and receivers need this information to establish secure multipoint communication. Note that octet alignment is not implied or required by this specification, and should not be expected by the receiving terminals. Section 5.1 specifies SCIP message framing and transport, including the Multipoint Cryptosync (MCS) Message, which is used to initiate SCIP secure application traffic. Section 5.2 specifies a multipoint session, including multipoint transmission and reception.

5.1 Multipoint Message Transport

For multipoint operation, information is transmitted (broadcast) one-way, without the ability to acknowledge the reception. An example transport signaling timeline for transmitting multipoint framed and full bandwidth traffic is shown in Figure 5.1-1. “Framed” traffic, as defined in Section 2.1.3, applies to SCIP multipoint signaling traffic, and multipoint “full bandwidth” traffic applies to encrypted application traffic that is transmitted with sync management information included, as specified in Section 5.2.1.2. Following SCIP multipoint signaling traffic, a transition occurs from multipoint “framed” traffic to “full bandwidth” traffic.



NOTES
EOT = End of Transmission

4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709

Figure 5.1-1 Multipoint Transport Signaling Timeline

5.1.1 Multipoint Transport Framing

Transport framing used for point-to-point operation, as specified in Section 2.1.3, is also used for multipoint operation with the following exceptions. The Frame Count (FC) is used by the receiving terminals to identify the corresponding frames in multiple transmissions of the MCS Message. The receiving terminals cannot request frame retransmissions as is done in point-to-point operation with the REPORT message.

The transmission frame group used for point-to-point and multipoint operation contains an SOM (see Section 2.1.3.1), one or more frames, and an EOM (see Section 2.1.3.5). Each frame contains an FC (see Section 2.1.3.2), Message Data (see Table 2.1-1), CRC (see Section 2.1.3.3),

4710 and FEC (see Section 2.1.3.4). The Frame Count shall be set to 0x01 at the start of each MCS
4711 Message transmission and incremented, by one, for each subsequent message frame transmitted.
4712 Frame Count = 0x00 is reserved for Transport Layer control messages. There are no Transport
4713 Layer control messages in multipoint operation.

4714
4715

4716 **5.1.1.1 Multipoint Message Transmission**

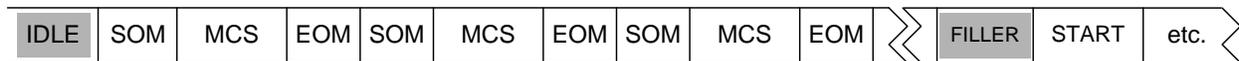
4717
4718

4718 Messages shall be transmitted in frame groups. Within a frame group, an SOM is transmitted
4719 first. Then message frames are transmitted followed by an EOM. If the same message is
4720 transmitted multiple times, the frames in each message repetition will have the same Frame
4721 Count values as the original transmission.

4722
4723

4723 Multiple copies of the MCS Message may be transmitted, as shown in Figure 5.1-2, to increase
4724 the likelihood that the receivers will either receive or assemble one copy of the message with no
4725 uncorrectable errors.

4726
4727



NOTES
SOM = Start of Message
MCS = Multipoint Cryptosync Message
EOM = End of Message

4728
4729

4730 **Figure 5.1-2 Multiple Multipoint Cryptosync Message Transmissions**

4731
4732

4733 **5.1.1.2 Multipoint Message Reception**

4734
4735

4735 When an SOM is received, the receiver shall parse a 20-octet frame from the incoming data
4736 stream. The receiver may perform an FEC decode and shall use the CRC to verify that the frame
4737 was received correctly. Note that FEC decoding may have corrected transmission errors.

4738
4739

4739 If the CRC passes, the frame shall be marked as received correctly.

4740
4741

4741 If the CRC does not pass, the frame shall be marked as received incorrectly. The receiver shall
4742 repeat the above processing for each subsequent 20-octet frame until either an EOM or an SOM
4743 is detected.

4744

Editor's Note: Note that the implementer may choose to consider a frame as being received incorrectly if FEC decoding is not successful. In this case, checking the CRC is not required.

4745

4746 If an EOM is received, the receiver waits for the next SOM or the START. If an SOM is
4747 received, the receiver immediately starts processing the frames that follow the SOM.
4748

Editor's Note: If a receiver is able to recognize and process frames in a frame group even when an SOM is not detected (e.g., by working backward from an EOM that is detected), this is permitted though it is not required.

4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760

5.1.2 Multipoint Cryptosync Message

SCIP multipoint signaling begins with the transmission of the MCS Message to provide synchronization to the receiving terminals and initiate multipoint secure application traffic. The Application IV for the multipoint secure application is transmitted in the MCS Message. The transmitter also generates the Sync Verification pattern that allows the receiving terminals to verify that encryption and decryption are operating properly, and transmits it in the MCS Message.

5.1.2.1 Multipoint Cryptosync Message Definition

The format of the Multipoint Cryptosync Message is shown in Table 5.1-1.

4761
4762
4763
4764
4765
4766
4767

Table 5.1-1 Multipoint Cryptosync Message – General Format

8 (msb)	7	6	5	4	3	2	1 (lsb)	
MID								← Bits Octets ↓
0-msb	0	0	0	0	0	0	0	1
Source ID								
0	0	0	0	1	0	0	1-lsb	2
Message Length								
X-msb	X	X	X	X	X	X	X	3
X	X	X	X	X	X	X	X-lsb	4
Message Version								
0	0	0	0	0	0	0	0	5
Application IV Length								
X-msb	X	X	X	X	X	X	X	6
X	X	X	X	X	X	X	X-lsb	7

4768

4769
4770
4771

Table 5.1-1 Multipoint Cryptosync Message – General Format (Cont.)

8 (msb)	7	6	5	4	3	2	1 (lsb)	← Bits Octets ↓
Application IV								8
X-msb	X	X	X	X	X	X	X	
...								7+N
X	X	X	X	X	X	X	X-lsb	
Sync Parameters Length								8+N
X	X	X	X	X	X	X	X	
First Sync Parameter ID								9+N
0	0	0	0	0	0	0	1	
First Sync Parameter Length								10+N
X	X	X	X	X	X	X	X	
First Sync Parameter								11+N
X-msb	X	X	X	X	X	X	X	
...								10+N+M ₁
X	X	X	X	X	X	X	X-lsb	
...								7+N+2L+ M _T -M _L
L'th Sync Parameter ID								
X	X	X	X	X	X	X	X	
L'th Sync Parameter Length								8+N+2L+ M _T -M _L
X	X	X	X	X	X	X	X	
L'th Sync Parameter								9+N+2L+ M _T -M _L
X-msb	X	X	X	X	X	X	X	
...								8+N+2L+ M _T
X	X	X	X	X	X	X	X-lsb	

4772 N = Length of Application IV. M₁ = Length of First Sync Parameter. L = Number of Sync Parameter Entries.
4773 M_L = Length of L'th Sync Parameter. M_T = Total Length of Sync Parameters (i.e., M₁ + M₂ + ... + M_L).
4774
4775

- 4776 • For the Multipoint Cryptosync Message, the value of the MID shall be 0x0009.
- 4777 • The Message Length shall contain the actual length of the MCS Message (including
- 4778 the length of the Message Length field itself but not including the length of the MID
- 4779 field) in octets. The value of the field shall be an unsigned binary integer with the
- 4780 high order bit being bit 8 of octet 3 and the low order bit being bit 1 of octet 4.
- 4781 • For the version of the MCS Message defined in this version of the Signaling Plan, the
- 4782 value of the Message Version field is 0x00.
- 4783 • The Application IV Length shall contain the length of the Application IV field in
- 4784 octets (plus the length of the Application IV Length field itself). The value of the
- 4785 field shall be an unsigned binary integer with the high order bit being bit 8 of octet 6
- 4786 and the low order bit being bit 1 of octet 7.
- 4787 • The Application IV shall contain the IV to be used with the application that has been
- 4788 selected. Details of the length, format, and content are found in SCIP-232, Section
- 4789 3.6.1.2 (SCIP-230 and SCIP-231 Sections TBD). The msb of the IV (as defined in
- 4790 SCIP-23x) is placed in bit 8 of octet 8.
- 4791 • The Sync Parameters Length shall contain the total length of the Sync Parameters in
- 4792 octets (plus the length of the Sync Parameters Length field itself). The value of the
- 4793 field shall be an unsigned binary integer with the high order bit being bit 8 and the
- 4794 low order bit being bit 1.
- 4795 • The Sync Parameter ID fields shall contain the IDs of the Sync Parameters listed in
- 4796 Table 5.1-2. Sync Parameter IDs are unique to each Sync Parameter. The value of
- 4797 the field shall be an unsigned binary integer with the high order bit being bit 8 and the
- 4798 low order bit being bit 1.
- 4799 • The Sync Parameter Lengths shall contain the lengths of the Sync Parameter fields in
- 4800 octets (plus the length of the Sync Parameter Length field itself). The value of the
- 4801 field shall be an unsigned binary integer with the high order bit being bit 8 and the
- 4802 low order bit being bit 1.
- 4803 • The Sync Parameter fields shall contain the Sync Parameters identified by the Sync
- 4804 Parameter IDs listed in Table 5.1-2, and specified in Section 5.1.2.2.

5.1.2.2 Multipoint Sync Parameters

4805
4806
4807 This section specifies the Sync Parameters for multipoint operation. The First Sync Parameter in
4808 the MCS Message is mandatory, and shall be the Sync Verification pattern. All other Sync
4809 Parameters are optional, and may be listed in any order. Currently defined Sync Parameters are
4810 listed in Table 5.1-2.
4811
4812

Table 5.1-2 Sync Parameters

Sync Parameter ID	Sync Parameter
0x01	Sync Verification pattern

4817

4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853

5.1.2.2.1 Sync Verification Pattern

The Sync Verification pattern is generated, as specified in SCIP-232, Section 3.6.2.2 (SCIP-230 and SCIP-231 Sections TBD), to verify proper operation of the cryptography. Its Sync Parameter ID shall be set to 0x01.

5.1.3 FILLER – Multipoint Operation

For multipoint operation, the transmitter inserts FILLER, which is the same pattern used for point-to-point operation, between the end of the MCS Message and the beginning of the START. The FILLER pattern is a 64-bit pseudorandom sequence that is repeated an integer number of times; however, there is no minimum duration of FILLER for multipoint operation. The purpose of FILLER is to allow the receivers sufficient time to process the MCS Message. The value of the FILLER pattern is specified in Table 2.5-3.

5.1.4 START – Multipoint Operation

For multipoint operation, the transmitter uses START, which is the same pattern used in point-to-point operation, to allow the receiving terminals to detect the beginning of multipoint full bandwidth traffic. The START pattern is a 64-bit pseudorandom sequence that allows acceptable detection performance in the anticipated error environments. The value of the START pattern is specified in Table 2.5-3.

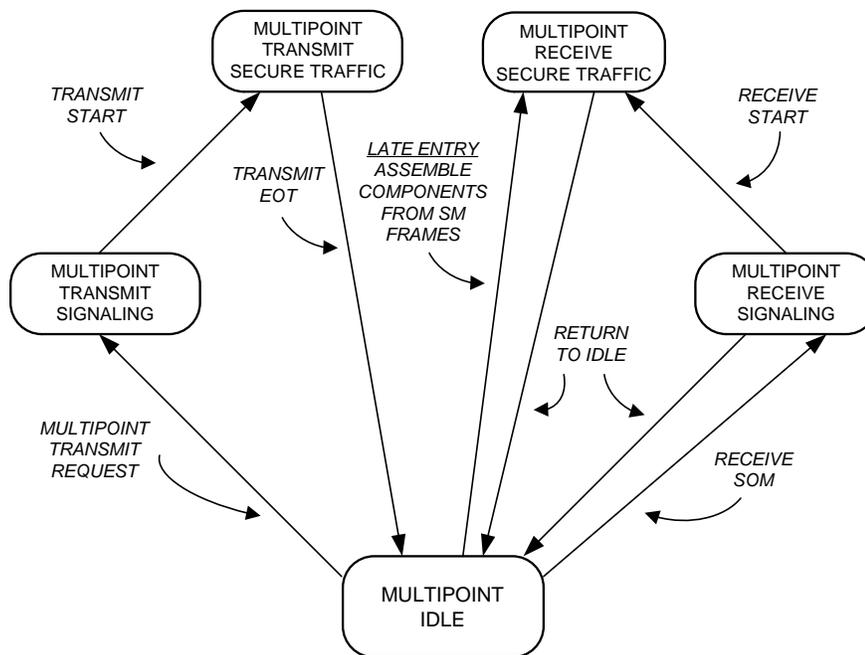
5.1.5 End of Transmission – Multipoint Operation

For multipoint operation, the transmitter uses the End of Transmission (EOT) sequence, which is the same pattern as the ESCAPE, to allow the receiving terminals to detect the end of multipoint traffic transmission. The EOT sequence is a 256-bit pseudorandom sequence that allows reliable detection in the background of full bandwidth traffic under expected channel conditions. The value of the EOT sequence is specified in Table 2.5-3.

4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867

5.2 Multipoint Session

A multipoint session begins when a SCIP terminal initiates multipoint transmit signaling and ends when an EOT sequence has been transmitted. During a multipoint session, a transmitting terminal transitions from the Multipoint IDLE state to the Multipoint Transmit Secure Traffic state via Multipoint Transmit Signaling, as shown in Figure 5.2-1. Receiving terminals transition from the Multipoint IDLE state to the Multipoint Receive Secure Traffic state via Multipoint Receive Signaling or Late Entry. At the end of the session, all terminals transition back to the Multipoint IDLE state. Section 5.2.1 specifies the multipoint transmit signaling and secure traffic. Section 5.2.2 specifies the reception and processing of multipoint receive signaling and secure traffic.



NOTES

SM = Sync Management frame

SOM = Start of Message

MCS = Multipoint Cryptosync Message

EOT = End of Transmission

MULTIPOINT IDLE = The state the terminal is in when it is not transmitting and not receiving SCIP multipoint traffic. The terminal is waiting for a multipoint transmit request and is also searching for SOMs and SM frames from far-end terminals.

RETURN TO IDLE (From Traffic) = (1) EOT transmitted/received, or
(2) Out-of-sync detected, or
(3) SOM received, or
(4) Timeout waiting for additional traffic

RETURN TO IDLE (From Signaling) = (1) Sync Verification failed, or
(2) Error-free MCS cannot be assembled, or
(3) SM frame received

4868
4869
4870

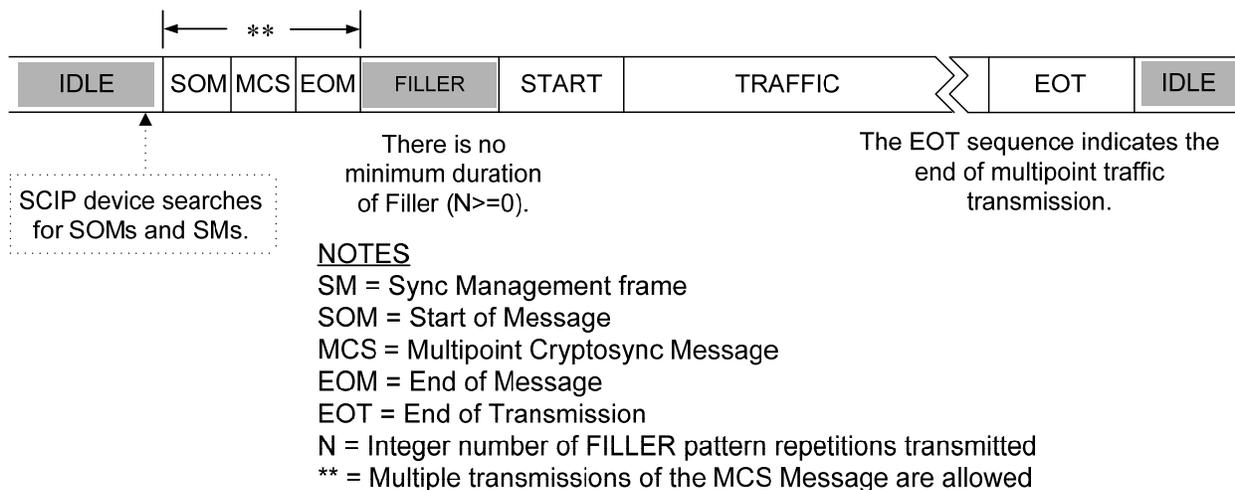
Figure 5.2-1 SCIP Multipoint State Diagram

4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895

5.2.1 Multipoint Transmission

This section specifies SCIP multipoint transmission. It is assumed that a point-to-multipoint digital channel has already been established, using the underlying channel protocols, by means outside the scope of this Signaling Plan. The signaling necessary to establish a SCIP multipoint secure application is then executed over this digital channel. The SCIP terminal transmits the signaling specified in this section to the receiving SCIP terminals to establish the multipoint session.

An example of the overall flow for SCIP multipoint transmit signaling is shown in Figure 5.2-2. During Multipoint IDLE periods, there is no transmission by the SCIP application, though there may actually be transmissions on individual links related to signaling performed by the underlying digital channel protocols. Multipoint IDLE periods are permitted at any time. SCIP multipoint transmit signaling shall begin with the transmission of the MCS Message, as specified in Section 5.2.1.1. If FILLER is transmitted, the pattern shall be sent an integer number of repetitions and follow the MCS Message transmission. START shall follow FILLER, if FILLER is transmitted. Otherwise, START shall follow the MCS Message transmission. Transmission of the START shall precede multipoint full bandwidth TRAFFIC. The transmission of multipoint secure traffic is specified in Section 5.2.1.2. The EOT sequence shall follow multipoint full bandwidth TRAFFIC. The end of multipoint secure traffic transmission is specified in Section 5.2.1.3. The MCS Message format and FILLER, START, and EOT patterns are specified in Section 5.1.



4896
4897
4898

Figure 5.2-2 Multipoint Secure Voice Transmit Signaling Time Line

4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916

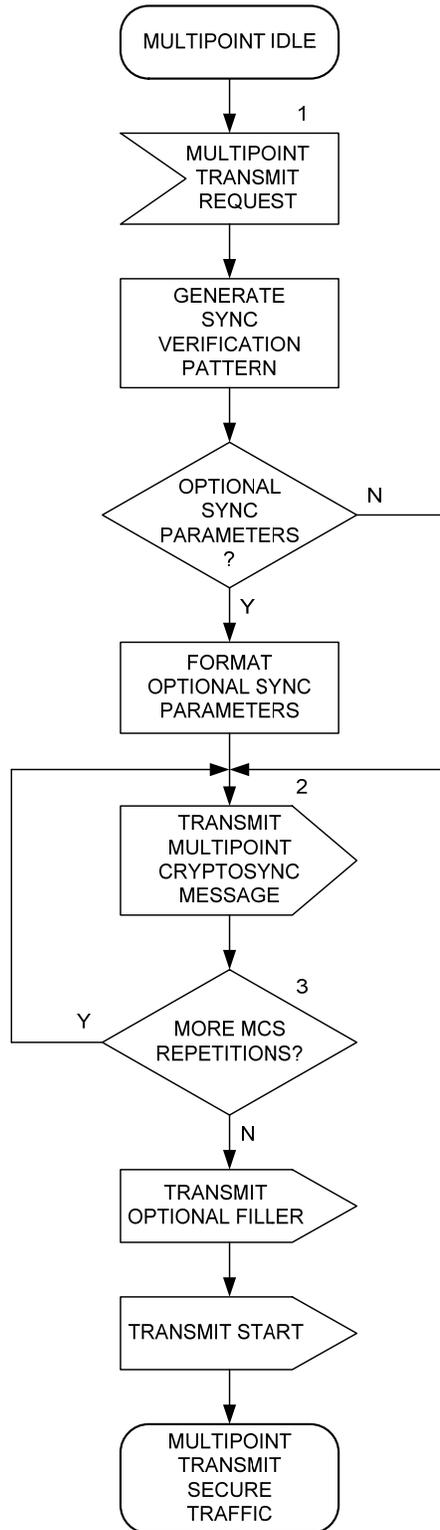
5.2.1.1 Multipoint Cryptosync Message Transmission

MCS Message transmission for SCIP multipoint signaling is shown in Figure 5.2-3. This signaling occurs at the beginning of a SCIP multipoint transmission, and starts from the Multipoint IDLE state (see Figure 5.2-1). Upon receipt of a locally generated Multipoint Transmit Request, the PPK attributes shall be displayed to the user, as defined in SCIP-232, Section 2.2.1 (SCIP-230 and SCIP-231 Sections TBD).

The terminal shall generate an Application IV and a Sync Verification pattern as defined in SCIP-232, Section 3.5.3.1 (SCIP-230 and SCIP-231 Sections TBD). If optional Sync Parameters are to be included in the MCS Message, the terminal shall format the optional Sync Parameters in addition to the mandatory Sync Verification pattern.

The terminal shall transmit the MCS Message, as specified in Sections 5.1.1.1 and 5.1.2, to the receiving terminals followed by optional FILLER. The terminal shall then transmit START and transition to the Multipoint Transmit Secure Traffic state specified in Section 5.2.1.2.

4917



NOTES:

1. Display the PPK attributes after the Multipoint Transmit Request.
2. If there are no optional sync parameters, the entire MCS is transmitted in two RT frames.
3. The MCS message may be transmitted multiple times.

4918

4919

4920

Figure 5.2-3 Multipoint Cryptosync Message Transmission

4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958

5.2.1.2 Multipoint Secure Traffic Transmission

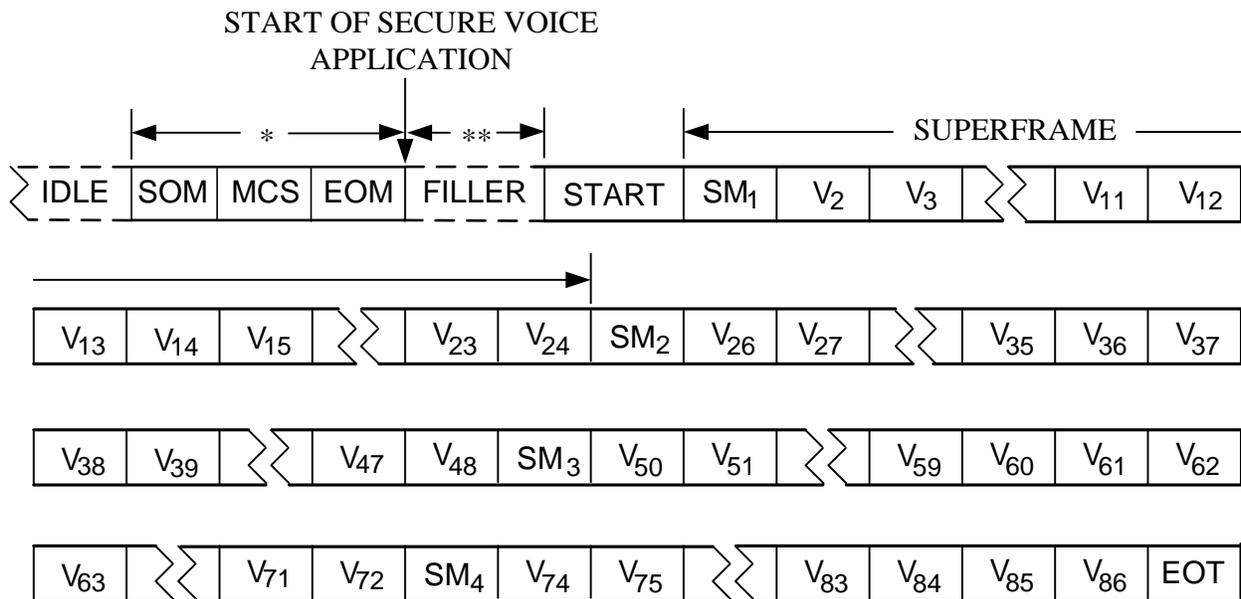
Multipoint secure traffic transmission processing, using the Application IV included in the MCS Message, shall begin after the terminal transitions to the Multipoint Transmit Secure Traffic state (see Figure 5.2-1). The Secure MELP Voice application is specified in Section 5.2.1.2.1 for multipoint secure traffic transmission. The Secure G.729D Voice (see Section 5.2.1.2.2) and Secure Data (see Section 5.2.1.2.3) applications are TBSL.

5.2.1.2.1 Multipoint Secure MELP Voice Transmission

Secure 2400 bps Blank and Burst MELP Voice with FCT is used for multipoint operation. A Sync Management frame is substituted periodically for a vocoder frame. The vocoder frame that would normally have been transmitted during the Sync Management frame transmission interval is discarded. The Sync Management frame contains information that allows late-entry cryptographic synchronization as well as cryptographic synchronization maintenance. The MELP vocoder is run continuously, and all frames that are generated (excluding blanked frames) are transmitted. DTX operation (see Section 3.3.1.4) is not supported for multipoint operation.

Secure 2400 bps Blank and Burst MELP Voice shall be transmitted in a "superframe" consisting of a 54-bit Sync Management frame followed by 23 54-bit MELP vocoder frames, except when shortened by the transmission of an EOT to end multipoint traffic transmission. The contents of the 54-bit MELP vocoder frame, representing 22.5 msec. of speech, shall be as specified in MIL-STD-3005 or NATO STANAG 4591. The MELP encryption and transmission bit ordering shall be the same as for point-to-point operation. The alternating 1/0 sync bit in the first MELP vocoder frame transmitted may have either value, and the receiver must be prepared to accept either value.

An example of multipoint Secure 2400 bps Blank and Burst MELP Voice transmission is shown in Figure 5.2-4. Secure traffic shall begin with a START and end with an EOT. MELP and Sync Management frames shall be transmitted between the START and EOT. Note that the superframe always begins with a Sync Management frame. Note also that the first vocoder frame shall be discarded (blanked) and replaced by a Sync Management frame. In all cases, however, the first MELP frame actually transmitted in a superframe is encrypted using the second half of the first state vector value for that superframe.



NOTES

SM = Sync Management frame

V = MELP Vocoder frame

MCS = Multipoint Cryptosync Message

N = Integer number of FILLER pattern repetitions transmitted

EOT = End of Transmission (does not need to be transmitted on a superframe boundary)

* = Multiple transmissions of the MCS Message are allowed

** = There is no minimum duration of FILLER (N>=0)

4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977

Figure 5.2-4 Multipoint MELP Voice Transmission Format – Blank and Burst

The Sync Management frame specified in Section 3.3.1.1.1 and encryption and transmission ordering specified in Section 3.3.1.1.2 for Secure 2400 bps Blank and Burst MELP Voice shall apply to multipoint operation.

5.2.1.2.2 Multipoint Secure G.729D Voice Transmission

The transmit format of Multipoint Secure G.729D Voice is TBSL.

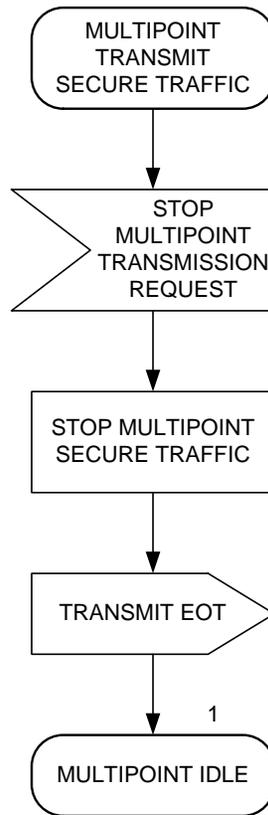
5.2.1.2.3 Multipoint Secure Data Transmission

The transmit format of Multipoint Secure Data is TBSL.

4978
4979
4980
4981
4982
4983
4984
4985
4986
4987

5.2.1.3 End of Multipoint Secure Traffic Transmission

The end of a SCIP multipoint secure traffic transmission is shown in Figure 5.2-5. Upon receipt of a locally generated request to stop multipoint transmission, the terminal shall cease transmitting SCIP multipoint secure traffic, transmit an EOT sequence to end the multipoint session, and transition to the Multipoint IDLE state (see Figure 5.2-1). Upon transition to the Multipoint IDLE state, the terminal shall remove the PPK attributes from the display.



NOTE:

1. Remove the PPK attributes from the display.

4988
4989
4990

Figure 5.2-5 End of Multipoint Secure Traffic Transmission

4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031

5.2.2 Multipoint Reception

If the entire MCS Message is received correctly, receiving terminals shall verify proper operation of the cryptography and wait for the START. The reception and processing of the MCS Message are specified in Section 5.2.2.1. The reception and processing of multipoint secure full bandwidth traffic are specified in Section 5.2.2.2.

If the entire MCS Message is not received or not received correctly, then cryptographic synchronization may be achieved through Late Entry as specified in Section 5.2.2.3.

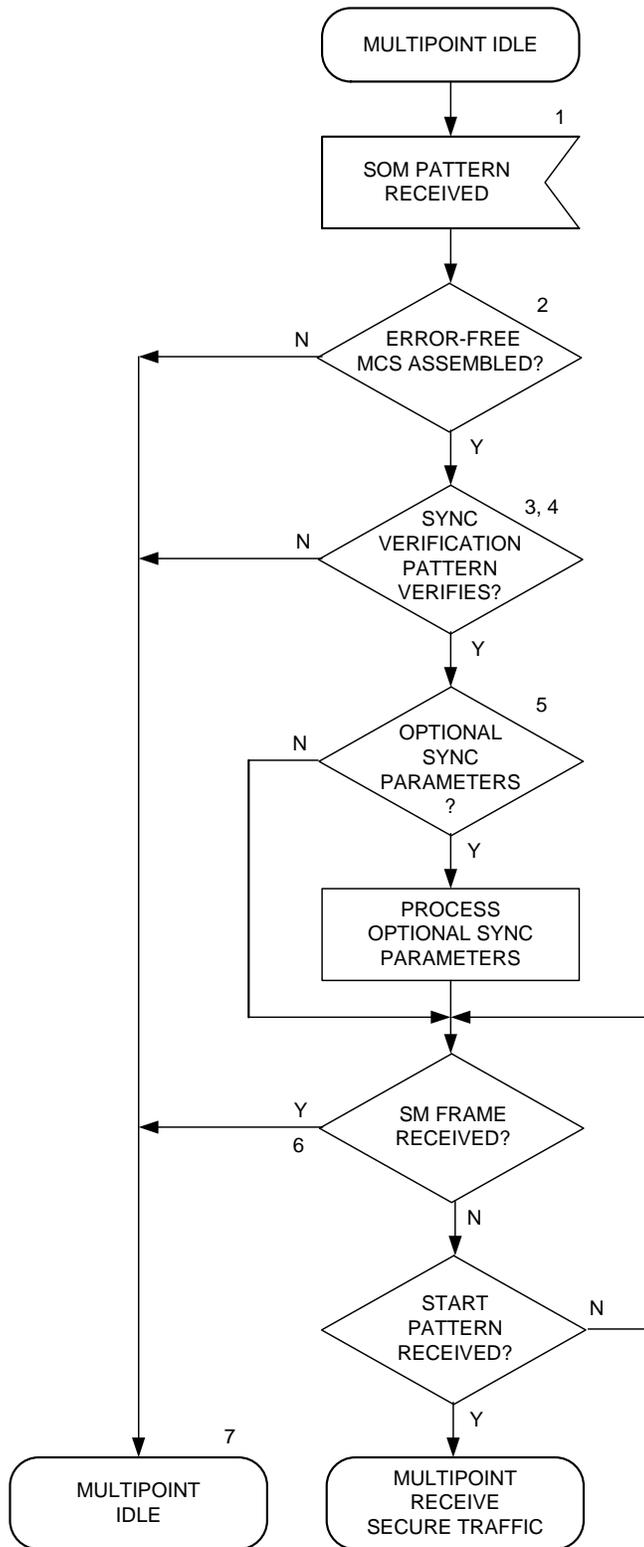
The end of multipoint secure traffic reception is specified in Section 5.2.2.4.

5.2.2.1 Multipoint Cryptosync Message Reception

MCS Message reception during SCIP multipoint signaling is shown in Figure 5.2-6. This signaling occurs at the beginning of a SCIP multipoint reception, and starts from the Multipoint IDLE state (see Figure 5.2-1). The receiving terminals shall process a received MCS Message, as specified in Sections 5.1.1.2 and 5.1.2. Receiving terminals may use the Frame Count to identify the frames that were received correctly, the frames that were received with errors, and the frames that were received multiple times. This information allows the receiving terminals to determine if one error-free copy of the MCS Message has been received or can be assembled.

When an error-free MCS Message is received or assembled, the receiving terminals shall verify the Sync Verification pattern contained in the MCS Message, as specified in SCIP-232, Section 3.5.3.2 (SCIP-230 and SCIP-231 Sections TBD). When the Sync Verification pattern has been verified, the PPK attributes shall be displayed to the user, as specified in SCIP-232, Section 2.2.1 (SCIP-230 and SCIP-231 Sections TBD). The receiving terminals shall then process any optional Sync Parameters that are contained in the MCS Message. If a Sync Parameter ID is not supported, the receiving terminals shall ignore the Sync Parameter and process any remaining Sync Parameter IDs. Upon receipt of the START, the receiving terminals shall transition to the Multipoint Receive Secure Traffic state specified in Section 5.2.2.2.

When an error-free MCS Message cannot be assembled, Sync Verification fails, or a Sync Management frame is received, the receiving terminals shall transition to the Multipoint IDLE state and execute Late Entry (see Figure 5.2-1). The Sync Management frames inserted in the traffic shall be used to achieve Late Entry cryptographic synchronization, as specified in Section 5.2.2.3.



NOTES:

1. SOM marks the beginning of the MCS Message that may be received multiple times.
2. Frame Count is used to identify frames that are error-free. The receivers of multiple MCS messages assemble error-free MCS frames as they are received.
3. Verification of the Sync Verification pattern is defined in SCIP-23x.
4. Display the PPK attributes after the Sync Verification pattern is verified.
5. If a receiving terminal does not recognize a Sync Parameter ID, it ignores it and processes any remaining Sync Parameter ID(s).
6. The START was not detected.
7. Late Entry will be executed.

5032
5033
5034

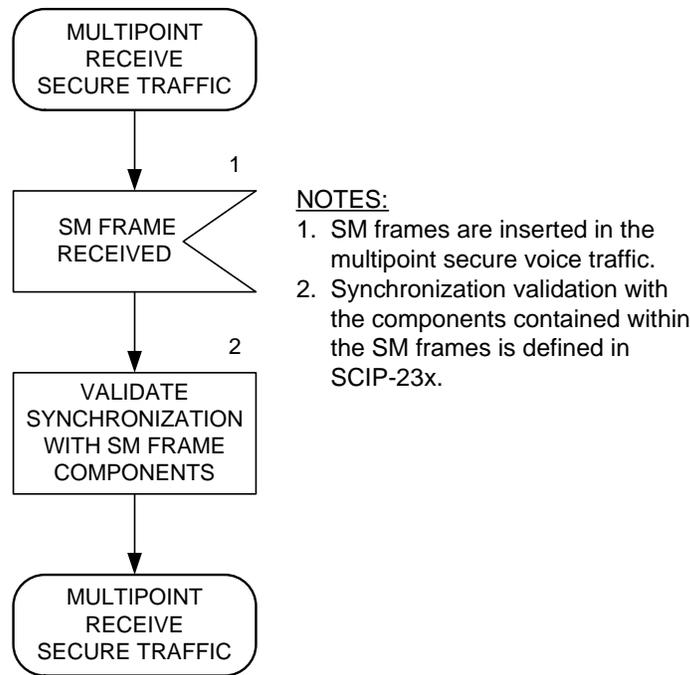
Figure 5.2-6 Multipoint Cryptosync Message Reception

5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052

5.2.2.2 Multipoint Secure Traffic Reception

Multipoint secure traffic reception processing shall begin after the terminal transitions to the Multipoint Receive Secure Traffic state (see Figure 5.2-1). The Application IV included in the MCS Message (see Section 5.2.2.1) or assembled from the components in the Sync Management frames (see Section 5.2.2.3) shall be used for decryption. The Secure MELP Voice application is specified in Section 5.2.2.2.1 for multipoint secure traffic reception. The Secure G.729D Voice (see Section 5.2.2.2.2) and Secure Data (see Section 5.2.2.2.3) applications are TBSL.

Multipoint secure voice traffic reception processing is shown in Figure 5.2-7. Receiving terminals shall process multipoint secure traffic frames when they are received. If a received frame is a Sync Management frame, the receiving terminals shall validate synchronization using the Partial Long and Short components contained within the Sync Management frame, as specified in SCIP-23x for each application. This process is used to maintain cryptographic synchronization.



5053
5054
5055
5056

Figure 5.2-7 Multipoint Secure Voice Traffic Reception

5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098

5.2.2.2.1 Multipoint Secure MELP Voice Reception

Upon receipt of a START (see Section 5.2.2.1), or upon assembling the Long and Short Components from the Sync Management frames during Late Entry (see Section 5.2.2.3), receiving terminals shall begin decrypting multipoint full bandwidth traffic. The superframe structure for multipoint secure MELP voice traffic is shown in Figure 5.2-4. Superframe alignment must be established in order to decrypt the secure MELP voice frames. The Sync Management frame specified in Section 3.3.1.1.1 and decryption and reception ordering specified in Section 3.3.1.1.2 for Secure 2400 bps Blank and Burst MELP Voice shall apply to multipoint operation.

5.2.2.2.2 Multipoint Secure G.729D Voice Reception

TBSL

5.2.2.2.3 Multipoint Secure Data Reception

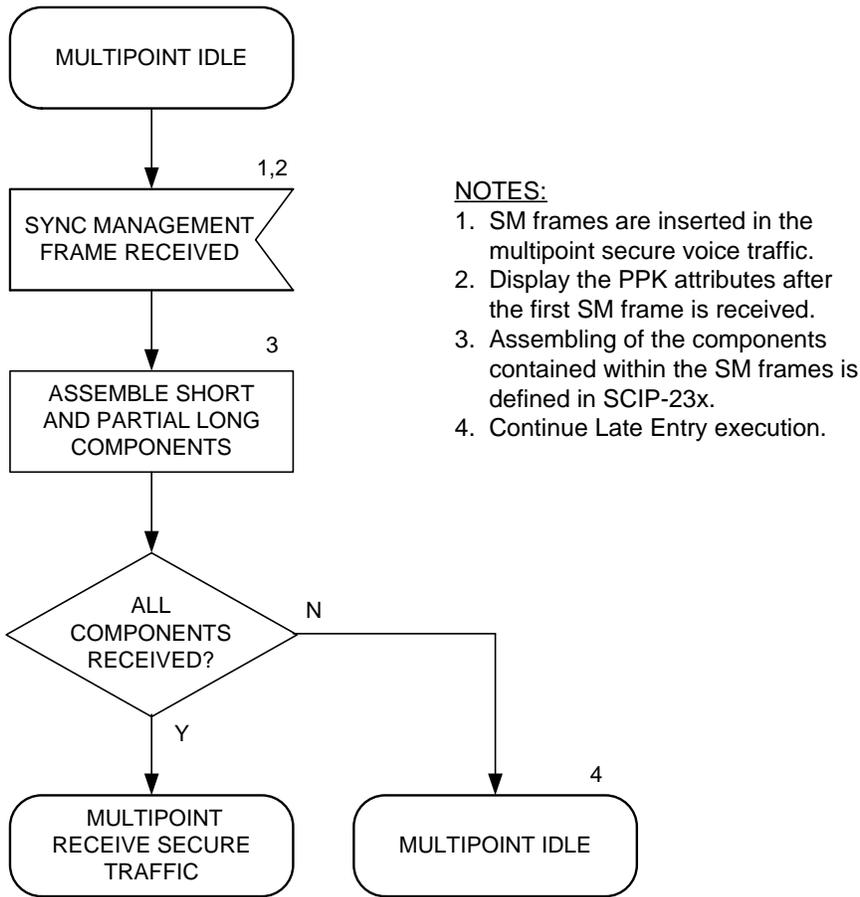
TBSL

5.2.2.3 Late Entry (Including Re-Entry)

Late Entry cryptographic synchronization during SCIP multipoint signaling is shown in Figure 5.2-8. This signaling occurs when receiving terminals start receiving secure multipoint full bandwidth traffic without first receiving and successfully processing the MCS Message. This signaling starts from the Multipoint IDLE state (see Figure 5.2-1). The receiving terminals shall search for the Sync Management frames inserted in multipoint full bandwidth traffic. When the first Sync Management frame has been received, the PPK attributes shall be displayed to the user, as defined in SCIP-232, Section 2.2.1 (SCIP-230 and SCIP-231 Sections TBD). In order to achieve cryptographic synchronization, the Partial Long Components and Short Component contained within the Sync Management frames shall be assembled as specified in SCIP-232, Section 4.1.1.2 (SCIP-230 and SCIP-231 Sections TBD). The receiving terminals shall then transition to the Multipoint Receive Secure Traffic state specified in Section 5.2.2.2.

Re-entry cryptographic synchronization follows the same process as Late Entry cryptographic synchronization, with one exception. In Re-entry, the Short Component is usually sufficient to re-establish cryptographic synchronization. Re-entry is executed when synchronization, after initially being established, is lost during the session.

5099
5100
5101
5102
5103
5104



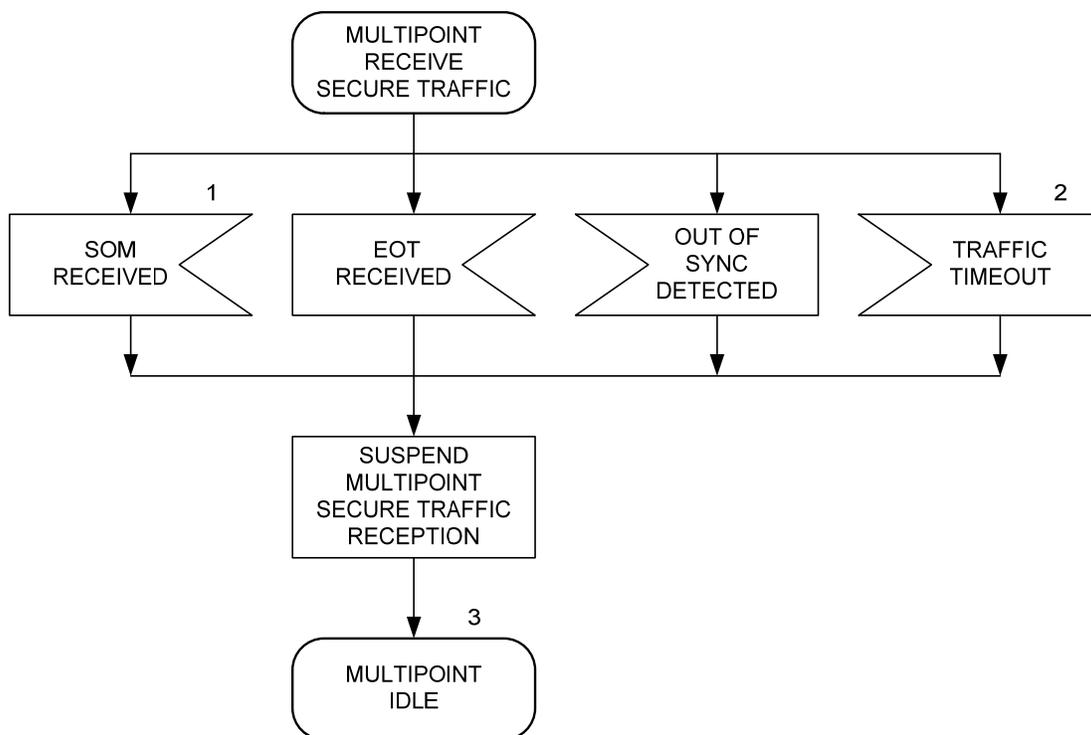
5105
5106
5107
5108

Figure 5.2-8 Multipoint Late Entry Cryptographic Synchronization

5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120

5.2.2.4 End of Multipoint Secure Traffic Reception

The end of SCIP multipoint secure traffic reception is shown in Figure 5.2-9. Upon receipt of an EOT or SOM, detection of loss of synchronization, or a timeout waiting for additional traffic, the receiving terminals shall suspend multipoint secure traffic reception and transition to the Multipoint IDLE state (see Figure 5.2-1). An SOM, which begins a new MCS Message, may be received if an EOT was not detected. A timeout may also be implemented to guarantee a transition to the Multipoint IDLE state, if an EOT is not detected. Upon transition to the Multipoint IDLE state, the receiving terminals shall remove the PPK attributes from the display.



NOTE:

1. An EOT was not detected.
2. A timeout may be implemented to guarantee a transition to Multipoint IDLE, if an EOT is not detected.
3. Remove the PPK attributes from the display.

5121
5122
5123
5124

Figure 5.2-9 End of Multipoint Secure Traffic Reception

5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150

THIS PAGE INTENTIONALLY LEFT BLANK.

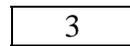
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169

APPENDICES

A.0 SCIP MESSAGE TRANSPORT PROTOCOL EXAMPLES

This appendix provides several examples of the operation of the SCIP message transport control protocol. It contains no requirements. These examples are used to show messages from the Message Layer of Terminal A being sent to the Message Layer of Terminal B. Messages may be transferred in the opposite direction simultaneously; however, for clarity this is not shown. The transmit directions operate independently.

The following notation is used in the examples shown in this appendix.



Frame #3



Frame #5 received with uncorrectable errors



Lost information



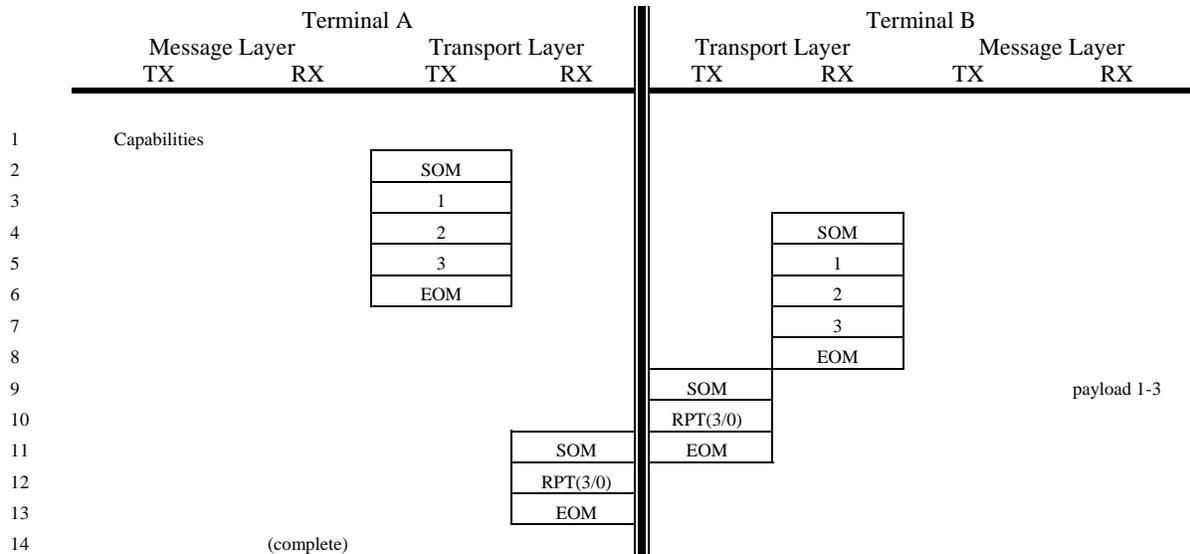
Application Timer running

RPT(4/5,30)

REPORT message acknowledging Block #4 and
requesting resend of Block #5 & Block #30

5170
5171
5172

A.1 Normal Capabilities Message Transfer



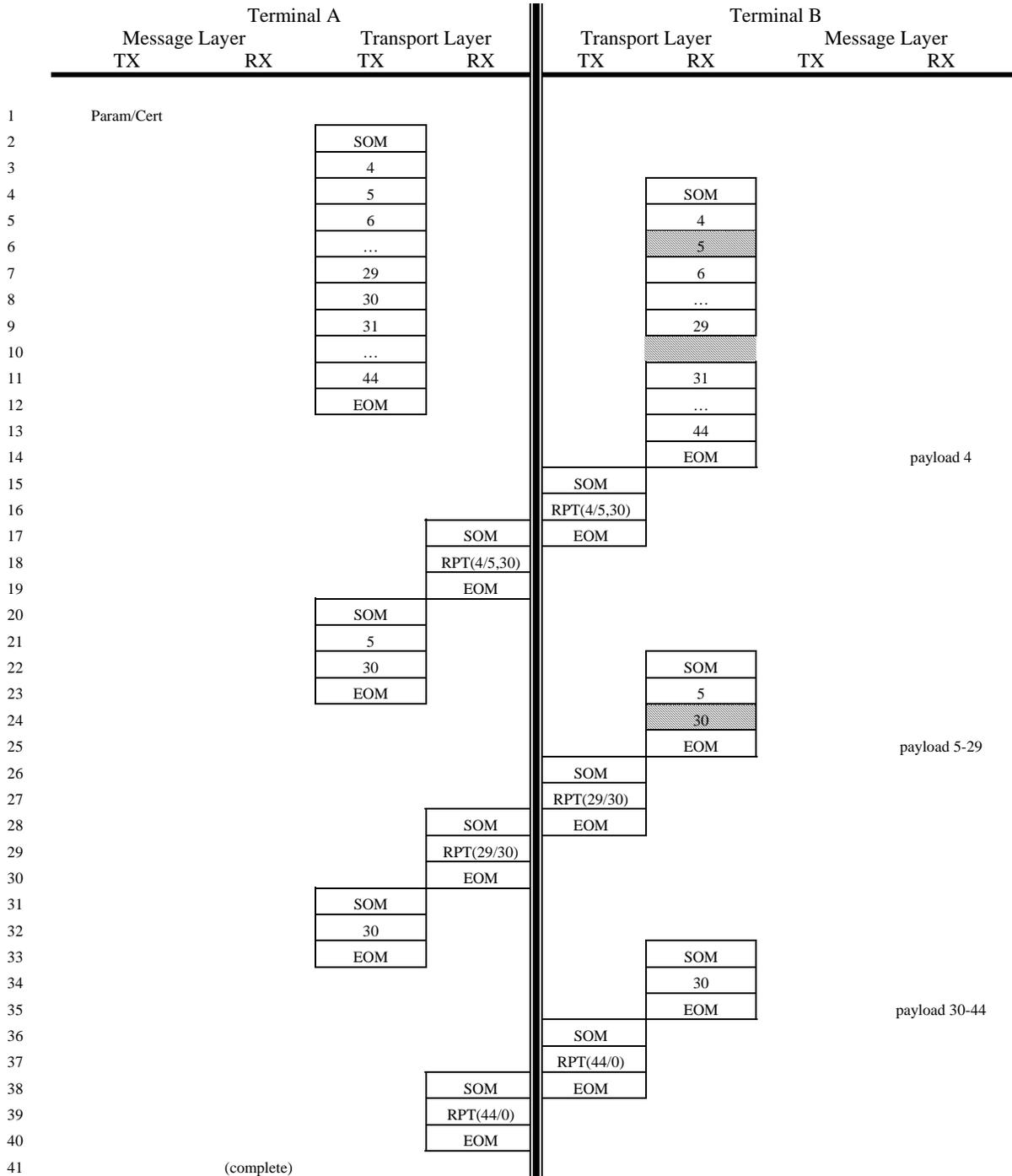
5173
5174

- 5175 1. The Message Layer at Terminal A determines that a CAPABILITIES message needs to be
- 5176 sent. This example assumes that the CAPABILITIES Message sent from the Message Layer
- 5177 at Terminal A is between 27 and 39 octets long, resulting in 3 frames at the Transport Layer.
- 5178 2. The Transport Layer at Terminal A receives the CAPABILITIES message from the Message
- 5179 Layer, divides it into 3 frames, and begins by sending SOM.
- 5180 3. Terminal A sends frame 1 & stores a local copy for possible retransmission.
- 5181 4. Terminal A sends frame 2 & stores a local copy for possible retransmission. Terminal B
- 5182 receives SOM, indicating an incoming message.
- 5183 5. Terminal A sends frame 3 & stores a local copy for possible retransmission. Terminal B
- 5184 receives frame 1.
- 5185 6. Terminal A sends EOM since all frames of the Capabilities message have been sent.
- 5186 Terminal B receives frame 2.
- 5187 7. Terminal B receives frame 3.
- 5188 8. Terminal B receives EOM, indicating that the incoming message is complete.
- 5189 9. Terminal B knows of no outstanding frames and therefore will acknowledge frames 1
- 5190 through 3. A SOM is sent to frame the REPORT. Terminal B concatenates the payload data
- 5191 from received frames 1-3 and passes it to the Message Layer, which determines that it forms
- 5192 a valid Capabilities message.
- 5193 10. Terminal B sends REPORT indicating that all frames up to and including frame 3 have been
- 5194 received correctly.
- 5195 11. Terminal A receives SOM, indicating a new incoming message. Terminal B sends EOM,
- 5196 indicating the end of the REPORT.
- 5197 12. Terminal A receives REPORT indicating that frames up to and including frame 3 have been
- 5198 received correctly. Terminal A may now delete its local copy of transmitted frames 1-3 since
- 5199 it knows that no further retransmissions of these frames will be necessary.
- 5200 13. Terminal A receives EOM, indicating the end of the received REPORT.

5201 14. If necessary, the Transport Layer at Terminal A may inform the Message Layer that the
5202 CAPABILITIES message has been successfully transported.
5203

5204
5205
5206

A.2 Parameters/Certificate Message Transfer with Corrupted and Missing Frames



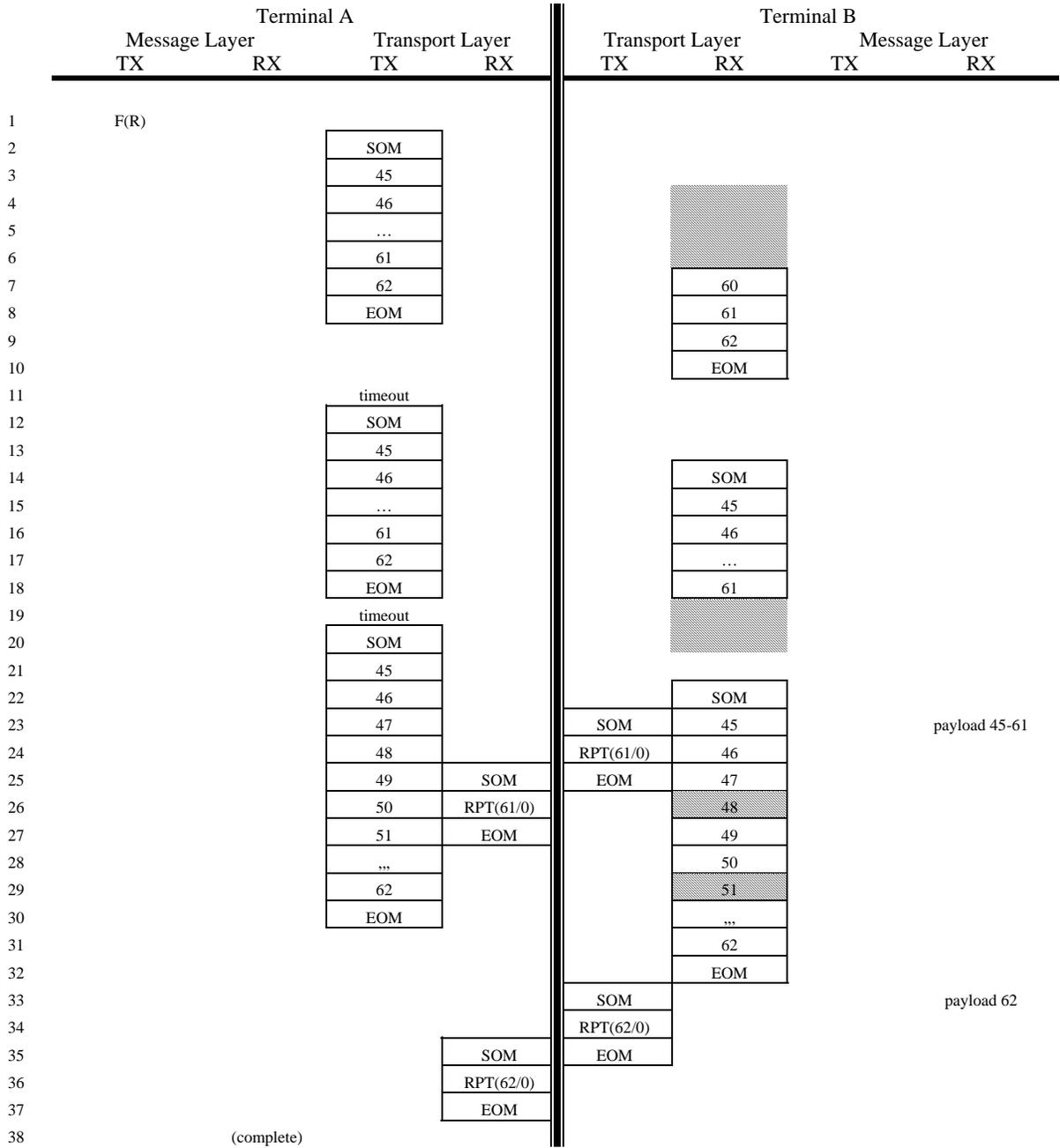
5207

- 5208
- 5209 1. The Message Layer at Terminal A determines that a PARAMETERS/CERTIFICATE
5210 message needs to be sent. This example assumes that the PARAMETERS/CERTIFICATE
5211 Message sent from the Message Layer at Terminal A is between 521 and 533 octets long,
5212 resulting in 41 frames at the Transport Layer.
- 5213 2. The Transport Layer at Terminal A receives the PARAMETERS/CERTIFICATE message
5214 from the Message Layer, divides it into 41 frames, and begins by sending SOM. This
5215 example assumes that the most recent transmitted frame was frame 3, so the first frame is
5216 assigned frame number 4.
- 5217 3. Terminal A sends frame 4 & stores a local copy for possible retransmission.
- 5218 4. Terminal A sends frame 5 & stores a local copy for possible retransmission. Terminal B
5219 receives SOM, indicating an incoming message.
- 5220 5. Terminal A sends frame 6 & stores a local copy for possible retransmission. Terminal B
5221 receives frame 4.
- 5222 6. Terminal A sends frames 7 through 28 & stores local copies for possible retransmission.
5223 Terminal B receives frame 5 which is corrupted (i.e. CRC failure). Terminal B may
5224 immediately send a REPORT message indicating that frame 5 needs to be retransmitted or,
5225 as is shown in this example, store up all retransmission requests until the end of the incoming
5226 message.
- 5227 7. Terminal A sends frame 29 & stores a local copy for possible retransmission. Terminal B
5228 receives frame 6.
- 5229 8. Terminal A sends frame 30 & stores a local copy for possible retransmission. Terminal B
5230 receives frames 7 through 28.
- 5231 9. Terminal A sends frame 31 & stores a local copy for possible retransmission. Terminal B
5232 receives frame 29.
- 5233 10. Terminal A sends frames 32 through 43 & stores local copies for possible retransmission.
5234 Note that in this example frame 30 is lost in transmission and does not arrive at Terminal B.
- 5235 11. Terminal A sends frame 44 & stores a local copy for possible retransmission. Terminal B
5236 receives frame 31. Terminal B was expecting frame 30 and therefore adds frame 30 to the
5237 list of frames to be included on the NAK list in the REPORT message.
- 5238 12. Terminal A sends EOM since all frames of the PARAMETERS/CERTIFICATE message
5239 have been sent. Terminal B receives frames 32 through 43.
- 5240 13. Terminal B receives frame 44.
- 5241 14. Terminal B receives EOM indicating that the incoming message is complete. Terminal B
5242 passes data from all correctly received contiguous frames to the Message Layer, in this
5243 example from frame 4 only. The Message Layer is responsible for checking length fields
5244 and realizing that this is only a partial PARAMETERS/CERTIFICATE message.
- 5245 15. Terminal B sends SOM to frame the REPORT.
- 5246 16. Terminal B sends REPORT indicating that up through frame 4 has been received correctly
5247 while frames 5 and 30 need to be retransmitted.
- 5248 17. Terminal A receives SOM indicating the beginning of an incoming message. Terminal B
5249 sends EOM to frame the REPORT.
- 5250 18. Terminal A receives REPORT indicating that up through frame 4 has been received correctly
5251 and requesting that frames 5 and 30 be retransmitted. Terminal A may now delete its local
5252 copy of transmitted frame 4 since it knows that no further retransmissions of this frame will
5253 be necessary.

- 5254 19. Terminal A receives EOM indicating that the incoming REPORT is complete.
- 5255 20. Terminal A sends SOM to frame the retransmitted frames.
- 5256 21. Terminal A retransmits frame 5.
- 5257 22. Terminal A retransmits frame 30. Terminal B receives SOM indicating the beginning of an
5258 incoming message.
- 5259 23. Terminal A sends EOM to frame the retransmitted frames. Terminal B receives frame 5.
- 5260 24. Terminal B receives frame 30, which in this example is corrupted (CRC failure).
- 5261 25. Terminal B receives EOM indicating that the incoming message is complete. Terminal B
5262 passes data from all correctly received contiguous frames to the Message Layer, in this
5263 example from frames 5 through 29. The Message Layer is responsible for checking length
5264 fields and realizing that this is still only a partial PARAMETERS/CERTIFICATE message.
- 5265 26. Terminal B sends SOM to frame the REPORT.
- 5266 27. Terminal B sends REPORT indicating that up through frame 29 has been received correctly
5267 while frame 30 needs to be retransmitted
- 5268 28. Terminal A receives SOM indicating the beginning of an incoming message. Terminal B
5269 sends EOM to frame the REPORT.
- 5270 29. Terminal A receives REPORT indicating that up through frame 29 has been received
5271 correctly and requesting that frame 30 be retransmitted. Terminal A may now delete its local
5272 copy of transmitted frames 5-29 since it knows that no further retransmissions of these
5273 frames will be necessary.
- 5274 30. Terminal A receives EOM indicating that the incoming REPORT is complete.
- 5275 31. Terminal A sends SOM to frame the retransmitted frames.
- 5276 32. Terminal A retransmits frame 30.
- 5277 33. Terminal A sends EOM to frame the retransmitted frame. Terminal B receives SOM
5278 indicating the beginning of an incoming message.
- 5279 34. Terminal B receives frame 30.
- 5280 35. Terminal B receives EOM indicating that the incoming message is complete. Terminal B
5281 passes data from all correctly received contiguous frames to the Message Layer, in this
5282 example from frames 30 through 44. The Message Layer is responsible for checking length
5283 fields and realizing that the PARAMETERS/CERTIFICATE message is now complete.
- 5284 36. Terminal B knows of no more outstanding frames and will therefore respond to the received
5285 EOM by sending a REPORT message containing only an acknowledge frame value.
5286 Terminal B sends SOM to frame the REPORT.
- 5287 37. Terminal B sends REPORT indicating that all frames up to and including frame 44 have been
5288 received correctly.
- 5289 38. Terminal A receives SOM, indicating a new incoming message. Terminal B sends EOM
5290 which frames the REPORT. Terminal A receives SOM indicating an incoming message.
- 5291 39. Terminal A receives REPORT indicating that frames up to and including frame 44 have been
5292 received correctly. Terminal A may now delete its local copy of transmitted frames 30-44
5293 since it knows that no further retransmissions of these frames will be necessary.
- 5294 40. Terminal A receives EOM, indicating the end of the received REPORT.
- 5295 41. If necessary, the Transport Layer may inform the Message Layer that the
5296 PARAMETERS/CERTIFICATE message has been successfully transported.
- 5297

5298
5299
5300

A.3 F(R) Message Transfer with Corrupted SOM and EOM Sequences



5301

- 5302
- 5303 1. The Message Layer at Terminal A determines that a F(R) message needs to be sent. This
- 5304 example assumes that the F(R) Message sent from the Message Layer at Terminal A is
- 5305 between 222 and 223 octets long, resulting in 18 frames at the Transport Layer.
- 5306 2. The Transport Layer at Terminal A receives the F(R) message from the Message Layer,
- 5307 divides it into 18 frames, and begins by sending SOM. This example assumes that the most
- 5308 recent transmitted frame was frame 44, so the first frame is assigned frame number 45.
- 5309 3. Terminal A sends frame 45 & stores a local copy for possible retransmission.
- 5310 4. Terminal A sends frame 46 & stores a local copy for possible retransmission. Terminal B
- 5311 should have received SOM at this time, but this example assumes that the SOM and first
- 5312 frames are not received.
- 5313 5. Terminal A sends frames 47 through 60 & stores a local copy for possible retransmission
- 5314 6. Terminal A sends frame 61 and stores a local copy for possible retransmission.
- 5315 7. Terminal A sends frame 62 and stores a local copy for possible retransmission. Terminal B
- 5316 receives frame 60, although it isn't recognized because it was not preceded by SOM
- 5317 8. Terminal A sends EOM since all frames of the F(R) message have been sent. Terminal B
- 5318 receives frame 61, although it isn't recognized because it was not preceded by SOM.
- 5319 9. Terminal B receives frame 62, although it isn't recognized because it was not preceded by
- 5320 SOM.
- 5321 10. Terminal B receives EOM without having seen SOM. As a local implementation option,
- 5322 Terminal B can work backwards from the EOM to identify missing frames based on which
- 5323 frames were expected (in this example frames 45 through 59 were missing) and then send
- 5324 REPORT with the NAK list indicating the missing frames. Another valid approach is for
- 5325 Terminal B to ignore the entire received message since it was not preceded by SOM. This
- 5326 more simplistic approach is shown in this example.
- 5327 11. The retransmission timeout at Terminal A expires because Terminal A has not received
- 5328 REPORT in response to the frames it transmitted. Terminal A will retransmit frames 45
- 5329 through 62.
- 5330 12. Terminal A begins the retransmission with SOM.
- 5331 13. Terminal A retransmits frame 45.
- 5332 14. Terminal A retransmits frame 46. Terminal B receives SOM, indicating an incoming
- 5333 message.
- 5334 15. Terminal A retransmits frames 47 through 60. Terminal B receives frame 45.
- 5335 16. Terminal A retransmits frame 61. Terminal B receives frame 46.
- 5336 17. Terminal A retransmits frame 62. Terminal B receives frames 47 through 60.
- 5337 18. Terminal A sends EOM since all frames have been retransmitted. Terminal B receives frame
- 5338 61.
- 5339 19. Terminal B stops receiving frames without having seen an EOM. As a local implementation
- 5340 option, Terminal B may timeout and send REPORT indicating all contiguously received
- 5341 frames (through frame 61 in this example). An alternative valid approach is for Terminal B
- 5342 to not send REPORT since EOM was not seen. This more simplistic approach is shown in
- 5343 this example. The retransmission timeout at Terminal A expires because Terminal A has not
- 5344 received REPORT in response to the frames it transmitted. Terminal A will retransmit
- 5345 frames 45 through 62.
- 5346 20. Terminal A begins the retransmission with SOM.
- 5347 21. Terminal A retransmits frame 45.

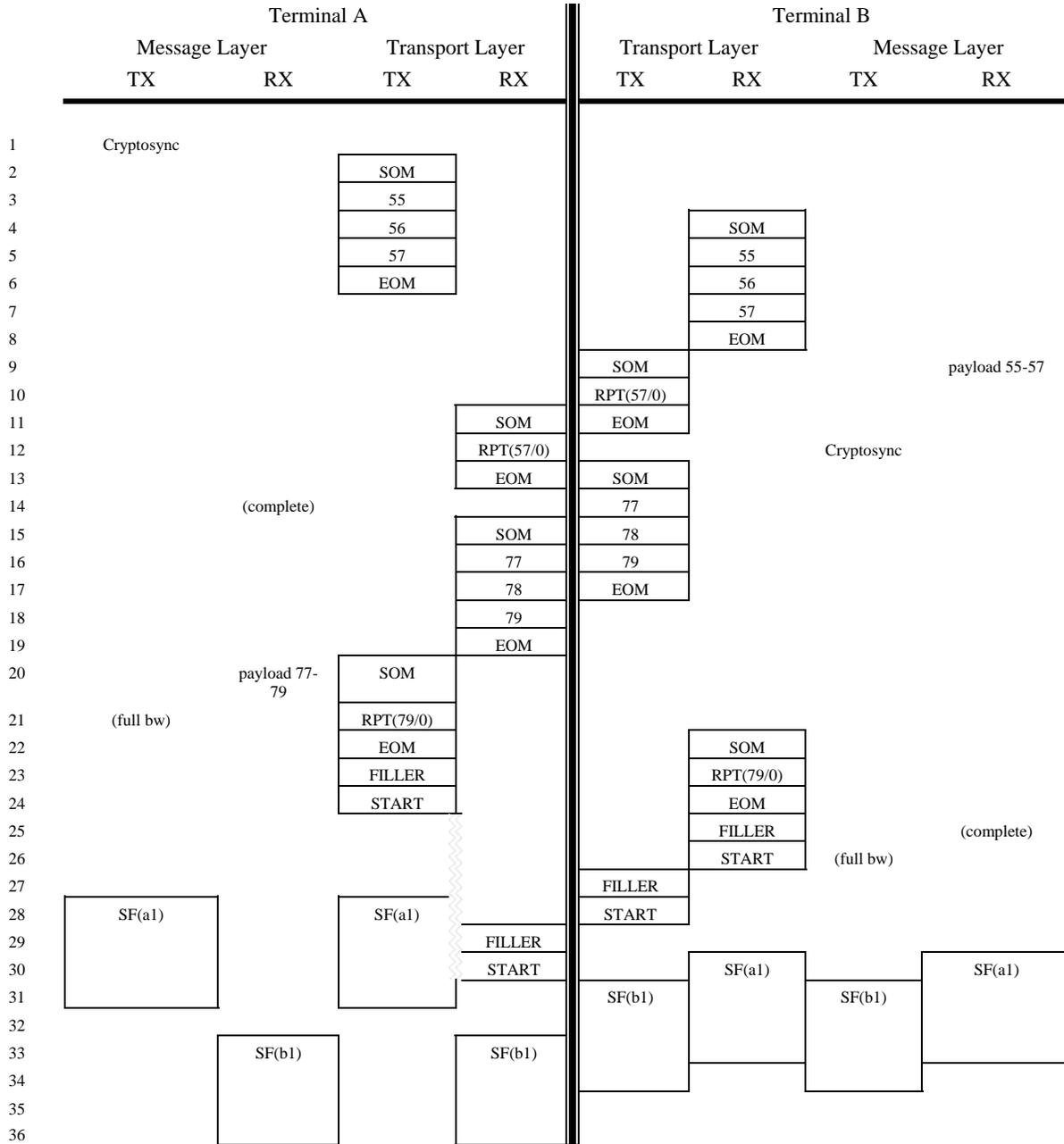
- 5348 22. Terminal A retransmits frame 46. Terminal B receives SOM, indicating an incoming
5349 message. Terminal B was expecting EOM, so is required to send REPORT in response to
5350 the out-of-sequence SOM.
- 5351 23. Terminal A retransmits frame 47. Terminal B sends SOM to frame the outgoing REPORT.
5352 The data from correctly received frames at Terminal B (frames 45 through 61 in this
5353 example) is passed to the Message Layer at Terminal B. Terminal B receives frame 45 but
5354 ignores it since it has already been received correctly.
- 5355 24. Terminal A retransmits frame 48. Terminal B sends REPORT indicating that all frames up
5356 through 61 have been received correctly. Terminal B receives frame 46 but ignores it since it
5357 has already been received correctly.
- 5358 25. Terminal A retransmits frame 49. Terminal A receives SOM, indicating an incoming
5359 message. Terminal B sends EOM. Terminal B receives frame 47 but ignores it since it has
5360 already been received correctly.
- 5361 26. Terminal A retransmits frame 50 and receives REPORT indicating that Terminal B has
5362 received frames through 61 correctly. Terminal B receives frame 48 but ignores it since it
5363 has already been received correctly. Note that even though frame 48 was received with
5364 uncorrectable errors, it is not added to the NAK list since it has previously been received
5365 correctly.
- 5366 27. Terminal A has received an acknowledge for frames up through 61, so it could skip up to that
5367 point in the frames it is resending. This example, however, shows the case of Terminal A
5368 continuing the sequence of frames it had started to transmit. Terminal A retransmits frame
5369 51 and receives EOM. Terminal B receives frame 49 but ignores it since it has already been
5370 received correctly.
- 5371 28. Terminal A retransmits frames 52 through 61. Terminal B receives frame 50 but ignores it
5372 since it has already been received correctly.
- 5373 29. Terminal A retransmits frame 62. Terminal B receives frame 51 but ignores it since it has
5374 already been received correctly. Note that even though frame 51 was received with
5375 uncorrectable errors, it is not added to the NAK list since it has previously been received
5376 correctly.
- 5377 30. Terminal A sends EOM indicating the end of the message. Terminal B receives frames 52
5378 through 61 but ignores them since they have already been received correctly.
- 5379 31. Terminal B receives frame 62.
- 5380 32. Terminal B receives EOM indicating the end of the message.
- 5381 33. Terminal B sends SOM to frame the outgoing REPORT. Terminal B passes to the Message
5382 Layer all contiguously received data not previously passed to the Message Layer (in this
5383 example, only information from frame 62 is passed at this point).
- 5384 34. Terminal B sends REPORT indicating that frames up to and including frame 62 have been
5385 received correctly.
- 5386 35. Terminal A receives SOM, indicating an incoming message. Terminal B sends EOM.
- 5387 36. Terminal A receives REPORT indicating that frames through 62 have been received at
5388 Terminal B.
- 5389 37. Terminal A receives EOM.
- 5390 38. If necessary, the Transport Layer may inform the Message Layer that the F(R) message has
5391 been successfully transported.
- 5392

- 5405 4. Terminal A sends frame 2 & stores a local copy for possible retransmission. Terminal B
5406 receives SOM indicating the beginning of an incoming message.
- 5407 5. Terminal A sends frame 3 & stores a local copy for possible retransmission. Terminal B
5408 receives frame 1.
- 5409 6. The Transport Layer at Terminal A has sent the entire message it received from the Message
5410 Layer so it sends EOM. Terminal B receives frame 2, which is corrupt (CRC failure).
- 5411 7. Terminal B receives frame 3.
- 5412 8. Terminal B receives EOM. Terminal B knows that there is a missing frame in the received
5413 sequence so it will add the missing frame number to the NAK list in the REPORT message.
- 5414 9. Terminal B sends SOM in preparation for sending REPORT. Terminal B also passes along
5415 to the Message Layer the data from all contiguously received frames (only frame 1 in this
5416 example).
- 5417 10. Terminal B sends REPORT indicating that up through frame 1 has been received correctly
5418 and requesting that that frame 2 be retransmitted.
- 5419 11. Terminal B sends EOM. Terminal A should begin receiving the SOM at this point, but this
5420 example assumes that the entire REPORT message, including the SOM and EOM framing, is
5421 lost.
- 5422 12. The retransmission timeout at Terminal A expires, indicating that Terminal A has not
5423 received REPORT. Terminal A must retransmit the entire message.
- 5424 13. Terminal A sends SOM in preparation for retransmitting the entire message.
- 5425 14. Terminal A retransmits frame 1.
- 5426 15. Terminal A retransmits frame 2. Terminal B receives SOM indicating an incoming message.
- 5427 16. Terminal A retransmits frame 3. Terminal B receives frame 1, recognizes that frame 1 has
5428 already been received error-free, and discards the newly received copy. Note that even
5429 though frame 1 is received with uncorrectable errors it is not added to the NAK list since it
5430 has previously been received error-free.
- 5431 17. Terminal A sends EOM. Terminal B receives frame 2.
- 5432 18. Terminal B receives frame 3, recognizes that frame 3 has already been received error-free,
5433 and discards the newly received copy.
- 5434 19. Terminal B receives EOM. All frames received by Terminal B at this point are contiguous,
5435 so Terminal B will respond with REPORT containing a null NAK list.
- 5436 20. Terminal B sends SOM to frame the REPORT. Terminal B also passes the information
5437 contained in all contiguously received frames (frames 2 and 3 in this example) to the
5438 Message Layer.
- 5439 21. Terminal B sends REPORT, indicating that frames up to and including frame 3 have been
5440 received correctly.
- 5441 22. Terminal B sends EOM. Terminal A should begin receiving the SOM at this point, but this
5442 example assumes that the entire REPORT message, including the SOM and EOM framing, is
5443 lost.
- 5444 23. The retransmission timeout at Terminal A expires, indicating that Terminal A has not
5445 received REPORT. Terminal A must retransmit the entire message.
- 5446 24. Terminal A sends SOM in preparation for retransmitting the entire message.
- 5447 25. Terminal A retransmits frame 1.
- 5448 26. Terminal A retransmits frame 2. Terminal B receives SOM indicating an incoming message.

- 5449 27. Terminal A retransmits frame 3. Terminal B receives frame 1, recognizes that frame 1 has
5450 already been received error-free, and discards the newly received copy. Note that even
5451 though frame 1 is received with uncorrectable errors it is not added to the NAK list since it
5452 has previously been received error-free.
- 5453 28. Terminal A sends EOM. Terminal B receives frame 2, recognizes that frame 2 has already
5454 been received error-free, and discards the newly received copy. Note that even though frame
5455 2 is received with uncorrectable errors it is not added to the NAK list since it has previously
5456 been received error-free.
- 5457 29. Terminal B receives frame 3, recognizes that frame 3 has already been received error-free,
5458 and discards the newly received copy. Note that even though frame 3 is received with
5459 uncorrectable errors it is not added to the NAK list since it has previously been received
5460 error-free.
- 5461 30. Terminal B receives EOM. All frames received by Terminal B at this point are contiguous,
5462 so Terminal B will respond with REPORT containing a null NAK list. Even though
5463 Terminal B has already sent a REPORT message acknowledging through block 3, it must
5464 send it again to prevent Terminal A from retransmitting again.
- 5465 31. Terminal B sends SOM to frame the REPORT.
- 5466 32. Terminal B sends REPORT, indicating that frames up to and including frame 3 have been
5467 received correctly.
- 5468 33. Terminal B sends EOM. Terminal A receives SOM, indicating an incoming message.
- 5469 34. Terminal A receives REPORT indicating that all frames through frame 3 have been received
5470 correctly. Terminal A may now discard the locally stored copies of frames 1, 2, and 3.
- 5471 35. Terminal A receives EOM.
- 5472 36. If necessary, the Transport Layer may inform the Message Layer that the CAPABILITIES
5473 message has been successfully transported.
- 5474

5475
5476
5477

A.5 Normal Transition from Signaling to Full Bandwidth Application



5478
5479
5480
5481
5482
5483
5484

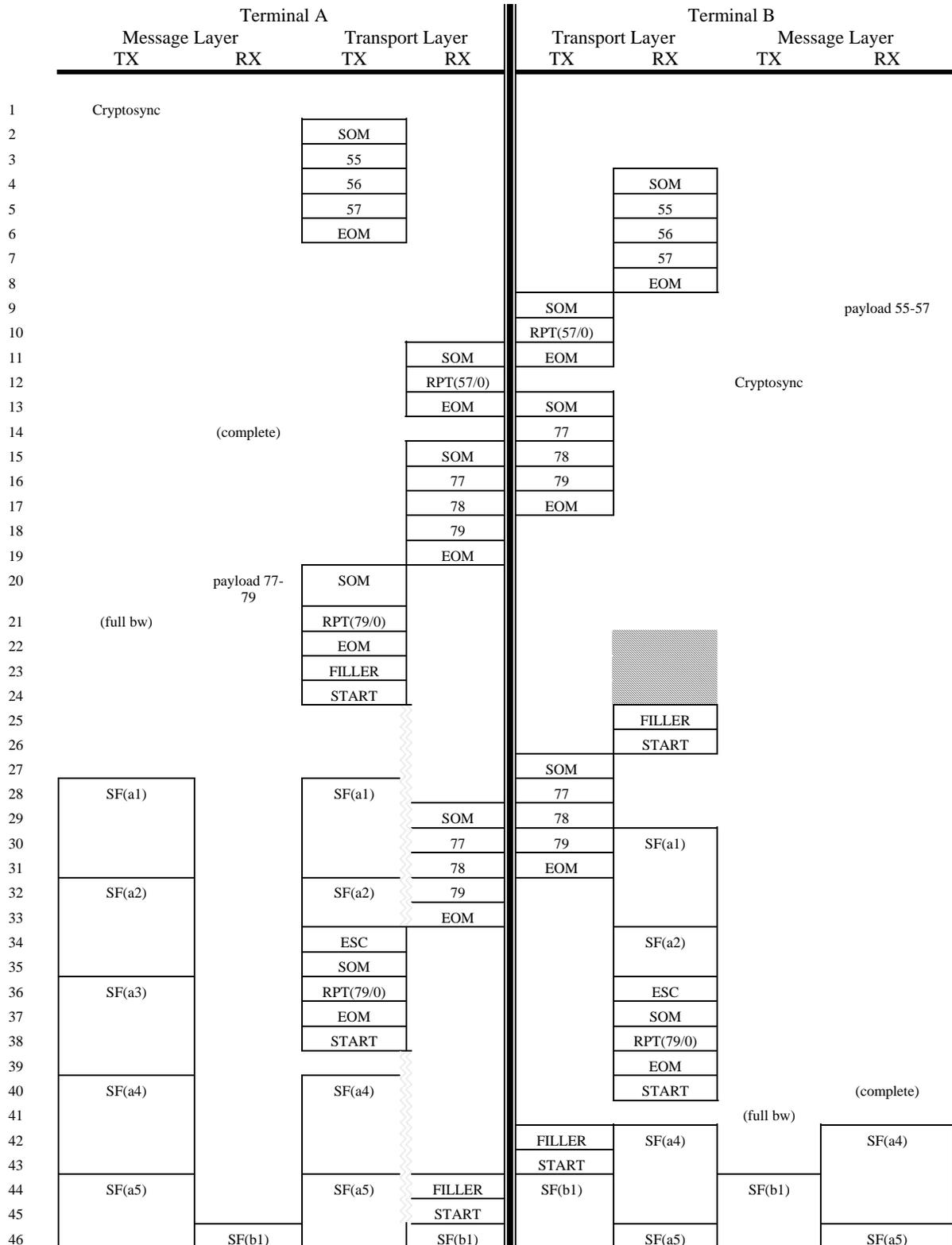
1. The Message Layer at Terminal A determines that a CRYPTOSYNC message needs to be sent. This example assumes that the CRYPTOSYNC message sent from the Message Layer at Terminal A is between 27 and 39 octets long, resulting in 3 frames at the Transport Layer.
2. The Transport Layer at Terminal A receives the CRYPTOSYNC message from the Message Layer, divides it into 3 frames, and begins sending SOM to frame the outgoing frames.

- 5485 3. This example assumes that the most recent transmitted frame from Terminal A was number
5486 54. Terminal A sends frame 55 & stores a local copy for possible retransmission.
- 5487 4. Terminal A sends frame 56 & stores a local copy for possible retransmission. Terminal B
5488 receives SOM, indicating an incoming message.
- 5489 5. Terminal A sends frame 57 & stores a local copy for possible retransmission. Terminal B
5490 receives frame 55.
- 5491 6. Terminal A sends EOM since all frames of the CRYPTOSYNC message have been sent.
5492 Terminal B receives frame 56.
- 5493 7. Terminal B receives frame 57.
- 5494 8. Terminal B receives EOM, indicating that the incoming message is complete.
- 5495 9. Terminal B knows of no outstanding frames and will therefore send a REPORT message
5496 indicating that frames up through 57 have been received correctly. A SOM is sent to frame
5497 the REPORT message. Terminal B concatenates the payload data from received frames 55-
5498 57 and passes it to the Message Layer, which determines that it forms a valid
5499 CRYPTOSYNC message.
- 5500 10. Terminal B sends REPORT indicating that all frames up to and including frame 57 have been
5501 received correctly.
- 5502 11. Terminal A receives SOM, indicating a new incoming message. Terminal B sends EOM,
5503 indicating the end of the REPORT.
- 5504 12. Terminal A receives REPORT indicating that frames up to and including frame 57 have been
5505 received correctly. Terminal A may now delete its local copy of transmitted frames 55-57
5506 since it knows that no further retransmissions of these frames will be necessary. This
5507 example assumes that at this time Terminal B determines that a CRYPTOSYNC message
5508 needs to be sent. The CRYPTOSYNC message is passed from the Message Layer to the
5509 Transport Layer at Terminal B.
- 5510 13. Terminal A receives EOM, indicating the end of the received REPORT. Terminal B sends
5511 SOM to frame the outgoing Transport Layer frames.
- 5512 14. The Transport Layer at Terminal A informs the Message Layer that the CRYPTOSYNC
5513 message has been successfully transported. Terminal B sends frame 77 and stores a local
5514 copy for possible retransmission.
- 5515 15. Terminal B sends frame 78 & stores a local copy for possible retransmission. Terminal A
5516 receives SOM, indicating the beginning of an incoming message.
- 5517 16. Terminal B sends frame 79 & stores a local copy for possible retransmission. Terminal A
5518 receives frame 77.
- 5519 17. Terminal A receives frame 78. Terminal B sends EOM since all frames of the
5520 CRYPTOSYNC message have been sent.
- 5521 18. Terminal A receives frame 79.
- 5522 19. Terminal A receives EOM, indicating that the incoming message is complete.
- 5523 20. Terminal A knows of no outstanding frames and will therefore send REPORT indicating that
5524 all frames up through 79 have been received correctly. A SOM is sent to frame the REPORT.
5525 Terminal A concatenates the payload data from received frames 77-79 and passes it to the
5526 Message Layer, which determines that it forms a valid CRYPTOSYNC message.
- 5527 21. Terminal A sends REPORT indicating that all frames up to and including frame 79 have
5528 been received correctly. Terminal A now knows that it is ready to transition to full bandwidth
5529 traffic. The Message Layer informs the Transport Layer that the change should occur as
5530 soon as any queued Transport Layer frames are sent.

- 5531 22. Terminal B receives SOM, indicating a new incoming message. Terminal A sends EOM,
5532 indicating the end of the REPORT.
- 5533 23. Terminal A sends FILLER in preparation for the transition from signaling to traffic.
5534 Terminal B receives REPORT indicating that all frames up through 79 have been received
5535 correctly.
- 5536 24. Terminal A sends START, indicating that subsequent transmissions will be full bandwidth
5537 traffic. Terminal A has not detected incoming START, so the Application Timer is started.
5538 Terminal B receives EOM, indicating the end of the received REPORT.
- 5539 25. Terminal B informs the Message Layer that the CRYPTOSYNC message has been
5540 successfully transported. Terminal B also receives FILLER.
- 5541 26. Terminal B now knows that it is ready to transition to full bandwidth traffic. The Message
5542 Layer informs the Transport Layer that the change should occur as soon as any queued
5543 Transport Layer frames are sent. Terminal B also receives START, indicating that
5544 subsequent incoming information will be full bandwidth. Terminal B begins searching for
5545 ESC and Sync Management patterns rather than SOM and START patterns
- 5546 27. Terminal B sends FILLER in preparation for the transition from signaling to traffic.
- 5547 28. This example assumes at this point that voice frames are available to be transmitted from
5548 Terminal A. The Message Layer at Terminal A begins transferring the first superframe (a1).
5549 Terminal B sends START, indicating that subsequent transmissions will be full bandwidth
5550 traffic. Terminal B does not start its Application Timer since incoming START has already
5551 been detected and Terminal B is no longer searching for incoming START.
- 5552 29. Terminal A continues sending superframe a1 and receives incoming FILLER.
- 5553 30. Terminal A continues sending superframe a1 and receives incoming START from Terminal
5554 B. Terminal A stops the Application Timer which has been running since START was
5555 transmitted. Terminal A begins searching for ESC and Sync Management patterns rather
5556 than SOM and START patterns. Terminal B begins receiving superframe a1 from Terminal
5557 A.
- 5558 31. Terminal A continues sending superframe a1. This example assumes at this point that voice
5559 frames are available to be transmitted from Terminal B. The Message Layer at Terminal B
5560 begins sending the first superframe (b1).
- 5561 32. Terminal B continues sending superframe b1 and receiving superframe a1.
- 5562 33. Terminal A begins receiving superframe b1. Terminal B continues sending superframe b1
5563 and receiving superframe a1.
- 5564 34. Terminal A continues receiving superframe b1. Terminal B continues sending superframe
5565 b1.
- 5566 35. Terminal A continues receiving superframe b1.
- 5567 36. Terminal A continues receiving superframe b1.
- 5568

5569
5570
5571

A.6 Transition from Signaling to Full Bandwidth Application with Final REPORT Lost



5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616

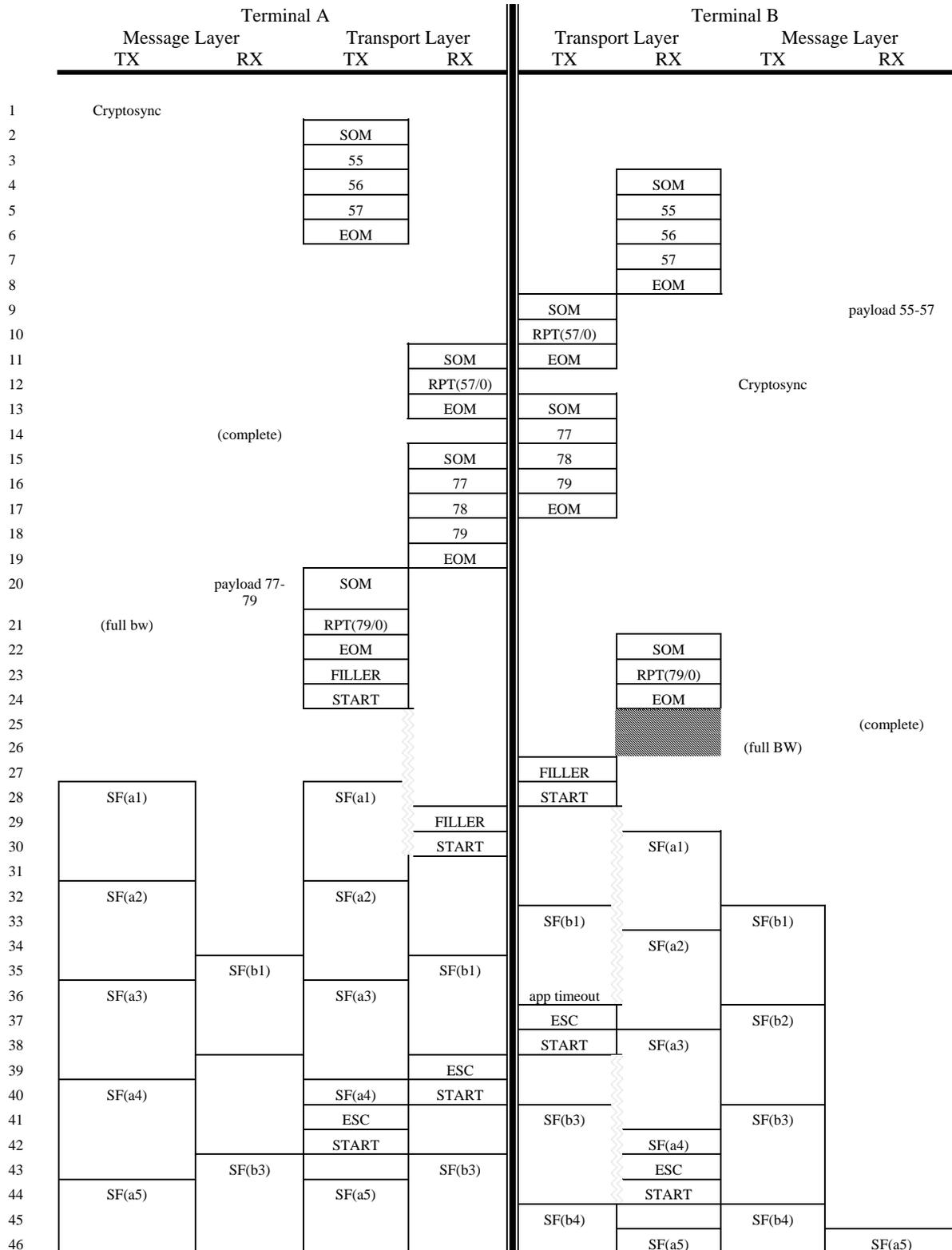
1. The Message Layer at Terminal A determines that a CRYPTOSYNC message needs to be sent. This example assumes that the CRYPTOSYNC message sent from the Message Layer at Terminal A is between 27 and 39 octets long, resulting in 3 frames at the Transport Layer.
2. The Transport Layer at Terminal A receives the CRYPTOSYNC message from the Message Layer, divides it into 3 frames, and begins sending SOM to frame the outgoing frames.
3. This example assumes that the most recent transmitted frame from Terminal A was number 54. Terminal A sends frame 55 & stores a local copy for possible retransmission.
4. Terminal A sends frame 56 & stores a local copy for possible retransmission. Terminal B receives SOM, indicating an incoming message.
5. Terminal A sends frame 57 & stores a local copy for possible retransmission. Terminal B receives frame 55.
6. Terminal A sends EOM since all frames of the CRYPTOSYNC message have been sent. Terminal B receives frame 56.
7. Terminal B receives frame 57.
8. Terminal B receives EOM, indicating that the incoming message is complete.
9. Terminal B knows of no outstanding frames and will therefore send a REPORT message indicating that frames up through 57 have been received correctly. A SOM is sent to frame the REPORT message. Terminal B concatenates the payload data from received frames 55-57 and passes it to the Message Layer, which determines that it forms a valid CRYPTOSYNC message.
10. Terminal B sends REPORT indicating that all frames up to and including frame 57 have been received correctly.
11. Terminal A receives SOM, indicating a new incoming message. Terminal B sends EOM, indicating the end of the REPORT.
12. Terminal A receives REPORT indicating that frames up to and including frame 57 have been received correctly. Terminal A may now delete its local copy of transmitted frames 55-57 since it knows that no further retransmissions of these frames will be necessary. This example assumes that at this time Terminal B determines that a CRYPTOSYNC message needs to be sent. The CRYPTOSYNC message is passed from the Message Layer to the Transport Layer at Terminal B.
13. Terminal A receives EOM, indicating the end of the received REPORT. Terminal B sends SOM to frame the outgoing Transport Layer frames.
14. The Transport Layer at Terminal A informs the Message Layer that the CRYPTOSYNC message has been successfully transported. Terminal B sends frame 77 and stores a local copy for possible retransmission.
15. Terminal B sends frame 78 & stores a local copy for possible retransmission. Terminal A receives SOM, indicating the beginning of an incoming message.
16. Terminal B sends frame 79 & stores a local copy for possible retransmission. Terminal A receives frame 77.
17. Terminal A receives frame 78. Terminal B sends EOM since all frames of the CRYPTOSYNC message have been sent.
18. Terminal A receives frame 79.
19. Terminal A receives EOM, indicating that the incoming message is complete.

- 5617 20. Terminal A knows of no outstanding frames and will therefore send REPORT indicating that
5618 all frames up through 79 have been received correctly. A SOM is sent to frame the REPORT.
5619 Terminal A concatenates the payload data from received frames 77-79 and passes it to the
5620 Message Layer, which determines that it forms a valid CRYPTOSYNC message.
- 5621 21. Terminal A sends REPORT indicating that all frames up to and including frame 79 have
5622 been received correctly. Terminal A now knows that it is ready to transition to full bandwidth
5623 traffic. The Message Layer informs the Transport Layer that the change should occur as
5624 soon as any queued Transport Layer frames are sent.
- 5625 22. This example assumes that Terminal B does not receive SOM from Terminal A which should
5626 have arrived at this point. Terminal A sends EOM indicating the end of the REPORT.
- 5627 23. Terminal A sends FILLER in preparation for the transition from signaling to traffic. This
5628 example assumes that Terminal B does not receive REPORT(79/0) from Terminal A which
5629 should have arrived at this point.
- 5630 24. Terminal A sends START, indicating that subsequent transmissions will be full bandwidth
5631 traffic. Terminal A has not detected incoming START, so the Application Timer is started.
5632 This example assumes that Terminal B does not receive EOM from Terminal A which should
5633 have arrived at this point.
- 5634 25. Terminal B receives FILLER.
- 5635 26. Terminal B receives START, indicating that subsequent incoming information will be full
5636 bandwidth. Terminal B begins searching for ESC and Sync Management patterns rather than
5637 SOM and START patterns.
- 5638 27. Terminal B times out waiting for Terminal A to acknowledge outstanding Transport Layer
5639 frames. Terminal B must therefore resend the previous frames to trigger another REPORT
5640 message from Terminal A. Terminal B sends SOM to frame the outgoing frames. Note that
5641 these outgoing frames do not need to be preceded by ESC since Terminal B has not sent a
5642 START message.
- 5643 28. This example assumes at this point that voice frames are available to be transmitted from
5644 Terminal A. The Message layer at Terminal A begins sending superframe a1.
- 5645 29. Terminal A continues sending superframe a1 and receives SOM, indicating an incoming
5646 Transport Layer message. Terminal B sends frame 78.
- 5647 30. Terminal A continues sending superframe a1 and receives frame 77. Terminal B sends frame
5648 79 and begins receiving superframe a1.
- 5649 31. Terminal A continues sending superframe a1 and receives frame 78. Terminal B sends EOM
5650 and continues receiving superframe a1.
- 5651 32. Terminal A begins sending superframe a2 and receives frame 79. Terminal B continues
5652 receiving superframe a1.
- 5653 33. Terminal A continues sending superframe a2 and receives EOM. Terminal A now knows
5654 that it must return a REPORT message and must therefore transition back to the signaling
5655 mode. The Application Timer which is running at Terminal A is stopped.
- 5656 34. Since Terminal A has already sent a START message, it must precede the outgoing
5657 Transport Layer frames with ESC. Terminal B begins receiving superframe a2.
- 5658 35. Terminal A sends SOM to frame the outgoing REPORT message. Terminal B continues
5659 receiving superframe a2.

- 5660 36. Terminal A sends REPORT indicating that frames up through 79 have been received
5661 correctly. Terminal B receives ESC indicating that subsequent information will be framed at
5662 the Transport Layer. Terminal B begins searching for SOM and START patterns rather than
5663 ESC and Sync Management patterns.
- 5664 37. Terminal A sends EOM to frame the outgoing REPORT message. Terminal B receives SOM
5665 indicating an incoming Transport Layer message.
- 5666 38. Terminal A recognizes that it has no more Transport Layer information to send and
5667 transitions back to traffic mode by sending START to indicate that subsequent information
5668 will be full bandwidth traffic. The Application Timer at Terminal A is reinitialized and
5669 restarted since incoming START has not been detected. Terminal B receives REPORT
5670 indicating that frames up through 79 have been received properly.
- 5671 39. Terminal B receives EOM, indicating the end of the received REPORT.
- 5672 40. This example assumes at this point that voice frames are available to be transmitted from
5673 Terminal A. The Message Layer at Terminal A begins sending superframe a4. Terminal B
5674 informs the Message Layer that the CRYPTOSYNC message has been successfully
5675 transported. Terminal B also receives START, indicating that subsequent incoming
5676 information will be full bandwidth traffic. Terminal B begins searching for ESC and Sync
5677 Management patterns rather than SOM and START patterns.
- 5678 41. Terminal A continues to send superframe a4. Terminal B now knows that it is ready to
5679 transition to full bandwidth traffic. The Message Layer informs the Transport Layer that the
5680 change should occur as soon as any queued Transport Layer frames are sent.
- 5681 42. Terminal A continues to send superframe a4. Terminal B sends FILLER in preparation for
5682 the transition from signaling to traffic. Terminal B also begins receiving superframe a4.
- 5683 43. Terminal A continues to send superframe a4. Terminal B sends START, indicating that
5684 subsequent transmissions will be full bandwidth traffic. Terminal B does not start its
5685 Application Timer since incoming START has already been detected and Terminal B is no
5686 longer searching for incoming START. Terminal B continues receiving superframe a4.
- 5687 44. Terminal A begins sending superframe a5 and receives incoming FILLER. This example
5688 assumes at this point that voice frames are available to be transmitted from Terminal B. The
5689 Message Layer at Terminal B begins sending superframe b1. Terminal B continues
5690 receiving superframe a4.
- 5691 45. Terminal A continues sending superframe a5 and receives incoming START from Terminal
5692 B. Terminal A stops the Application Timer which has been running since START was
5693 transmitted. Terminal A begins searching for ESC and Sync Management patterns rather
5694 than SOM and START patterns. Terminal B continues receiving superframe a4 from
5695 Terminal A.
- 5696 46. Terminal A continues sending superframe a5 and begins receiving superframe b1. Terminal
5697 B continues sending superframe b1 and begins receiving superframe a5.
- 5698

5699
5700
5701

A.7 Transition from Signaling to Full Bandwidth Application with START Lost



5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746

1. The Message Layer at Terminal A determines that a CRYPTOSYNC message needs to be sent. This example assumes that the CRYPTOSYNC message sent from the Message Layer at Terminal A is between 27 and 39 octets long, resulting in 3 frames at the Transport Layer.
2. The Transport Layer at Terminal A receives the CRYPTOSYNC message from the Message Layer, divides it into 3 frames, and begins sending SOM to frame the outgoing frames.
3. This example assumes that the most recent transmitted frame from Terminal A was number 54. Terminal A sends frame 55 & stores a local copy for possible retransmission.
4. Terminal A sends frame 56 & stores a local copy for possible retransmission. Terminal B receives SOM, indicating an incoming message.
5. Terminal A sends frame 57 & stores a local copy for possible retransmission. Terminal B receives frame 55.
6. Terminal A sends EOM since all frames of the CRYPTOSYNC message have been sent. Terminal B receives frame 56.
7. Terminal B receives frame 57.
8. Terminal B receives EOM, indicating that the incoming message is complete.
9. Terminal B knows of no outstanding frames and will therefore send a REPORT message indicating that frames up through 57 have been received correctly. A SOM is sent to frame the REPORT message. Terminal B concatenates the payload data from received frames 55-57 and passes it to the Message Layer, which determines that it forms a valid CRYPTOSYNC message.
10. Terminal B sends REPORT indicating that all frames up to and including frame 57 have been received correctly.
11. Terminal A receives SOM, indicating a new incoming message. Terminal B sends EOM, indicating the end of the REPORT.
12. Terminal A receives REPORT indicating that frames up to and including frame 57 have been received correctly. Terminal A may now delete its local copy of transmitted frames 55-57 since it knows that no further retransmissions of these frames will be necessary. This example assumes that at this time Terminal B determines that a CRYPTOSYNC message needs to be sent. The CRYPTOSYNC message is passed from the Message Layer to the Transport Layer at Terminal B.
13. Terminal A receives EOM, indicating the end of the received REPORT. Terminal B sends SOM to frame the outgoing Transport Layer frames.
14. The Transport Layer at Terminal A informs the Message Layer that the CRYPTOSYNC message has been successfully transported. Terminal B sends frame 77 and stores a local copy for possible retransmission.
15. Terminal B sends frame 78 & stores a local copy for possible retransmission. Terminal A receives SOM, indicating the beginning of an incoming message.
16. Terminal B sends frame 79 & stores a local copy for possible retransmission. Terminal A receives frame 77.
17. Terminal A receives frame 78. Terminal B sends EOM since all frames of the CRYPTOSYNC message have been sent.
18. Terminal A receives frame 79.
19. Terminal A receives EOM, indicating that the incoming message is complete.

- 5747 20. Terminal A knows of no outstanding frames and will therefore send REPORT indicating that
5748 all frames up through 79 have been received correctly. A SOM is sent to frame the REPORT.
5749 Terminal A concatenates the payload data from received frames 77-79 and passes it to the
5750 Message Layer, which determines that it forms a valid CRYPTOSYNC message.
- 5751 21. Terminal A sends REPORT indicating that all frames up to and including frame 79 have
5752 been received correctly. Terminal A now knows that it is ready to transition to full bandwidth
5753 traffic. The Message Layer informs the Transport Layer that the change should occur as
5754 soon as any queued Transport Layer frames are sent.
- 5755 22. Terminal B receives SOM, indicating a new incoming message. Terminal A sends EOM
5756 indicating the end of the REPORT.
- 5757 23. Terminal A sends FILLER in preparation for the transition from signaling to traffic.
5758 Terminal B receives REPORT indicating that all frames up through 79 have been received
5759 correctly.
- 5760 24. Terminal A sends START, indicating that subsequent transmissions will be full bandwidth
5761 traffic. Terminal A has not detected incoming START, so the Application Timer is started.
5762 Terminal B receives EOM, indicating the end of the received REPORT.
- 5763 25. Terminal B informs the message layer that the CRYPTOSYNC message has been
5764 successfully transported. This example assumes that the FILLER transmitted from Terminal
5765 A is not received at Terminal B.
- 5766 26. Terminal B now knows that it is ready to transition to full bandwidth traffic. The Message
5767 Layer informs the Transport Layer that the change should occur as soon as any queued
5768 Transport Layer frames are sent. This example assumes that the START transmitted from
5769 Terminal A is not received at Terminal B.
- 5770 27. Terminal B sends FILLER in preparation for the transition from signaling to traffic.
- 5771 28. This example assumes at this point that voice frames are available to be transmitted from
5772 Terminal A. The Message Layer at Terminal A begins transferring superframe a1. Terminal
5773 B sends START, indicating that subsequent transmissions will be full bandwidth traffic.
5774 Terminal B has not detected incoming START, so the Application Timer is started.
- 5775 29. Terminal A continues sending superframe a1 and receives incoming FILLER.
- 5776 30. Terminal A continues sending superframe a1 and receives incoming START. Terminal A
5777 stops the Application Timer which has been running since START was transmitted.
5778 Terminal A begins searching for ESC and Sync Management patterns rather than SOM and
5779 START patterns in the incoming data stream. Terminal B begins receiving superframe a1.
5780 Note that superframe a1 is not detected at Terminal B since Terminal B has not seen START
5781 and is therefore looking for SOM and START patterns rather than Sync Management
5782 patterns.
- 5783 31. Terminal A continues sending superframe a1. Superframe a1 is still not detected at Terminal
5784 B.
- 5785 32. Terminal A begins sending superframe a2. Superframe a1 is still not detected at Terminal B.
- 5786 33. Terminal A continues sending superframe a2. This example assumes at this point that voice
5787 frames are available to be transmitted from Terminal B. The Message Layer at Terminal B
5788 begins transferring superframe b1. Superframe a1 is still not detected at Terminal B.
- 5789 34. Terminal A continues sending superframe a2. Terminal B continues sending superframe b1.
5790 Terminal B begins receiving superframe a2. Note that superframe a2 is not detected at
5791 Terminal B since Terminal B has not seen START and is therefore looking for SOM and
5792 START patterns rather than Sync Management patterns.

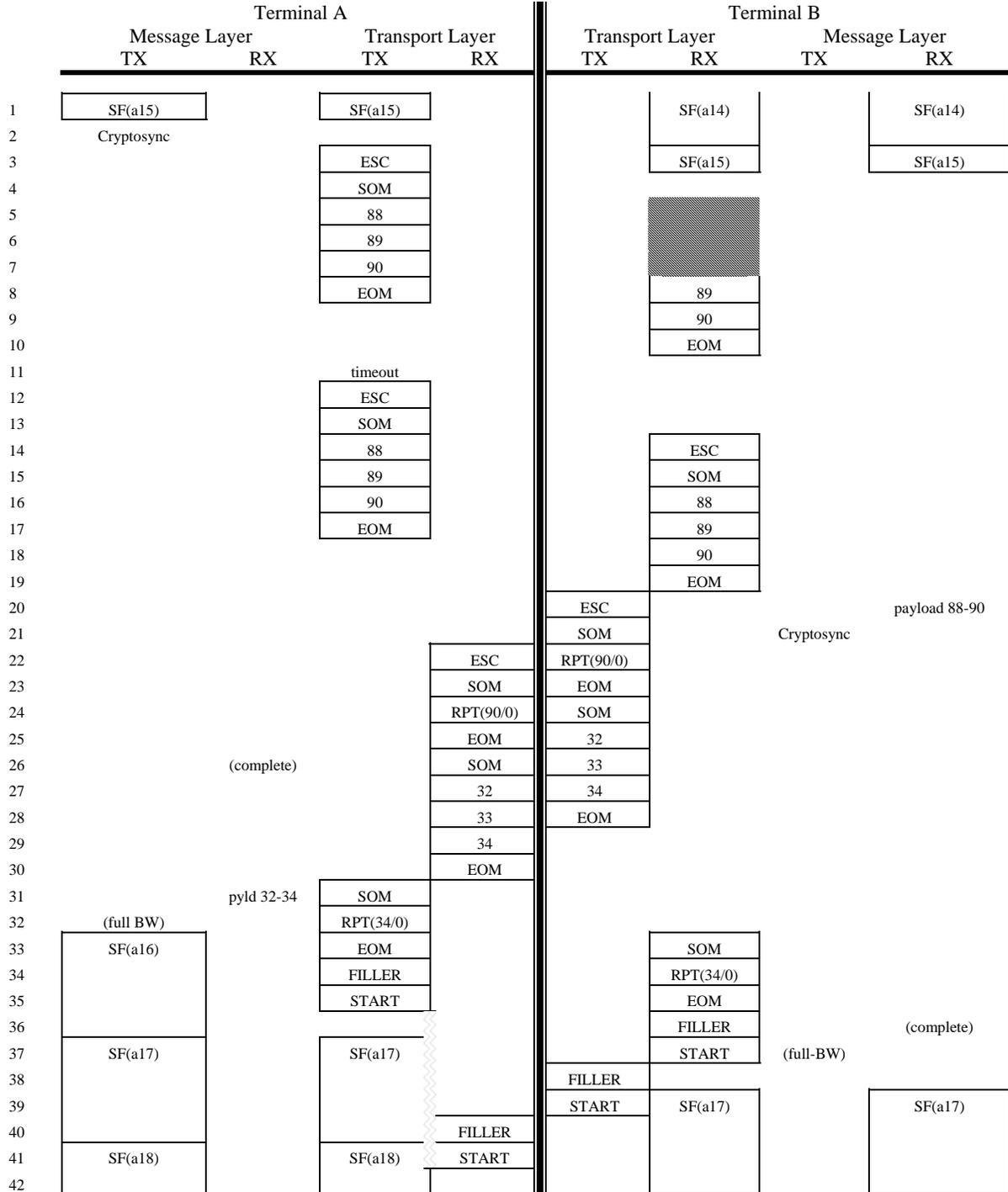
- 5793 35. Terminal A continues sending superframe a2 and begins receiving superframe b1. Terminal
5794 B continues sending superframe b1. Superframe a2 is still not detected at Terminal B.
- 5795 36. Terminal A begins sending superframe a3 and continues receiving superframe b1. Terminal
5796 B continues sending superframe b1. Superframe a2 is still not detected at Terminal B. The
5797 Application Timer at Terminal B expires, indicating that Terminal B has not detected
5798 incoming START.
- 5799 37. Terminal A continues sending superframe a3 and receiving superframe b1. Terminal B
5800 sends ESC as a result of the Application Timer expiring. Superframe a2 is still not detected
5801 at Terminal B.
- 5802 38. Terminal A continues sending superframe a3 and receiving superframe b1. Terminal B
5803 sends START. Terminal B has not detected incoming START, so the Application Timer is
5804 reinitialized and restarted. Terminal B begins receiving superframe a3. Note that
5805 superframe a3 is not detected at Terminal B since Terminal B has not seen START and is
5806 therefore looking for SOM and START patterns rather than Sync Management patterns.
- 5807 39. Terminal A continues sending superframe a3 and receives ESC. Terminal A therefore knows
5808 that subsequent incoming data will be Transport Layer framed data and begins looking for
5809 SOM and START patterns rather than ESC and Sync Management patterns. Superframe a3
5810 is still not detected at Terminal B.
- 5811 40. Terminal A begins sending superframe a4 and receives START, indicating that subsequent
5812 incoming information will be full bandwidth traffic. Terminal A recognizes that incoming
5813 START was detected while the Application Timer is not running, and is therefore required to
5814 send ESC and START. Superframe a3 is still not detected at Terminal B.
- 5815 41. Terminal A stops sending superframe a4 and sends ESC. Terminal B begins sending
5816 superframe b3. Superframe a3 is still not detected at Terminal B.
- 5817 42. Terminal A sends START. Terminal A does not start its Application Timer since incoming
5818 START has already been detected and Terminal A is no longer searching for incoming
5819 START. Terminal A begins searching for ESC and Sync Management patterns rather than
5820 SOM and START patterns in the incoming data stream. Terminal B begins receiving
5821 superframe a4. Note that superframe a4 is not detected at Terminal B since Terminal B has
5822 not seen START and is therefore looking for SOM and START patterns rather than Sync
5823 Management patterns.
- 5824 43. Terminal A begins receiving superframe b3. Terminal B continues sending superframe b3
5825 and receives ESC. Terminal B therefore knows that subsequent incoming data will be
5826 Transport Layer framed data and continues looking for SOM and START patterns rather than
5827 ESC and Sync Management patterns.
- 5828 44. Terminal A begins sending superframe a5 and continues receiving superframe b3. Terminal
5829 B continues sending superframe b3 and receives START, indicating that subsequent
5830 incoming information will be full bandwidth traffic. Terminal B stops the Application Timer
5831 which has been running since START was transmitted. Terminal B begins searching for
5832 ESC and Sync Management patterns rather than SOM and START patterns in the incoming
5833 data stream.
- 5834 45. Terminal A continues sending superframe a5 and receiving superframe b3. Terminal B
5835 begins sending superframe b4.
- 5836 46. Terminal A continues sending superframe a5 and receiving superframe b3. Terminal B
5837 continues receiving superframe b4 and begins receiving superframe a5.
- 5838

- 5851 5. This example assumes that the Cryptosync message is between 27 and 39 bytes long,
5852 resulting in three frames at the Transport Layer, and that the most recently transmitted
5853 Transport Layer frame was #87. Frame 88 is therefore sent. Terminal B receives ESCAPE
5854 indicating the beginning of an incoming Transport Layer message.
- 5855 6. Terminal A sends frame 89. Terminal B receives the SOM.
- 5856 7. Terminal A sends frame 90. Terminal B receives frame 88.
- 5857 8. Terminal A sends EOM to frame the outgoing Cryptosync message. Terminal B receives
5858 frame 89.
- 5859 9. Terminal B receives frame 90.
- 5860 10. Terminal B receives EOM indicating the end of the incoming Transport Layer message.
- 5861 11. Terminal B knows of no outstanding frames and will therefore acknowledge frame 90 using a
5862 REPORT message. Terminal B sends ESCAPE to alert the remote end that Transport Layer
5863 signaling is occurring. Terminal B passes the payload information from frames 88 to 90 to
5864 the Message Layer, which determines that it is a valid Cryptosync message.
- 5865 12. Terminal B sends SOM to frame the REPORT message. This example assumes that
5866 Terminal B is ready to send Cryptosync at this point, so it is transferred to the Transport
5867 Layer.
- 5868 13. Terminal B sends REPORT indicating that frames through 90 have been received correctly.
5869 Terminal A receives ESCAPE indicating the beginning of an incoming Transport Layer
5870 message.
- 5871 14. Terminal A receives SOM indicating an incoming message. Terminal B sends EOM,
5872 framing the REPORT message.
- 5873 15. Terminal A receives REPORT. Terminal B sends SOM to frame the outgoing Cryptosync
5874 message.
- 5875 16. Terminal A receives EOM. This example assumes that the Cryptosync message is between
5876 27 and 39 bytes long, resulting in three frames at the Transport Layer, and that the most
5877 recently transmitted Transport Layer frame was #31. Frame 32 is therefore sent.
- 5878 17. The Transport Layer at Terminal A informs the Message Layer that the Cryptosync message
5879 has been successfully transported. Terminal A receives SOM indicating an incoming
5880 message. Terminal B sends frame 33.
- 5881 18. Terminal A receives frame 32. Terminal B sends frame 34.
- 5882 19. Terminal A receives frame 33. Terminal B sends EOM to frame the outgoing Cryptosync
5883 message.
- 5884 20. Terminal A receives frame 34.
- 5885 21. Terminal A receives EOM.
- 5886 22. Terminal A knows of no outstanding frames and will therefore acknowledge frame 34 using
5887 a REPORT message. SOM is sent to frame the REPORT. Terminal A passes the payload
5888 information from frames 32 to 34 to the Message Layer, which determines that it is a valid
5889 Cryptosync message.
- 5890 23. The Message Layer at Terminal A now knows that Cryptosync has been sent and received,
5891 so it informs the Transport Layer to transition back to traffic mode. Terminal A sends
5892 REPORT indicating that frames through 34 have been received correctly.
- 5893 24. The Message Layer at Terminal A begins passing superframes to the Transport Layer, which
5894 is still busy with Transport Layer signaling. Terminal A sends EOM to frame the REPORT.
5895 Terminal B receives SOM.

- 5896 25. Terminal A is now ready to transition to full bandwidth mode and sends FILLER since
5897 Cryptosync was the last message transferred. Terminal B receives REPORT.
- 5898 26. Terminal A sends START to complete the transition to full bandwidth mode. Terminal A
5899 starts the Application Timer since START has not been received. Terminal B receives EOM.
- 5900 27. Terminal B receives FILLER. The Transport Layer at Terminal B informs the Message
5901 Layer that the Cryptosync message has been successfully transported
- 5902 28. Terminal A begins sending superframe a17. Terminal B receives START. The Message
5903 Layer at Terminal B recognizes that Cryptosync has been received and sent and therefore
5904 instructs the Transport Layer to transition back to full bandwidth mode.
- 5905 29. Terminal A continues sending superframe a17. Terminal B sends FILLER in preparation for
5906 transitioning to full bandwidth mode.
- 5907 30. Terminal A continues sending superframe a17. Terminal B sends START to complete the
5908 transition to full bandwidth mode. The Application Timer at Terminal B is not started since
5909 incoming START has already been detected. Terminal B begins receiving superframe a17.
- 5910 31. Terminal A continues sending superframe a17 and receives incoming FILLER. Terminal B
5911 continues receiving superframe a17.
- 5912 32. Terminal A begins sending superframe a18 and receives incoming START. The Application
5913 Timer at Terminal A is stopped. Terminal B continues receiving superframe a17.
- 5914 33. Terminal A continues sending superframe a18. Terminal B continues receiving superframe
5915 a17.
- 5916

5917
5918
5919
5920

A.9 Two Way Resync from Full Bandwidth Application with Corrupted ESC Sequence, Terminal A is Leader



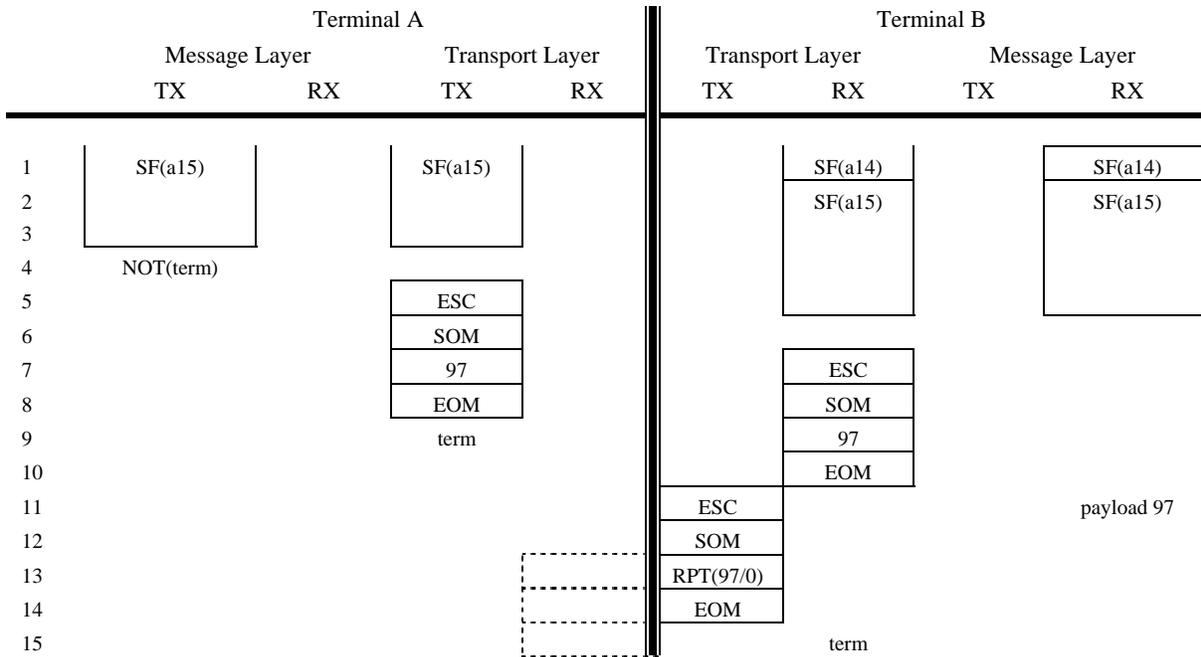
5921
5922

- 5923 1. This example begins by assuming that both Terminal A and Terminal B are in traffic mode.
5924 Terminal A is sending superframe #15. Terminal B is receiving superframe #14.
- 5925 2. Terminal A begins the Two Way Resync procedure by transferring a Cryptosync message to
5926 its Transport Layer for transmission to the remote terminal.
- 5927 3. Since Terminal A is in traffic mode, an ESCAPE is sent to alert the remote end that
5928 Transport Layer signaling is occurring. Terminal B receives superframe a15.
- 5929 4. Terminal A sends SOM to frame the outgoing Cryptosync message.
- 5930 5. This example assumes that the Cryptosync message is between 27 and 39 bytes long,
5931 resulting in three frames at the Transport Layer, and that the most recently transmitted
5932 Transport Layer frame was #87. Frame 88 is therefore sent. Terminal B should receive ESC
5933 at this point, but this example assumes that the ESC is lost.
- 5934 6. Terminal A sends frame 89. Terminal B should receive SOM but this example assumes that
5935 it is lost.
- 5936 7. Terminal A sends frame 90. Terminal B should receive frame 88 but this example assumes
5937 that it is lost.
- 5938 8. Terminal A sends EOM to frame the outgoing Cryptosync. Terminal B receives frame 89
5939 but it is not detected since Terminal B missed ESC and is therefore searching for ESC and
5940 Sync Management frames.
- 5941 9. Terminal B receives frame 90 but it is not detected since Terminal B missed ESC and is
5942 therefore searching for ESC and Sync Management frames.
- 5943 10. Terminal B receives EOM but it is not detected since Terminal B missed ESC and is
5944 therefore searching for ESC and Sync Management frames.
- 5945 11. The Retransmission Timer at Terminal A eventually times out and forces Terminal A to
5946 resend the previous frames.
- 5947 12. Since the previous Transport Layer transmission from Terminal A began with ESCAPE, an
5948 ESCAPE is again sent.
- 5949 13. Terminal A sends SOM to frame the outgoing Cryptosync message retransmission.
- 5950 14. Terminal B receives ESCAPE indicating the beginning of an incoming Transport Layer
5951 message. Terminal A sends frame 88
- 5952 15. Terminal A sends frame 89. Terminal B receives the SOM.
- 5953 16. Terminal A sends frame 90. Terminal B receives frame 88.
- 5954 17. Terminal A sends EOM to frame the outgoing Cryptosync message. Terminal B receives
5955 frame 89.
- 5956 18. Terminal B receives frame 90.
- 5957 19. Terminal B receives EOM indicating the end of the incoming Transport Layer message.
- 5958 20. Terminal B knows of no outstanding frames and will therefore acknowledge frame 90 using a
5959 REPORT message. Terminal B sends ESCAPE to alert the remote end that Transport Layer
5960 signaling is occurring. Terminal B passes the payload information from frames 88 to 90 to
5961 the Message Layer, which determines that it is a valid Cryptosync message.
- 5962 21. Terminal B sends SOM to frame the REPORT message. This example assumes that
5963 Terminal B is ready to send Cryptosync at this point, so it is transferred to the Transport
5964 Layer.
- 5965 22. Terminal B sends REPORT indicating that frames through 90 have been received correctly.
5966 Terminal A receives ESCAPE indicating the beginning of an incoming Transport Layer
5967 message.

- 5968 23. Terminal A receives SOM indicating an incoming message. Terminal B sends EOM,
5969 framing the REPORT message.
- 5970 24. Terminal A receives REPORT. Terminal B sends SOM to frame the outgoing Cryptosync
5971 message.
- 5972 25. Terminal A receives EOM. This example assumes that the Cryptosync message is between
5973 27 and 39 bytes long, resulting in three frames at the Transport Layer, and that the most
5974 recently transmitted Transport Layer frame was #31. Frame 32 is therefore sent.
- 5975 26. The Transport Layer at Terminal A informs the Message Layer that the Cryptosync message
5976 has been successfully transported. Terminal A receives SOM indicating an incoming
5977 message. Terminal B sends frame 33.
- 5978 27. Terminal A receives frame 32. Terminal B sends frame 34.
- 5979 28. Terminal A receives frame 33. Terminal B sends EOM to frame the outgoing Cryptosync
5980 message.
- 5981 29. Terminal A receives frame 34.
- 5982 30. Terminal A receives EOM.
- 5983 31. Terminal A knows of no outstanding frames and will therefore acknowledge frame 34 using
5984 a REPORT message. SOM is sent to frame the REPORT. Terminal A passes the payload
5985 information from frames 32 to 34 to the Message Layer, which determines that it is a valid
5986 Cryptosync message.
- 5987 32. The Message Layer at Terminal A now knows that Cryptosync has been sent and received,
5988 so it informs the Transport Layer to transition back to traffic mode. Terminal A sends
5989 REPORT indicating that frames through 34 have been received correctly.
- 5990 33. The Message Layer at Terminal A begins passing superframes to the Transport Layer, which
5991 is still busy with Transport Layer signaling. Terminal A sends EOM to frame the REPORT.
5992 Terminal B receives SOM.
- 5993 34. Terminal A is now ready to transition to full bandwidth mode and sends FILLER since
5994 Cryptosync was the last message transferred. Terminal B receives REPORT.
- 5995 35. Terminal A sends START to complete the transition to full bandwidth mode. Terminal A
5996 starts the Application Timer since START has not been received. Terminal B receives EOM.
- 5997 36. Terminal B receives FILLER. The Transport Layer at Terminal B informs the Message
5998 Layer that the Cryptosync message has been successfully transported
- 5999 37. Terminal A begins sending superframe a17. Terminal B receives START. The Message
6000 Layer at Terminal B recognizes that Cryptosync has been received and sent and therefore
6001 instructs the Transport Layer to transition back to full bandwidth mode.
- 6002 38. Terminal A continues sending superframe a17. Terminal B sends FILLER in preparation for
6003 transitioning to full bandwidth mode.
- 6004 39. Terminal A continues sending superframe a17. Terminal B sends START to complete the
6005 transition to full bandwidth mode. The Application Timer at Terminal B is not started since
6006 incoming START has already been detected. Terminal B begins receiving superframe a17.
- 6007 40. Terminal A continues sending superframe a17 and receives incoming FILLER. Terminal B
6008 continues receiving superframe a17.
- 6009 41. Terminal A begins sending superframe a18 and receives incoming START. The Application
6010 Timer at Terminal A is stopped. Terminal B continues receiving superframe a17.
- 6011 42. Terminal A continues sending superframe a18. Terminal B continues receiving superframe
6012 a17.
- 6013

6014
6015
6016

A.10 Normal Termination from Full Bandwidth Application, Terminal A is Leader



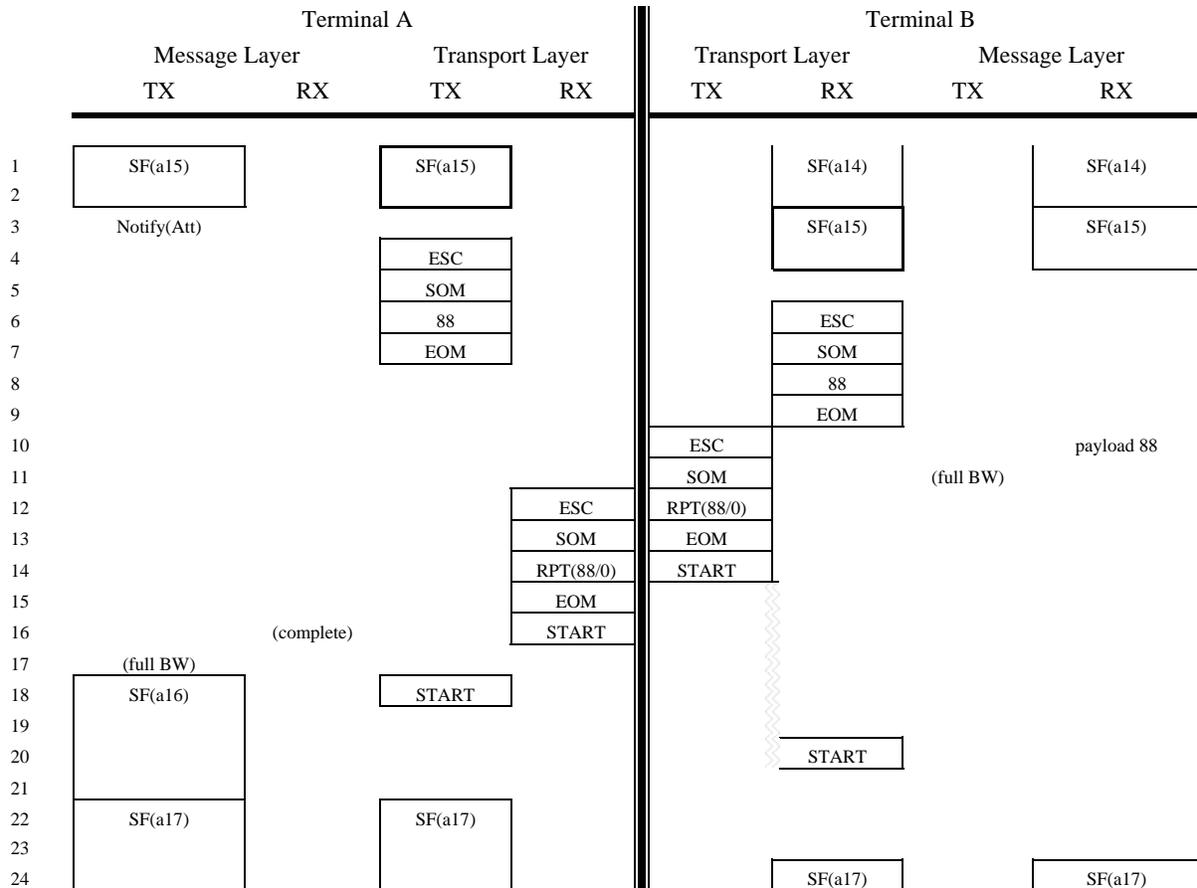
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040

1. This example begins by assuming that both Terminal A and Terminal B are in full bandwidth traffic mode. Terminal A is sending superframe a15. Terminal B is receiving superframe a14.
2. Terminal A continues sending superframe a15. Terminal B begins receiving superframe a15.
3. Terminal A continues sending superframe a15. Terminal B continues receiving superframe a15.
4. This example assumes that at this point the Message Layer at Terminal A determines that the connection will be terminated. A NOTIFY(terminate) message is transferred to the Transport Layer at Terminal A. Terminal B continues receiving superframe a15.
5. Since Terminal A is in full bandwidth traffic mode, an ESCAPE is sent to alert the remote end that Transport Layer signaling is occurring. Terminal B continues receiving superframe a15.
6. Terminal A sends SOM to frame the outgoing NOTIFY message.
7. This example assumes that the NOTIFY message is between 1 and 13 bytes long, resulting in one frame at the Transport Layer, and that the most recently transmitted Transport Layer frame was #96. Frame 97 is therefore sent to transfer the NOTIFY message. Terminal B receives ESCAPE indicating the beginning of an incoming Transport Layer message.
8. Terminal A sends EOM to frame the outgoing Notify message. Terminal B receives SOM.
9. Terminal A initiates the native signaling to terminate the underlying data connection. Note that Terminal A is not required to wait for an acknowledgement that Terminal B has received frame 97. Terminal B receives frame 97.
10. Terminal B receives EOM indicating the end of the incoming Transport Layer message.

- 6041 11. Terminal B knows of no outstanding frames and will therefore acknowledge frame 97 using a
6042 REPORT message. An ESCAPE is sent to alert the remote end that Transport Layer
6043 signaling is occurring. Terminal B passes the payload information from frame 97 to the
6044 Message Layer, which determines that it is a valid NOTIFY message.
6045 12. SOM is sent to frame the REPORT.
6046 13. Terminal B sends REPORT indicating that frames through 97 have been received correctly.
6047 In this example it is assumed that the underlying channel has been terminated at Terminal A
6048 before the ESC arrives.
6049 14. In this example it is assumed that the underlying channel has been terminated at Terminal A
6050 before the SOM arrives. Terminal B sends EOM to frame the REPORT.
6051 15. Terminal B initiates the native signaling to terminate the underlying data connection. No
6052 additional SCIP signaling is possible.
6053

6054
6055
6056

A.11 Terminal A Sends Notify(Attention) from Full Bandwidth Application



6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074

1. This example begins by assuming that both Terminal A and Terminal B are in full bandwidth traffic mode. Terminal A is sending superframe a15. Terminal B is receiving superframe a14.
2. Terminal A continues sending superframe a15. Terminal B continues receiving superframe a14.
3. This example assumes that at this point the Message Layer at Terminal A determines that an Notify(Attention) message is required. A Notify(Attention) message is transferred to the Transport Layer at Terminal A. Terminal B begins receiving superframe a15.
4. Since Terminal A is in full bandwidth traffic mode, ESCAPE is sent to alert the remote end that Transport Layer signaling is occurring. Terminal B continues receiving superframe a15.
5. Terminal A sends SOM to frame the outgoing Notify message.
6. This example assumes that the Notify message is between 1 and 13 bytes long, resulting in one frame at the Transport Layer, and that the most recently transmitted Transport Layer frame was #87. Frame 88 is therefore sent to transfer the Notify message. Terminal B receives ESCAPE indicating the beginning of an incoming Transport Layer message.
7. Terminal A sends EOM to frame the outgoing Notify message. Terminal B receives SOM.

- 6075 8. Terminal B receives frame 88.
- 6076 9. Terminal B receives EOM indicating the end of the incoming Transport Layer message.
- 6077 10. Terminal B knows of no outstanding frames and will therefore acknowledge frame 88 using a
6078 REPORT message. ESCAPE is sent to alert the remote end that Transport Layer signaling is
6079 occurring. Terminal B passes the payload information from frame 88 to the Message Layer,
6080 which determines that it is a valid Notify(Attention) message.
- 6081 11. Terminal B sends SOM to frame the outgoing REPORT. The Message Layer at Terminal B
6082 indicates to the Transport Layer that the full bandwidth traffic mode is to resume.
- 6083 12. Terminal A receives ESCAPE indicating the beginning of an incoming Transport Layer
6084 message. Terminal B sends REPORT indicating that frames through 88 have been received
6085 correctly.
- 6086 13. Terminal A receives SOM, indicating an incoming message. Terminal B sends EOM,
6087 framing the outgoing REPORT.
- 6088 14. Terminal A receives REPORT. Terminal B sends START to resume the full bandwidth
6089 traffic application. Note that FILLER is not required since Cryptosync was not transferred.
6090 Terminal B has not received incoming START, so the Application Timer is started.
- 6091 15. Terminal A receives EOM.
- 6092 16. The Transport Layer at Terminal A informs the Message Layer that the Notify(Attention)
6093 message has been successfully transported.
- 6094 17. The Message Layer at Terminal A indicates to the Transport Layer that the full bandwidth
6095 traffic mode is to resume.
- 6096 18. Terminal A sends START to reinitiate traffic. The Application Timer is not started since
6097 incoming START has already been detected.
- 6098 19. The Transport Layer at Terminal A waits for the beginning of a superframe to begin full
6099 bandwidth transmission.
- 6100 20. Terminal B receives START indicating incoming traffic. The Application Timer is now
6101 stopped.
- 6102 21. The Transport Layer at Terminal A waits for the beginning of a superframe to begin full
6103 bandwidth transmission.
- 6104 22. Terminal A begins sending superframe a17.
- 6105 23. Terminal A continues sending superframe a17.
- 6106 24. Terminal A continues sending superframe a17. Terminal B begins receiving superframe a17.
6107

6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131

THIS PAGE INTENTIONALLY LEFT BLANK.

6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150

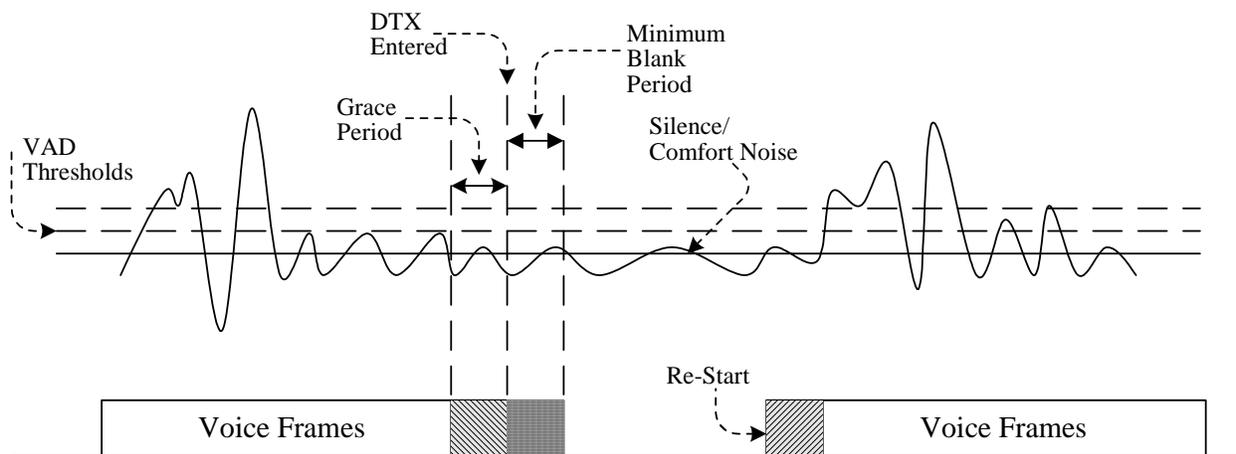
B.0 DISCONTINUOUS VOICE (DTX)

This appendix describes the requirements associated with Discontinuous Voice (DTX Voice) Operation beyond that described within the signaling plan itself. DTX voice operation is described in general terms with specific values provided in Tables associated with particular modes of operation.

The following features must be managed during DTX voice operation.

- Voice Activity Detection (VAD)
- Grace Period
- Blank Period
- Comfort Noise
- ReStart

Figure B.1 provides a pictorial description of the above features.



6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165

Figure B-1 DTX Voice

B.1 Voice Activity Detection (VAD)

Voice activity detection is used to determine whether speech is present or not in an input signal. A voice activity detection method shall be implemented such that a Voice Activity Factor (VAF), as specified in Table B.1-1, is achieved in accordance with the SCIP DTX Voice VAF performance criteria specified in SCIP-210 Appendix C – PERFORMANCE REQUIREMENTS. The voice activity detection (VAD) algorithm described below is provided as a default solution. Source code for this default VAD algorithm is available as GFE. The GFE source code shall have precedence over the description provided below.

6166
6167
6168
6169

Table B.1-1 DTX VAF Values

Voice Mode	Voice Activity Factor
MELP Blank and Burst	≤ 0.6

6170
6171

Editor's Note: MELP Blank and Burst VAF of ≤ 0.6 is relative to testing performed with test vectors provided by the Government. The VAF test vectors are available from the International ICWG Web site (<http://198.184.128.72/iicwg>) or on disk from NSA.

6172
6173
6174
6175

B.2 Default Voice Activity Detection (VAD) Algorithm

6176
6177
6178
6179

The following VAD is provided as a default solution. The GFE source code shall have precedence over the description provided below. The VAD uses the energy level of the input speech to determine whether speech or silence is present. The equation

6180

$$Energy = \sqrt{(A^H \times A) / (FrameSize)}$$

6181
6182
6183
6184
6185
6186
6187

is used to calculate the energy of each speech frame, where A is a vector of one frame of input data, A^H is the complex conjugate transpose of A , and $FrameSize$ is the number of samples per vocoder frame. The minimum (Low RMS) and maximum (High RMS) energy levels are set based on the energy of the input vector. These values are used to calculate an energy threshold that is compared to the present frame's energy level. The equation

6188

$$Threshold = (0.07 \times HighRMS) + (K \times LowRMS),$$

6189
6190
6191
6192
6193
6194

where K is a constant, is used to calculate the energy threshold. If frame's energy is less than the threshold, then the frame is marked as silence. If more than four consecutive frames of speech have energy levels less than the threshold, then it is determined that silence is detected and comfort noise is written out. This mode continues until an input vector's energy level is above the threshold.

6195
6196 In order to compensate for low energy anomalies, the minimum energy value is slowly increased
6197 each time through the loop by a defined delta,
6198

$$6199 \quad \text{LowRMS} = \text{LowRMS} \times \text{DeltaUp}.$$

6200
6201 *DeltaUp* is initially set to 1.01 and is adjusted depending on whether the LowRMS is reset or not
6202 as follows

$$6203 \quad \text{DeltaUp} = \text{DeltaUp} \times 1.0001.$$

6204
6205 **Editor's Note:** Source code for the default VAD is available, as GFE, from the International
ICWG Web site (<http://198.184.128.72/iicwg>) or on disk from NSA.

6206
6207
6208 **B.3 Grace Period**

6209
6210 The Grace Period is a variable period of silence/background noise that is transmitted after silence
6211 is detected and before DTX mode is entered.

6212
6213 The Grace Period shall contain a minimum of two (2) vocoder frames. These vocoder frames
6214 shall be uniquely identifiable as silence. The information being transmitted in the Grace Period
6215 vocoder frames shall contain vocoder compatible parameters, such that processing these frames
6216 through the vocoder does not produce unacceptable noise.

6217
6218 For MELP Blank and Burst, the Grace Period shall be populated with MELP vocoder frames as
6219 defined in Table B.3-1 – MELP Comfort Noise Parameter Values. All MELP vocoder parameter
6220 values shall be set to zero (0) except *msvq*[0], *gain*[1] and *sync*.

6221

6222
6223
6224

Table B.3-1 MELP Comfort Noise Parameter Values

MELP Parameter	Value
msvq[0] (line spectral frequencies)	* See Note (1)
msvq[1] (line spectral frequencies)	Set to 0
msvq[2] (line spectral frequencies)	Set to 0
msvq[3] (line spectral frequencies)	Set to 0
fsvq (Fourier magnitudes)	Set to 0
gain[0] (gain)	Set to 0
gain[1] (gain)	* See Note (1)
pitch (pitch – overall voicing)	Set to 0
bp (bandpass voicing)	Set to 0
af (aperiodic flag/jitter index)	Set to 0
sync (sync bit)	Continue Alternations

6225
6226
6227
6228
6229
6230
6231
6232

Notes:

1. The default value shall be the respective parameter value from the previous vocoder frame. It is recommended that msvq[0] and gain[1] values be derived by averaging the respective parameters from some number of previous vocoder frames.

B.4 Blank Period

The Blank Period is defined as a variable amount of time that DTX mode (no voice traffic transmissions) must be executed once it has been entered.

The Blank Period shall have a minimum duration equivalent to “*n*” vocoder frames as defined in Table B.4-1.

6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243

Table B.4-1 Blank Period Values

Voice Mode	Blank Period “<i>n</i>”
MELP Blank and Burst	2

6244

6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277

B.5 Comfort Noise

Comfort noise is generated so that a user is not annoyed by the disappearance of background noise during periods of silence. It is recommended that comfort noise be generated and provided to the user at the receiver.

For MELP Blank and Burst, the MELP vocoder frame defined in Table B.3-1 shall be used as the comfort noise value. The default comfort noise method shall be to repeat the vocoder frame from the Grace Period at the receiver. It is recommended that the averaged values of these parameters be computed at the transmitter and inserted as the Grace Period frames.

Editor's Note: Generation of comfort noise for GSM is specified in GSM standards 6.12, 6.22 and 6.62.
--

B.6 Re-Start

Upon detection of voice activity, voice traffic mode shall be re-entered, after fulfilling the minimum Blank Period, by sending a Re-Start message.

For MELP Blank and Burst, the Re-Start message shall be the Sync Management Frame as defined in SCIP-210 Section 3.3.1.1.

Upon the Re-Start of voice traffic, there are three alternative ways to manage the onset of voice activity and associated voice quality issues:

- BUFFER/DELAY initial vocoder frame while sync management frame is sent;
- CLIP initial vocoder frame and substitute sync management frame; and
- Skew Time by comparing several vocoder frames and delete, prior to encryption, the least useful vocoder frame to make room for the Sync Management frame.

For the BUFFER/DELAY option, a maximum delay equivalent to one (1) vocoder frame is permitted.

6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302

THIS PAGE INTENTIONALLY LEFT BLANK.

6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320

C.0 PERFORMANCE

C.1 DTX Voice

The Voice Activity Detection algorithm shall provide a Voice Activity Factor (VAF) as defined in SCIP-210 Appendix B, Table B.1-1 – DTX VAF Values.

C.1.1 MELP Blank and Burst

The VAF of ≤ 0.6 shall be measured as the percentage of all frames transmitted, including voice, silence (e.g. during silence detection period) and Grace Period frames, while processing the Government provided test vectors. The test vectors are heli_mp_rh.spd, jeep_ch_vw.spd and off_ch2_vw.spd. These test vectors are available from the International ICWG Web site (<http://198.184.128.72/iicwg>) or on disk from NSA.