Frame Relay Privacy Implementation Agreement

FRF.17

Frame Relay Forum Technical Committee

January 21, 2000

Note: The user's attention is called to the possibility that implementation of the frame relay implementation agreement contained herein may require the use of inventions covered by patent rights held by third parties. By publication of this frame relay implementation of the specification will not infringe on any third party rights. The Frame Relay Forum takes no position with respect to any claim that has been or may be asserted by any third party, the validity of any patent rights related to such claims, or the extent to which a license to use such rights may or may not be available.

For additional information contact:
The Frame Relay Forum
Worldwide Headquarters
39355 California St., Suite 307
Fremont, CA 94538
Phone: + 1 510.608.5920
FAX: + 1 510.608.5917
E-mail:frf@frforum.com

Editor: David Sinicrope
Lucent Technologies

Table of Contents

List of Tables

List of Figures

Revision History

Version	Date	Changes
FRF priv	January 21, 2000	baseline document

1 INTRODUCTION

1.1 PURPOSE

This document is a frame relay privacy implementation agreement. This Implementation Agreement (IA) covers the case where the user's equipment is attached to a non-ISDN frame relay network or to an ISDN network when using case A only.

The agreements herein were reached in the Frame Relay Forum, and are based on the relevant frame relay standards referenced in Section 1.3. They address the optional parts of these standards, and document agreements reached among vendors/suppliers of frame relay network products and services regarding the options to be implemented.

Except as noted, these agreements will form the basis of conformance test suites produced by the Frame Relay Forum.

This document may be submitted to different bodies involved in ratification of implementation agreements and conformance testing to facilitate multi-vendor interoperability.

1.2 DEFINITIONS

- Must, Shall, or Mandatory **-** the item is an absolute requirement of this implementation agreement.
- Should **-** the item is highly desirable.
- May or Optional **-** the item is not compulsory, and may be followed or ignored according to the needs of the implementor.
- Not Applicable **-** the item is outside the scope of this implementation agreement.

1.3 RELEVANT STANDARDS

The following is a list of standards on which these implementation agreements are based upon:

[1] FRF.1.2, D. Sinicrope (ed.), User-to-Network Implementation Agreement (UNI), Frame Relay Forum, January 19, 2000.

[2] FRF.3.1, R. Cherukuri (ed.), Multiprotocol Encapsulation Implementation Agreement, Frame Relay Forum, June 22, 1995.

[3] FRF.9, D. Cantwell (ed.), Data Compression Over Frame Relay Implementation Agreement, Frame Relay Forum, January 22, 1996.

[4] FRF.12, A. Malis (ed.), Frame Relay Fragmentation Implementation Agreement, Frame Relay Forum, December 1997.

[5] ITU-T Recommendation Q.922, ISDN Data Link Layer Specification for Frame Mode Bearer Services, ITU, Geneva, 1992.

[6] ITU-T Recommendation Q.921, ISDN User-Network Interface-Data Link Layer Specification ITU, Geneva, 1993 .

[7] ITU-T Recommendation Q.933, ISDN Signaling Specifications for Frame Mode Switched and Permanent Virtual Connections Control and Status Monitoring, ITU, Geneva, 1995.

[8] ITU-T Recommendation Q.931, ISDN User-Network Interface Layer 3 Specification for Basic Call Control, ITU, Geneva, 1993.

[9] RFC 1661/STD 51, W. Simpson (ed.), PPP Link Control Protocol, IETF, July 1994

[10] RFC 1968, G. Meyer, The PPP Encryption Control Protocol (ECP), IETF, June 1996

[11] RFC 1969, K. Sklower, G. Meyer, The PPP DES Encryption Protocol, IETF, June 1996

[12] ANSI X3.92-1981, Data Encryption Algorithm, ANSI, New York, 1981

[13] RFC 1570, W Simpson (ed.), PPP LCP Extensions, IETF, January 1994

[14] RFC 1851, P. Karn, et al., ESP Triple DES Transform, September 1995.

[15] RFC 2284, Blunk, Vollbrecht, PPP Extensible Authentication Protocol, March 1998.

[16] RFC 2451, R. Pereira, R. Adams, The ESP CBC-Mode Cipher Algorithms, November 1998.

[17] ISO/IEC 9979, Information technology -- Security techniques -- Procedures for the registration of cryptographic algorithms, ISO/IEC, Geneva, 1999.

[18] B. Schneier, "Applied Cryptography Second Edition", John Wiley & Sons, New York, NY, 1995. ISBN 0-471-12845-7

2 IMPLEMENTATION AGREEMENTS

The remainder of this document contains the agreements reached in the Frame Relay Forum.

2.1 OVERVIEW

The Frame Relay Privacy Protocol (FRPP) includes three facilities, authentication, encryption and key exchange. These protocols are based on PPP Link Control Protocol (RFC 1661), PPP Encryption Control Protocol (RFC 1968 and 1969) and Frame Relay Forum Data Compression (FRF.9).

The FRPP allows optional authentication procedures that are based on the PPP authentication functions. Authentication in PPP is also optional.

FRPP provides two modes of operation for the encryption facility:

- Mode 1: Mode 1 is the mandatory mode and uses the default algorithms and frame formats defined in this I/A. It allows negotiation of parameters and use of the optional Key update facility. The default encryption algorithm is ANSI X3.92, the US Data Encryption Standard (DES) with Cipher Block Chaining (CBC). This is a standardized encryption algorithm and is the default used in PPP Encryption Control Protocol (RFC 1968). To facilitate export and importability, DES is used for Class A compliant devices. For more robust security "Triple DES" is used for Class B compliant devices.
- Mode 2: Mode 2 is optional and allows full negotiation of encryption algorithms, both standard and proprietary, and their associated parameters. This mode uses the Encryption Control Protocol for PPP (RFC 1968). Support of key exchange within the Encryption Control Protocol for PPP is for further study.

FRPP provides an optional facility under Mode 1 encryption for key update.

FRPP relies on different phases similar to PPP. The order of the phases is as follows:

1. VC Establishment - This phase is controlled by the signaling procedures (PVC (FRF.1.2) or SVC (FRF.4.1)) and is outside the scope of this agreement. The VC is assumed to be established.
2. Authentication phase- When used, initial peer authentication must be done prior to the encryption phase. Subsequent authentication challenges, if supported by the chosen authentication protocol, may be done during the data transfer phase. Authentication messages are clearly identified in the FRPP header with the A bit.
3. Encryption negotiation phase - Used to negotiate the mode and parameters to be used for encryption during the data transfer phase.
4. Data Transfer Phase - Transfer of ciphered messages which includes the user data, some control information and key updates.

2.1.1 Scope

Frame Relay Privacy Protocol (FRPP) is used per VC, end user to end user. It allows negotiation of encryption and authentication protocols, key update and transport of encrypted data. This protocol is used exclusively on the frame relay user plane, i.e., DLCIs used for user data transfer, not DLCI 0. FRPP is intended as a privacy protocol to discourage data observation. It is not intended to prevent every type of security attack. Precautions for critical data should be made at higher layers. This protocol provides some protection against passive eavesdropping but essentially no security against an active adversary who can forge or alter messages.

2.1.2 Compliance

Compliance to the I/A is according to the following table of facilities:

Facility	Support
Authentication	optional
Encryption • Mode 1 • Mode 2	mandatory (one of the following) Class A – DES Class B – three key EDE Triple DES Class AB – DES or three key EDE Triple DES optional
Mode 1 Key Update	optional

2.2 FACILITIES AND INTERDEPENDENCE

This specification allows the selection and specification of authentication, encryption and key update. Authentication and encryption facilities are independent of each other and may be used together or individually. The key update facility is optional and is only used when the mode 1 encryption facility is used.

2.3 REFERENCE MODEL

When used between DTEs, as shown in Figure 1, the privacy procedure is transparent to Frame Relay network(s) between the transmitting and receiving DTEs. Note that the functionality specified in this Implementation Agreement has been illustrated as an integrated privacy function, but stand-alone implementations are not prohibited.

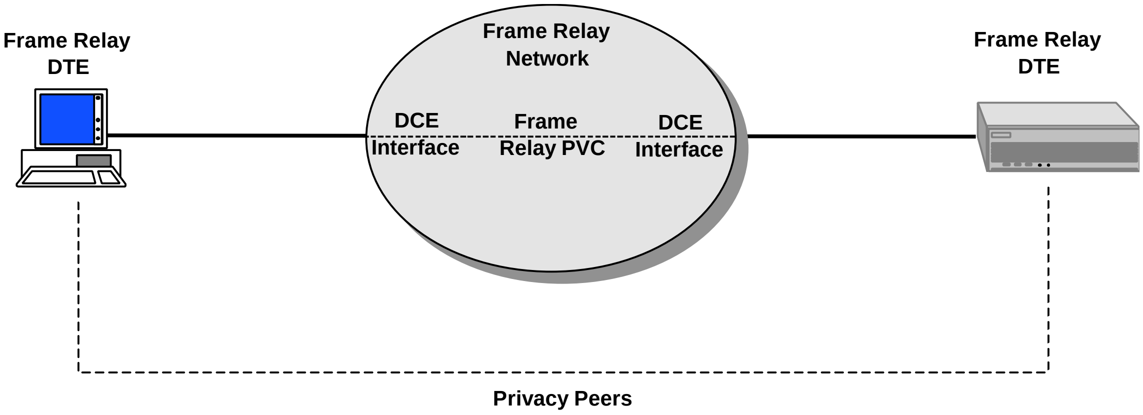


Figure 1 - Reference Diagram

3 COMMON MODE SPECIFICATION:

This section defines the frame formats and procedures common to all facilities.

3.1 GENERAL FRAME FORMAT

Figure 2 is the general frame structure for the frame relay privacy protocol (FRPP). All frames are sent on the frame relay virtual connection between end systems and follow the frame format of FRF.3.1. The frame contents are transparent to the frame relay network. The general frame format in Figure 2 is used for both negotiation control as well as data transfer. Control frames contain the information vital to negotiating the authentication, encryption and key exchange and their parameters. The data transfer frames contain the actual data if sent encrypted. The distinction between control and data frames is explained below.

<table>
<tr><th colspan="4">Description</th><th>Octet</th></tr>
<tr><td colspan="4">Q.922 Address</td><td>1</td></tr>
<tr><td colspan="4">(2 octets)[1]</td><td>2</td></tr>
<tr><td colspan="4">Control
(UI: 0x03)</td><td>3</td></tr>
<tr><td colspan="4">NLPID
(0xB3)</td><td>4</td></tr>
<tr><td colspan="4">FRPP Header</td><td></td></tr>
<tr><td>ext
1</td><td>Spare</td><td>A</td><td>C/D</td><td>5</td></tr>
<tr><td colspan="4">FRPP Payload</td><td>6
n</td></tr>
<tr><td colspan="4">FCS</td><td>n+1</td></tr>
<tr><td colspan="4">(2 octets)</td><td>n+2</td></tr>
</table>

Figure 2 - Frame Relay Privacy Protocol Frame Format

Field	Description
Q.922 Address	Frame Relay Address structure containing DLCI, FECN, BECN, DE and C/R. The C/R bit is not used
Control	Frame Relay Q.922 Unnumbered information frame (UI) (x03)
NLPID	Network Layer Protocol ID from ISO/IEC TR9577
FRPP Header	The Frame Relay Privacy Protocol Header consists of the following: ▪ extension bit **-** set to one, but included for future enhancement ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) ▪ Authentication (A) bit **-** Can only be set to 1 when C/D = 1. Indicates that frame contains authentication information ▪ Control / Data (C/D) bit 0 Data Frame 1 Control Frame
FRPP Payload	Control information or transfer data depending on how the FRPP Header bits are set.
FCS	Q.922 Frame Check Sequence

Table 1 - Frame Relay Privacy Protocol Frame Format

If the FRPP Header C/D bit is set to 1, the frame is a control frame. The Authentication bit (A) bit then determines whether the frame is an authentication frame or an encryption control frame. When the A bit is 1 it indicates that the frame contains authentication information using the format in section Authentication Facility. Otherwise the A bit is set to 0, indicating that the frame contains encryption control information. The frame format for encryption control is described in section Encryption Facilities.

If the C/D bit is set to 0, the frame is a data transfer frame. The A bit is not applicable in this case. The format of a data transfer frame is dependent on the encryption facility mode chosen, Mode 1 or 2. Mode 1 data transfer formats are described in section Mode 1 Data Transfer Format. The format of Mode 2 data transfer frames is described in section Mode 2 Data Transfer.

[1] The 2 octet frame relay address (DLCI) is shown here for illustrative purposes. The 3 and 4 octet address formats are not prohibited.

4 AUTHENTICATION FACILITY

This facility is used to authenticate two devices based on a pre-selected authentication protocol. The authentication facility is optional. If authentication is desired, an implementation must perform the initial authentication *before* invoking the encryption facility.

Authentication packets are identified via the A bit in the FRPP header of a control message (C/D bit=1). In general the authentication features of PPP are used. The authentication protocol is configured in each peer device for PVCs or negotiated during call establishment for SVCs. Specifics of the SVC signalling are for further study. The authentication protocol is identified in octets 6, 7, and 7a, if applicable. Octets 8-n contain authentication information or configuration options in an authentication packet format specific to the protocol identified in octet groups 6 & 7.

The FRPP Authentication mechanism supports any of the authentication protocols defined for PPP. For example, the PPP Extensible Authentication Protocol (EAP) which itself supports a number of authentication protocols, PPP Challenge Handshake Authentication Protocol (CHAP) and PPP Password Authentication Protocol (PAP).

This section describes the format and structure to support PPP authentication. Details of the PPP authentication protocols may be found in each of the individual PPP authentication RFCs.

4.1 AUTHENTICATION FRAME FORMAT

Description				**Octet**
Frame Relay address, control and NLPID information				1 - 4
FRPP Header				
ext 1	Spare	A 1	C/D 1	5
Authentication Protocol ID				6
Note 2				7
Authentication Algorithm Note 2				7a* (Note 1)
Authentication Packet Format Note 2				8 n
FCS				n+1
(2 octets)				n+2

Note 1: Octet 7a is only present if octets 6 and 7 indicate CHAP (xC223)
Note 2: Contents and formats are defined from PPP RFCs.

Figure 3 - General Authentication Frame Format

Field	Description
DLCI, control and NLPID	See section General Frame Format for details.
FRPP Header	▪ ext: extension bit = 1 ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) ▪ Authentication (A) bit = 1 ▪ Control / Data (C/D) bit 1 Control
Authentication Protocol ID (octets 6 and 7)	Identifies the authentication protocol to be used, e.g. PAP, CHAP, etc. The values for this field are taken from the PPP protocol field values. See RFC 1340 "Assigned Numbers" for most recent values.
Authentication Algorithm (octet 7a)	If present, identifies the CHAP authentication method to be used. See RFC 1994 for details. Only present if octets 6 and 7 indicate CHAP.
Authentication Packet Format (octets 8-n)	In general, uses packet format of specific PPP authentication method. See section Authentication Packet Format for detail.
FCS	Q.922 Frame Check Sequence

Table 2 - General Authentication Frame Format

4.1.1 Authentication Packet Format

The PPP style packets shall be encapsulated within octets 8-n of the frame format above. These packets are of the general format: Code, Identifier, Length, Values. For example, in the CHAP case, the packet format of RFC 1994 section 4 is used.

Description	Octet
Authentication Packet Format Code	8
Identifier	9
Length (2 octets)	10 11
Values/Data as defined by authentication protocol	12 n

Figure 4 - Authentication Packet Format

Field	Description
Code	from PPP Authentication method indicated in octets 6-7a. usually indicates the type of packet or message, e.g. Request, Response, etc.
Identifier:	A transaction number to correlate a request with a response. Sent in request and echoed in corresponding response.
Length (2 octets)	Including: Code, Identifier, Length and all Configuration Options
Configuration Options	Values/Data depending on the PPP Authentication protocol used. See specific PPP Authentication method for details. E.g., CHAP, EAP, PAP, etc.

Table 3 - FRPP Control Primitive Structure

4.2 AUTHENTICATION PROCEDURES

Follow the procedures described in the applicable RFC for the PPP authentication protocol used. For example, if the authentication protocol configured is CHAP (xCC23) then follow the procedures in RFC 1994.

5 ENCRYPTION FACILITIES

The encryption facility is responsible for enabling and initiating data encryption algorithms on both ends of the point to point link. Encryption uses a similar packet exchange mechanism as the PPP Link Control Protocol.

The use of the encryption facility is negotiated between peer devices. The mode and algorithms are selected independently for each direction of a virtual connection.

Requested Mode	**Configured Mode for DTE**	
	Mode 1	**Mode 2**
Mode 1	Respond with Mode 1 and use Mode 1. See section Mode 1 Specification	Respond with Mode 1 and use Mode 1. See section Mode 1 Specification
Mode 2	Respond with Mode 1 and use Mode 1. See section Mode 1 Specification	Respond with Mode 2 and use Mode 2. Mode 2 Specification

Table 4 -Mode Transition Table

It is assumed that each peer device has an initial secret to be used for encryption. The method by which the secret becomes known to both communicating devices is outside the scope of this agreement.

Encryption negotiation must be completed successfully before allowing transfer of data frames. Once negotiated all data frames exchanged on a VC are encrypted.

5.1 MODE 1 SPECIFICATION

Mode 1 encryption support is mandatory for implementation agreement compliance. Mode 1 consists of a simple handshake to enable the default encryption algorithm and negotiate the encryption parameters for both directions of the VC. The default encryption algorithm for Class A compliant devices is the US Data Encryption Standard (DES)[12] with Cipher Block Chaining, (CBC). For Class B compliant devices, “Triple” DES is used. See section 5.1.5. The cyphertext is transferred using the packet format defined in section 5.1.2 Mode 1 Data Transfer Format.
This mode has two parameters to be negotiated: Initialization vector and Key Update capability.

5.1.1 Mode 1 Control Frame Formats

This frame is used to negotiate Mode 1 parameters.

5.1.1.1 Mode 1 Control Message

<table>
<tr><th colspan="5">Description</th><th>Octet</th></tr>
<tr><td colspan="5">Frame Relay address, control and NLPID information</td><td>1-4</td></tr>
<tr><td colspan="5">FRPP Header</td><td>5</td></tr>
<tr><td>ext
1</td><td>Spare</td><td>Mode 1
Compliance
Class</td><td>A
0</td><td>C/D
1</td><td></td></tr>
<tr><td colspan="5">FRPP Control Primitive
(Note 1)
Code</td><td>6</td></tr>
<tr><td colspan="5">Identifier</td><td>7</td></tr>
<tr><td colspan="5">Length
(2 octets)</td><td>8
9</td></tr>
<tr><td colspan="5">Configuration Option
Type: Mode 1 (254)</td><td>10</td></tr>
<tr><td colspan="5">Length</td><td>11</td></tr>
<tr><td colspan="2">Negotiation Codes</td><td colspan="3">Version</td><td>12</td></tr>
<tr><td colspan="5">Parameter Elements</td><td>13
n</td></tr>
<tr><td colspan="5">FCS
(2 octets)</td><td>n+1
n+2</td></tr>
</table>

Note 1: FRPP Control Primitive includes octets 6-n

Figure 5 - Mode 1 Control Frame

Field	Description
DLCI, control and NLPID	See section General Frame Format for details.
FRPP Header	▪ ext: extension bit = 1 ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) (3 bits)Mode 1 Compliance Class 0 0 Reserved 0 1 Class A 1 0 Class B 1 1 Class AB ▪ Authentication (A) bit = 0 ▪ Control / Data (C/D) bit 1 Control
FRPP Control Primitive Code	1 - Config-Req 2 - Config-Ack 14 - Reset Req 15 - Reset Ack (values given in decimal)
Identifier:	A transaction number to correlate a request with a response. Sent in request and echoed in corresponding response.
Length (2 octets)	Including: Code, Identifier, Length and all Configuration Options
Configuration Options Type	254 (decimal)- indicates Mode 1

Field	Description
Length	varies depending on number of parameters
Negotiation Codes	4bits 0000 reply with Response only 0001 reply with Response and initiate Request all other values are reserved
Version	Version number of the FRPP I/A 4 bits 0001 Version 1.0
Parameter Elements	Zero or more of the Mode 1 parameter elements. See section 5.1.1.2 Mode 1 Parameter Elements below
FCS	Q.922 Frame Check Sequence

Table 5- Mode 1 Control Frame

5.1.1.2 Mode 1 Parameter Elements

The Parameter Element ID identifies a parameter element. The length is the length of the whole parameter element including the Parameter Element ID field and the Length field. The Values field lists the individual parameter values of the element. The parameter elements must consist of an integral number of octets. These start after octet 12 of the Mode 1 configuration option

Description	Octet
Parameter Element ID	a
Length	b
Parameter Element Values	c m

Figure 6 - Mode 1 General Parameter Element Structure

5.1.1.2.1 Mode 1 Initialization vector

Inclusion of the initialization vector parameter in the Config-Req is optional. The presence of the initialization vector parameter in the Config-Req, indicates the 64-bit initial nonce the sending device will use for the cipher block chaining (CBC). The Config-Ack acknowledges receipt of the initialization vector. The initialization vector parameter is not sent in the Config-Ack.

If the initialization vector parameter is not included in the Config-Req, a secret known to both devices is used to derive the initialization vector. The means by which the secret and derivation procedure becomes known to both devices is beyond the scope of this agreement.

Initialization vector ID 0 0 0 0 0 0 0 1	1
Length: 10	2
Initial Nonce (8 octets)	3 10

Figure 7 - Mode 1 Initialization vector (Nonce) Parameter Element

Field	Description
Initialization vector ID	Octet identifying the initialization vector parameter
Length	10 (decimal)
Initial Nonce	64-bit quantity that is used by the peer device to encrypt the first packet transmitted. To guard against replay attacks the device should offer a different value during each negotiation.

Table 6 - Mode 1 Initialization vector (Nonce) Parameter

5.1.1.1.1 Mode 1 Key Update Parameter

This parameter is used to enable the optional key update facility. Inclusion of the key update parameter in the Config_Req is optional. The presence of the key update parameter in the Config-Req, indicates that sending device is requesting the optional automated key update capability be used in this direction. If the receiver supports the key update capability it shall include the key update parameter in the Config-Ack. This indicates that the receiver will receive and process key update messages in the direction of the Config_Reqest. This facility is negotiated independently per direction.

If the receiver does not support the key update capability, it does not include the key update parameter in the Config-Ack.

Key Update ID								
0	0	0	0	0	0	1	0	1
Length: 3								2
Version								3

Figure 8 - Mode 1 Key Update Parameter Element

Field	Description
Key Update ID	Octet identifying the key update parameter
Length	3 (decimal)
Version	Number indicating the key update capability version.

Table 7 - Mode 1 Key Update Parameter Fields

5.1.2 Mode 1 Data Transfer Format

This format is used for transferring ciphered data.

Description			Octet
Frame Relay address, control and NLPID information			1-4
FRPP Header			
ext 1	Spare	C/D 0	5
Sequence number			6
Ciphertext Identifier (Note 1)			7
User Data(Note 1)			8 m
Self Describing Padding (Note 1)			m n
LCB			n+1
FCS (2 octets)			n+2 n+3

Note 1: This field is encrypted.

Figure 9 - Mode 1 Data Transfer Frame Format

Field	Description
DLCI, control and NLPID	See section General Frame Format for details.
FRPP Header	▪ ext: extension bit ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) ▪ A bit: Authentication bit - Not applicable. ▪ C/D: Control / Data bit 0 Data
Ciphertext Identifier	Identifies the type of ciphered text 0 User Data 1 Key Update all other values are reserved
Sequence Number	Number assigned by the encryptor sequentially starting with 0 and incremented modulo 256.
User Data	The generation of this data is described in section Mode 1 Data Transfer Procedures User Data Encryption and Self Describing Padding
Self Describing Padding	The generation of this data is described in section Mode 1 Data Transfer Procedures User Data Encryption and Self Describing Padding
LCB	Longitudinal Check Byte - calculated on cleartext of octets 7 through n. Details of the calculation are in section Longitudinal Check Byte
FCS	Q.922 Frame Check Sequence

Table 8 - Mode 1 Data Transfer Frame Format

5.1.3 Mode 1 Control procedures

FRPP Mode-1 provides a simple negotiation protocol to enable privacy function with the default algorithm and parameter values. Once FRPP is successfully enabled, data transfer to the peer end system may be encrypted. To disable FRPP, an implementation may force the virtual connection to the inactive state, or send a Mode-1 request and not send a Mode-1 response.

FRPP Mode-1 consists of three phases: Disabled, Initialization, and Operation. The Disabled phase is entered upon power-up or when a frame relay virtual connection is released. The Initialization phase is entered upon frame relay virtual connection establishment and when FRPP is enabled. The Operation phase is entered upon the successful completion of the Initialization phase. Unsuccessful completion of the Initialization phase causes FRPP to enter the Disabled Phase. FRPP data PDUs are transferred only when Mode-1 is in the Operation phase. FRPP control PDUs may be transferred in any phase.

FRPP Mode 1 PDUs are tagged as Class A, Class B or Class AB depending on the default encryption support in the sending device. See Table 5- **Mode 1 Control Frame**. The Configure Request and Configure Response messages are tagged according to the following table.

		Requested Class		
		A	**B**	**AB**
Supported Class	**A**	Respond w/ A	Ignore	Respond w/ A
	B	Ignore	Respond w/ B	Respond w/ B
	AB	Respond w/ A	Respond w/ B	Respond w/ B

Table 9 - Negotiation of Mode 1 class

If the message is ignored, notification should be sent to the management system that the two peer devices are misconfigured.

5.1.3.1 Mode 1 States

The FRPP Mode-1 states which may exists on either side of the frame relay connection are:

Disabled (D): (initial state)
FRPP does not exist.

Request Initiated (I_1)

A Mode-1 configuration request message has been sent to the peer. Awaiting response to own request and peer configure request.

Request Received (I_3)

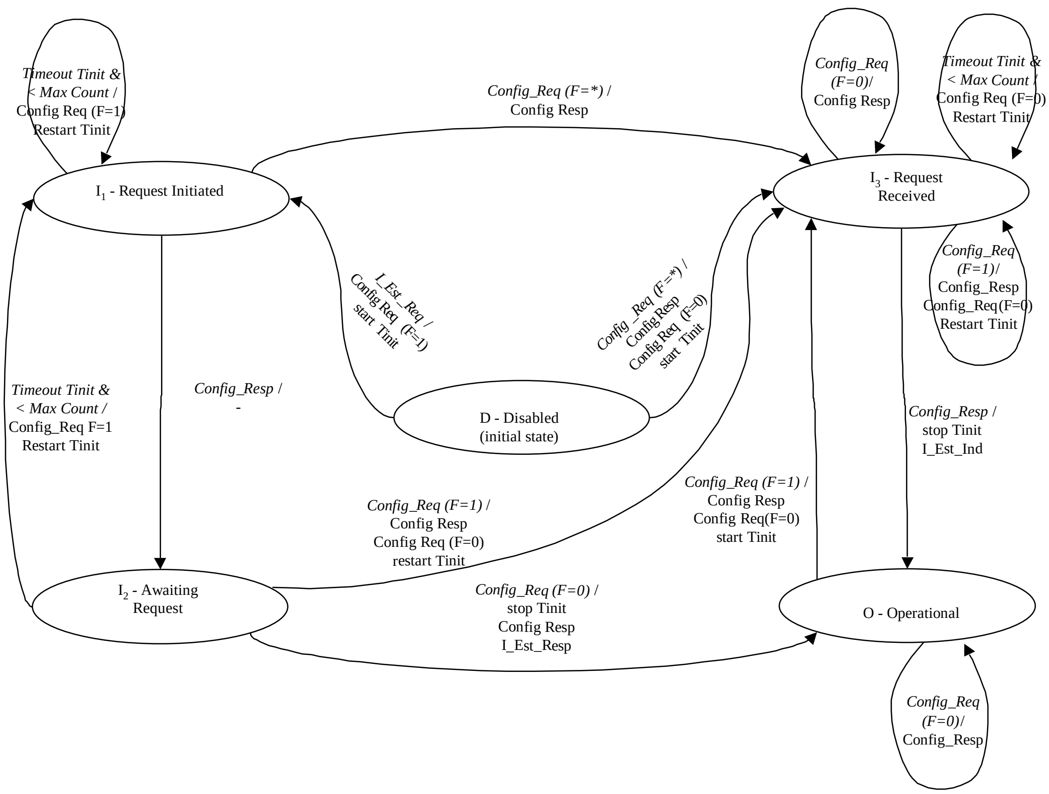
A Mode-1 configuration request message has been received from the peer. Configure response to peer request message and a configure request are send to the peer. Awaiting response to own request.

Awaiting Request (I_2)

Received configure repose to own request and awaiting for the peer configure request.

Operational (O)

Mode-1 negotiation completed.



Note: F indicates the Negotiation Codes defined in Table 5**- Mode 1 Control Frame**.

Figure 10 - Mode 1 State Diagram

5.1.3.1 Initialization Request

The Mode-1 Initialization will start when a frame relay virtual connection to a peer is established and FR privacy function is administratively enabled (by the user). A signal for PVC establishment is obtained via link management Procedures (e.g. Q.933 Annex A) when there is a transition from a PVC status of “inactive” to “active” and/or the presence of both “active” and “new” for a PVC. A signal for SVC establishment is obtained via Q.933 call control procedures when a call transitions to the Active state (U10 or N10).

Frame Relay privacy protocol negotiation procedures are initiated, by sending a Configure_Request message with negotiation codes set to "reply with response and initiate request" to its peer; start a handshake completion timer and enter *Request Initiated (I_1)* state.

Upon receiving a Configure_Response message, the entity shall enter *Awaiting Request (I_2)* state. When a Configure_Request is received from the peer, the following procedure apply:

1. For calls in *Request Initiated (I_1)* state, send a Configure_Response message; enter *Request Received (I_3)* State.
2. For calls in *Awaiting Request (I_2)* state, the action taken depends on Configure_Request message negotiation code.
   - If the negotiation code is set to "reply with response only", send a Configure_Response message; Stop handshake completion timer; send a configure_response primitive that the negotiation is complete; enter *Operational (O)* state.
   - If the negotiation code is set to "reply with response and initiate request", send a Configure_Response message; sending a Configure_Request message with negotiation codes set to "reply with response"; to its peer; restart a handshake completion timer and enter *Request Received (I_3)* state.

If the handshake completion timer expires before the handshake procedure is completed and the number of retries are less than the count, the following procedure apply:

1. For calls in *Request Initiated (I_1)* state, send a Configure_Request message with negotiation codes set to "reply with response and initiate request" to its peer; restart a handshake completion timer.
2. For calls in *Awaiting Request (I_2)* state, send a Configure_Request message with negotiation codes set to "reply with response and initiate request" to its peer; restart a handshake completion timer; and enter *Request Initiated (I_1)* state.

5.1.3.2 Receipt of a Configuration Request

Upon receipt of a configuration request from the peer in *Disabled (D)* state, send a Configure_Response message; sending a Configure_Request message with negotiation codes set to "reply with response"; to its peer; start a handshake completion timer and enter *Request Received (I_3)* state.

Upon receiving a Configure_Response message in the *Request Received (I_3)* state, Stop handshake completion timer; send a configure_indication primitive that the negotiation is complete; enter *Operational (O)* state.

When a Configure_Request message is received from the peer, for calls in *Request Received (I_3)* state, the action taken depends on the message negotiation code:

- If the negotiation code is set to "reply with response only", send a Configure_Rsponse message.
- If the negotiation code is set to "reply with response and initiate request", send a Configure_Response message; sending a Configure_Request message with negotiation codes set to "reply with response" to its peer; restart a handshake completion timer.

If the handshake completion timer expires before the handshake procedure is completed and the number of retries are less than the count, send a Configure_Request message with negotiation codes set to "reply with response" to its peer; restart a handshake completion timer.

5.1.3.3 Operational Phase

When a Configure_Request message is received from the peer, for calls in the *Operational (O)* state, the action taken depends on the message negotiation code:

- If the negotiation code is set to "reply with response only", send a Configure_Response message.
- If the negotiation code is set to "reply with response and initiate request", send a Configure_Response message; sending a Configure_Request message with negotiation codes set to "reply with response" to its peer; restart a handshake completion timer and enter *Request Received (I_3)* state.

Note: If the key update facility was successfully negotiated , the key update interval timer is started at this point for the sending direction.

5.1.3.4 Disable Phase

The Mode-1 Disabled phase shall be entered when a frame relay virtual connection to a peer is released. A frame relay virtual connection is released when a signal for PVC inactive is obtained via link management Procedures (e.g. Q.933 Annex A), or a signal for SVC release is obtained via Q.933 call control procedures. N10).

If Max Count is exceeded on a handshake completion timer expiry, return to the Disabled state and notify the management entity.

5.1.4 Mode 1 Data Transfer Procedures

5.1.4.1 User Data Encryption and Self Describing Padding

Once the negotiation between the two encryption peers is completed and both peers are in the Operational state frames are encrypted using the procedures below.

The encryption method used to create the ciphertext fields for Mode 1 data transfer is the Data Encryption Standard (DES)[12] with the Cipher Block Chaining (CBC) mode. For Class A compliance the DES keys used by DES used as is. For Class B compliance the Triple DES is used as in [16]. For Class AB compliance both DES and three key EDE triple DES are supported and selection is made during the Mode 1 negotiation procedures according to Table 9 - Negotiation of Mode 1 class.

The initialization vector for the CBC mode is deduced from the explicit 64-bit nonce, exchanged during mode 1 negotiation. If no nonce is exchanged by the peers, it must be coordinated and configured in the respective peers of the virtual connection

The encryption CBC extends beyond each payload to the next. A sequence number is used to detect when a received frame is out of order.

Keys for the encryption algorithm, or initial keys in the case where key update is supported, are deduced from a mutual secret shared between two peers. How the secret becomes known to the two peers is beyond the scope of this specification.

When data is to be sent, the sender appends the User Data to a Ciphertext identifier that is set to 0. The data and the Ciphertext ID are padded to the next multiple of 8 octets as described below to form a cipherpayload. The LCB is calculated on the cipherpayload, see below. The encryptor ciphers the cipherpayload and the result is positioned into the frame as in Figure 9. The sender increments the sequence number modulo 256, appends the LCB onto the payload and sends the frame.

The receiver first checks the sequence number to determine if a frame was lost. If a frame was lost, the last 8 octets of the ciphertext are kept as the initialization vector for the next frame and the received frame is discarded. If the frame is in sequence, the receiver deciphers the fields identified in Figure 9 and calculates the LCB, see below. The calculated LCB, is compared to the received LCB. If they do not match the last 8 octets of the data are kept as the initialization vector for the next frame and the received frame is discarded. If the optional key update procedures are supported a key reset request may be sent to resynchronize the encryption keys. See section 5.1.6. If the LCBs match, the deciphered data is then processed by removing the ciphertext identifier and padding as described below. The decrypted user data is passed on to be processed.

The padding mechanism used is LCP Self-Describing Padding (SDP) ([13]section 2.2).

5.1.4.2 Longitudinal Check Byte

The LCB is calculated for each frame by

1. Exclusive ORing 0xFF to the first octet of the cleardata to be encrypted and storing the result. Then,
2. Each subsequent octet of the cleardata to be encrypted is XORed to the result generating the next value of the result.

5.1.5 “Triple” DES / 3DES / DES-EDE3-CBC

Below is a quote describing the Triple DES algorithm used for Class B. Analysis and further details on the algorithm can be found in [16].

> The DES-EDE3-CBC algorithm is a simple variant of the DES-CBC algorithm. The "outer" chaining technique is used. In DES-EDE3-CBC, an Initialization Vector (IV) is XOR'd with the first 64-bit (8 byte) plaintext block (P1). The keyed DES function is iterated three times, an encryption (Ek1) followed by a decryption (Dk2) followed by an encryption (Ek3), and generates the ciphertext (C1) for the block. Each iteration uses an independent key: k1, k2 and k3.
>
> For successive blocks, the previous ciphertext block is XOR'd with the current plaintext (Pi). The keyed DES-EDE3 encryption function generates the ciphertext (Ci) for that block.
>
> To decrypt, the order of the functions is reversed: decrypt with k3, encrypt with k2, decrypt with k1, and XOR the previous ciphertext block.
>
> Note that when all three keys (k1, k2 and k3) are the same, DES- EDE3-CBC is equivalent to DES-CBC. This property allows the DES-EDE3 hardware implementations to operate in DES mode without modification.
>
> For more explanation and implementation information for Triple DES, see [18].

5.1.6 Key Update Protocol for Mode 1

The Key Update Protocol is used in Mode 1 to automatically update the session keys. This allows the session key to be changed without the need for manual operator intervention. The transmitter is responsible for initiating the key update for the direction of the data on the VC.

5.1.6.1 Key Update Formats

This format is used for transferring key updates. Key updates are sent when the source device wishes to refresh the key used to encrypt the data being sent.

A different key is typically used per direction. The peer sending the data for a direction controls when the update procedures are used. When the peer wishes to update the key for the outbound direction it sends a Mode 1 Key Update message (ciphertext identifier equal = 1). The message contains a correlator information element, the key information element , and a key padding information element. The correlator information element is used to correlate the Key Update Acknowledge message returned by the receiving peer with the Key Update message. The Key information element contains the new session key to be used. The Key Padding information element is used to pad the message to a length similar to naormal data frames. This makes it more difficult to differentiate key update messages from normal data frames. The sequence number, self-describing padding and LCB fields are all used the same as if the key update messages were data transfer frames.

Description								Octet
Q.922 Address								1
(2 octets)								2
Control (UI: 0x03)								3
NLPID (0xB3)								4
FRPP Header								
ext 1	Spare						C/D 0	5
Sequence number								6
Ciphertext Identifier (Note 1)								7
Correlator IE Identifier (Note 1)								8
0	0	0	0	0	0	0	1	
Correlator IE Length (Note 1)								9
Correlator (Note 1)								10
Key IE Identifier (Note 1)								11
0	0	0	0	0	0	1	0	(Note 2) (Note 3)
Key IE Length (Note 1)								12
Key (Note 1)								13 k
Key Padding IE Identifier (Note 1)								k+1
0	0	0	0	0	0	1	1	
Key Padding IE Length (Note 1)								k+2
Key Padding (Note 1)								k+3 m
Self-Describing Padding (Note 1)								m n
LCB								n+1
FCS								n+2
(2 octets)								n+3

Note 1: This field is encrypted.
Note 2: Only included in the Key Update message
Note 3: For Class B or AB devices using Triple DES this information element is repeated such that all 3 keys are exchanged in the order k_1, k_2, k_3 at one time.

Figure 11 - Mode 1 Key Update Frame Format

Field	Description	
Q.922 Address	Frame Relay Address structure containing DLCI, FECN, BECN, DE and C/R. The C/R bit is not used	
Control	Frame Relay Q.922 Unnumbered information frame (UI) (x03)	
NLPID	Network Layer Protocol ID from ISO/IEC TR9577	
FRPP Header	▪ ext: extension bit ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) ▪ A bit: Authentication bit - Not applicable. ▪ C/D: Control / Data bit 0 Data	
Sequence Number	Number assigned by the encryptor sequentially starting with 0 and incremented modulo 256.	
Ciphertext Identifier	Identifies the type of ciphered text 0 User Data 1 Key Update 2 Key Update Acknowledge all other values are reserved	
Correlator IE	Correlator IE ID	identifies the correlator IE
	Correlator IE Length	length of the correlator IE including the ID and length fields
	Correlator	number assigned by the sender of the Key Update that is echoed in the Key Update Acknowledge to correlate the two messages
Key IE (may be repeated for Triple DES)	Key IE ID	identifies the key IE
	Key IE Length	length of the key IE including the ID and length fields
	Key	new key assigned by the sender to be used for frames received in this direction
Key Padding IE	Key Padding IE ID	identifies the key padding IE
	Key Padding IE Length	length of the key padding IE including the ID and length fields
	Key Padding	padding assigned by the sender to make key update messages appear as data in length. Note: this does NOT include the Self Describing Padding needed to pad the cipherpayload to the next multiple of 8 octets.
Self-Describing Padding	Used to pad frame to next multiple of 8 octets. The padding mechanism used is LCP Self-Describing Padding (SDP) ([13]section 2.2).	
LCB	Longitudinal Check Byte - calculated on octets 6 through n. Details of the calculation are in section Longitudinal Check Byte	
FCS	Q.922 Frame Check Sequence	

Table 10 - Mode 1 Key Update Frame Format

5.1.6.2 Key Update Procedures

The *Key Update Pending* and *Awaiting New Key* states are substates of the operational state used in negotiating the Mode 1 parameters.

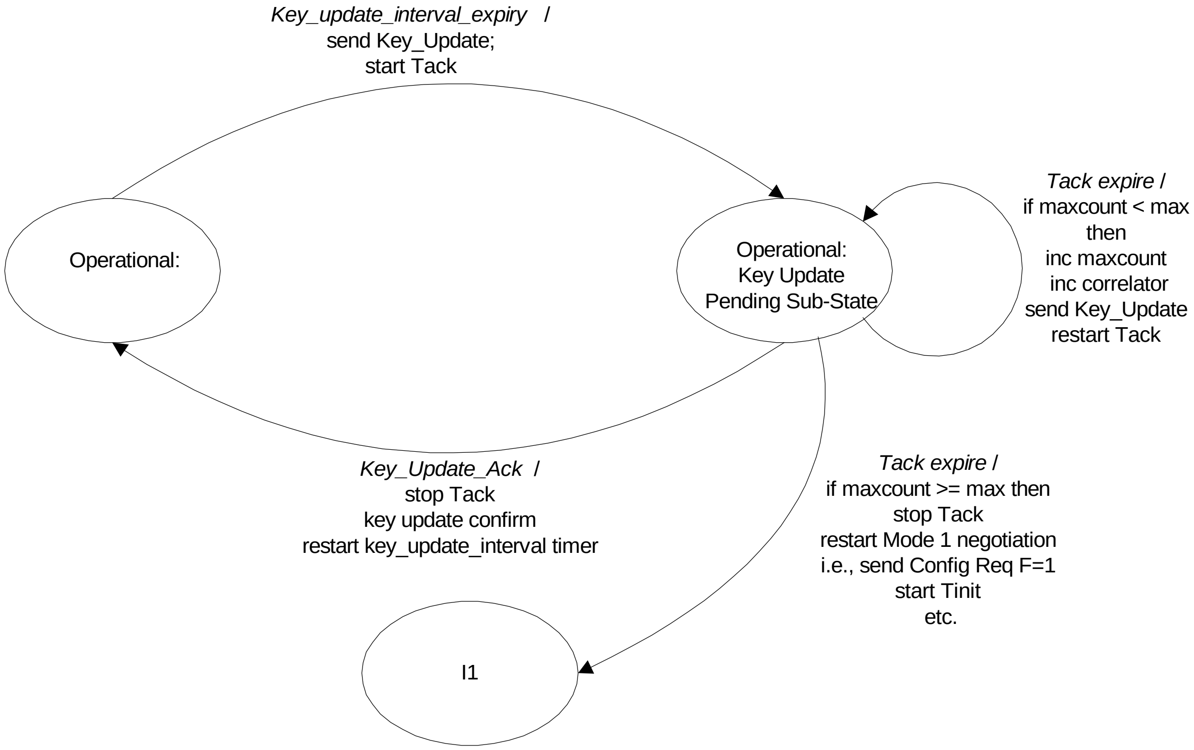


Figure 12 - Mode 1 Key Update State Machine for initiating side

When the sending peers Key Update interval timer expires it creates a Key Update message (ciphertext id = 1) by doing the following:

1. increments the correlator
2. generates a new session key
3. generates the Key Padding

and sends the message, encrypted, to the receiving peer via the data transfer process. It then starts Tack, transitions to Key Update Pending substate and waits for a Key Update Ack from the peer. All data transfer frames continue to be sent encrypted using the OLD KEY.

When the peer receives the Key Update message it decrypts the message via the data transfer process, checks the ciphertext identifier (=1) and extracts the new key information. It then acknowledges receipt of the new key by creating a Key Update Acknowledge message (ciphertext id = 2) by doing the following

1. includes the correlator sent in the Key Update message
2. generates the key padding

and sends the Key Update Acknowledge, encrypted, to the originating peer via the data transfer process. It transitions to the Awaiting New Key sub state and continues to decrypt received data frames using the OLD KEY. If it cannot decrypt a frame with the old key, it attempts to decrypt with the new key. If the frame is successfully decrypted with the new key, the old key is discarded, the device transitions to the Operational state and the new key is used to decrypt all subsequent frames. If the frame is not successfully decrypted with the new key, the frame is discarded.

When the originating peer receives the Key Update Acknowledge, it checks the correlator against the one sent in the Key Update. If the correlator does not match the message is discarded. If the correlator does match, the originating peer stops Tack, transitions to Operational state and encrypts all subsequent data transfer frames are encrypted with the NEW KEY.

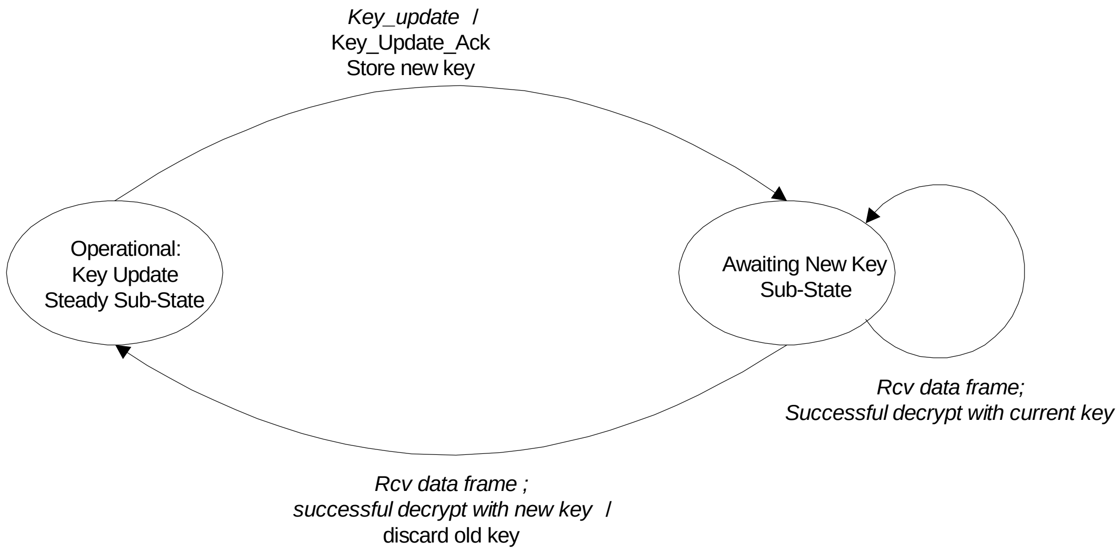


Figure 13 - Mode 1 Key Update State Machine for receiving side

5.2 MODE 2 SPECIFICATION

Mode 2 encryption support consists of the full negotiation procedures of RFC 1968. These procedures allow two peer frame relay devices to negotiate and converge on encryption methods and parameters to be used between them on a virtual connection. Different encryption methods may be neogtiated in each direction of the virtual connection. Encryptions methods that may be used with RFC 1968 are intended to be used with this protocol.

In general the control formats and procedures of RFC 1968 are used. The control formats of RFC1968 are encapsulated into the frame relay privacy structure defined in section 3.1 General Frame Format of this document. See section 5.2.1 Mode 2 Control Frame Formats below.

5.2.1 Mode 2 Control Frame Formats

This frame is used to negotiate Mode 2 parameters.

5.2.1.1 Mode 2 Control Message

Description				Octet
Frame Relay address, control and NLPID information				1-4
FRPP Header				5
ext 1	Spare	A 0	C/D 1	
FRPP Control Primitive (Note 1) Code				6
Identifier				7
Length (2 octets)				8 9
Configuration Options Type				10
Length				11
Values				12 n
FCS (2 octets)				n+1 n+2

Note 1: FRPP Control Primitive includes octets 6-n

Figure 14 - Mode 2 Control Frame

Field	Description
DLCI, control and NLPID	See section General Frame Format for details.
FRPP Header	▪ ext: extension bit = 1 ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) ▪ Authentication (A) bit = 0 ▪ Control / Data (C/D) bit 1 Control
FRPP Control Primitive Code	See RFC 1661 section 5 LCP Packet Formats and RFC 1968 section 3 Additional Packets. (values given in decimal)
Identifier:	See RFC 1661 section 5 LCP Packet Formats and RFC 1968 section 3 Additional Packets.
Length (2 octets)	See RFC 1661 section 5 LCP Packet Formats and RFC 1968 section 3 Additional Packets. Including: Code, Identifier, Length and all Configuration Options data
Configuration Options Type	See RFC 1968 section 4 ECP Configuration Options, 4.1 Proprietary Encryption OUI and 4.2 Publicly Available Encryption Types 254 (decimal)- is reserved and indicates FRPP Mode 1
Length	Length of Configuration option including Type, Length and Values fields.
Values	Zero or more octets, containing data as determined by the configuation options defined in RFC 1968 section 4.
FCS	Q.922 Frame Check Sequence

Table 11- Mode 2 Control Frame

5.2.2 Mode 2 Negotiation

Mode 2 of the Frame Relay Privacy Protocol encapsulates the same packet exhange mechanism as the PPP Encryption Control Protocol (RFC 1968) which is in turn modeled on the PPP Link Control Protocol (LCP) (RFC 1661). Mode 2 shall use the procedures described in sections 3.1 Reset-Request and Reset-Ack and 4.3 Negotiating an Encryption Algorithm of RFC 1968 using the frame formats described in section 5.2.1 Mode 2 Control Frame Formats of this document. The following exceptions apply to sections 3.1 and 4.3 of RFC 1968 and referenced sections of RFC 1661 (sections 4 The Option Negotiation Automaton):

If at any time a Mode 1 Config Request is received the receiving device will begin Mode 1 negotiation.

An entity may abandon Mode 2 and enter Mode 1 initialization phase at any time.

If an entity that supports Mode 2 is currently in Mode 1 and receives a Mode 2 Config Request, it shall begin Mode 2 negotiation.

Note: The (lower layer) Up/Down events for the automaton should be generated by the virtual connection status given by the PVC and SVC signalling protocols. Mode 2 packets received before this phase should be ignored.

Before any encrypted data is exchanged, the entity must reach the Opened state.

5.2.3 Mode 2 Data Transfer

This format is used for transferring ciphered data in Mode 2.

Description			**Octet**
Frame Relay address, control and NLPID information			1-4
FRPP Header			
ext 1	Spare	C/D 0	5
Encryption Payload			6-n
FCS			n+1
(2 octets)			n+2

Figure 15 - Mode 2 Data Transfer Frame Format

Field	Description
DLCI, control and NLPID	See section General Frame Format for details.
FRPP Header	▪ ext: extension bit ▪ spare: spare bits for future use set to 0 (message not rejected if set to 1) ▪ A bit: Authentication bit - Not applicable. ▪ C/D: Control / Data bit 0 Data
Encryption Payload	The payload octets are generated as dictated by the negotiated configuration options of RFC 1968.
FCS	Q.922 Frame Check Sequence

Table 12 - Mode 2 Data Transfer Frame Format

6 INTERACTION WITH OTHER PROTOCOLS

The following are examples of frame formats used with FRPP Mode 1 data transfer and other Frame Relay Forum implementation agreements. In general, the order of protocol application is multiprotocol encapsulate, compress, encrypt, and fragment.

Mode 2 data transfer formats are dependent on the type of encryption algorithm used.

6.1 MULTI-PROTOCOL ENCAPSULATION (FRF.3.1)

The following example shows an encrypted multiprotocol encapsulated frame (FRF.3.1). The FRF.3.1 frame starting with the control field, not including the FCS, is encrypted and becomes the payload of the FRPP frame.

Description			Octet
Q.922 Address			1
(2 octets)			2
Control (UI: 0x03)			3
NLPID (0xB3)			4
FRPP Header			
ext 1	Spare	C/D 0	5
Sequence number			6
Ciphertext Identifier (Note 1)			7
FRF 3.1 Information (octets 3- m)	**Control (UI: 0x03) (Note 1)**		**8**
	Optional Pad (Note 1)		**9**
	NLPID (Note 1)		**10**
	FRF.3.1 Data (Note 1)		**11 m**
Self Describing Padding (Note 1)			m+1 n
LCB			n+1
FCS			n+2
(2 octets)			n+3

Note 1: This field is encrypted.

Figure 16 - encrypted FRF.3.1 frame

6.2 DATA COMPRESSION OVER FRAME RELAY (FRF.9)

The following example shows an encrypted, compressed frame using FRF.9. The FRF.9 frame starting with the control field, not including the FCS, is encrypted and becomes the payload of the FRPP frame. For effective compression the frame must be compressed BEFORE it is encrypted. It is also recommended the FRPP negotiation be completed successfully before initiating the data compression negotiation.

Description			**Octet**
Q.922 Address			1
(2 octets)			2
Control (UI: 0x03)			3
NLPID (0xB3)			4
FRPP Header			
ext 1	Spare	C/D 0	5
Sequence number			6
Ciphertext Identifier (Note 1)			7
FRF 9 Information (octets 3- m)	**Control (UI: 0x03) (Note 1)**		**8**
	Optional Pad (Note 1)		**9**
	NLPID (Note 1)		**10**
	FRF.9 DCP Header and DCP Payload (Note 1)		**11 m**
Self Describing Padding (Note 1)			m+1 n
LCB			n+1
FCS (2 octets)			n+2 n+3

Note 1: This field is encrypted.

Figure 17 - encrypted FRF.9 frame

6.3 FRAGMENTATION (FRF.11 & FRF.12)

The following example shows an encrypted frame that is fragmented using FRF.12 end-to-end fragmentation. For efficient encryption, the frame is encrypted before it is fragmented. An encrypted frame is fragmented the same as any other frame. The encrypted frame, starting with the control field and not including the FCS, is fragmented and becomes the payload fragments of the FRF.12 frames. This is the same method used for FRF.11 Annex J. FRF.12 UNI/NNI fragmentation is done as specified in FRF.12.

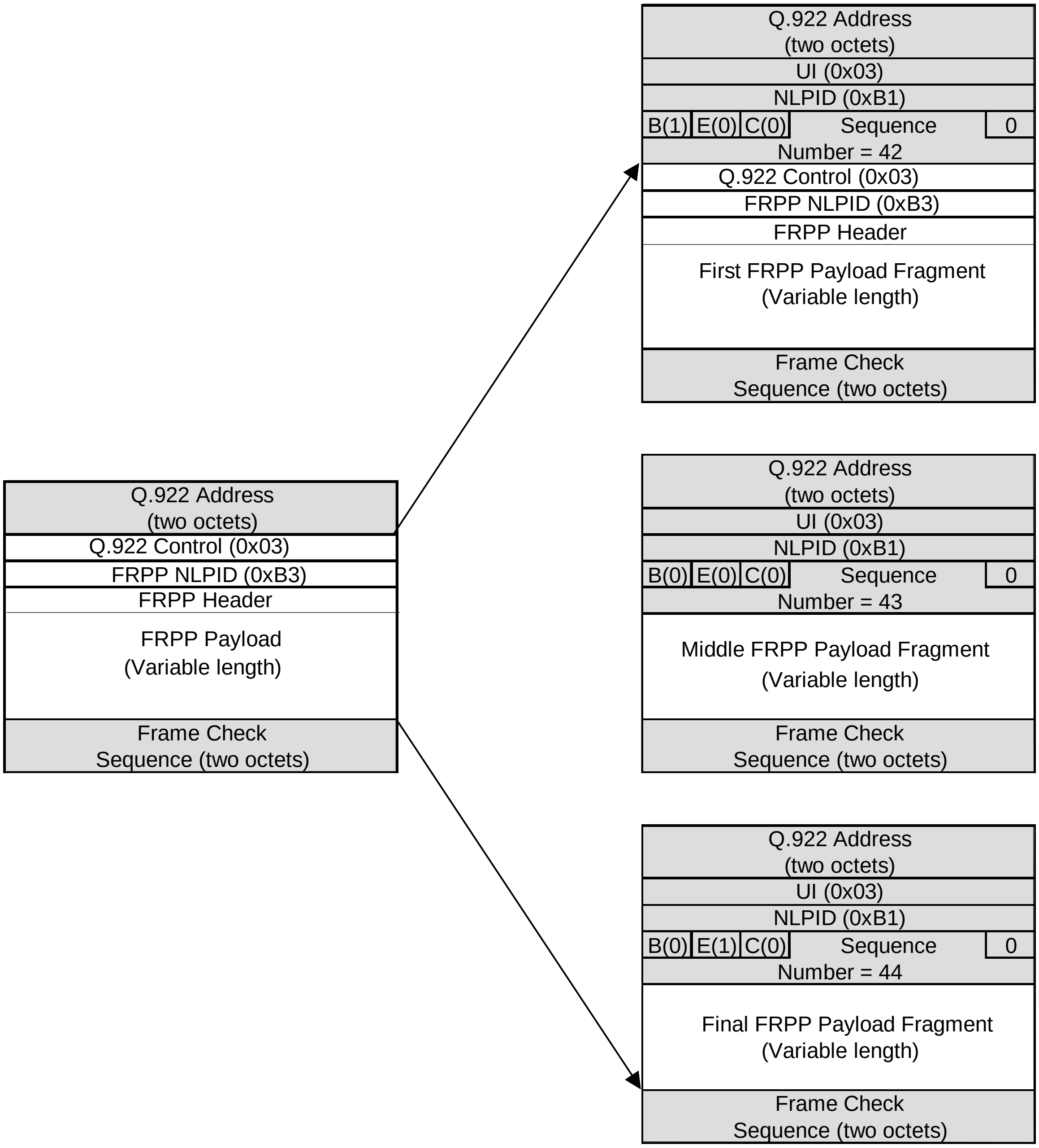


Figure 18 - encrypted frame fragmented with FRF.12 (end-to-end) and FRF.11 Annex J

For FRF.11, without using Annex J, a similar method would be used for data sent in the voice frame structure. The data would first be encrypted, and then fragmented according to the procedures in FRF.11. It would then be encapsulated in the voice frame structure described in FRF.11.