Network Working Group
Request for Comments: 2419
Obsoletes: 1969
Category: Standards Track

K. Sklower
University of California, Berkeley
G. Meyer
Shiva
September 1998

The PPP DES Encryption Protocol, Version 2 (DESE-bis)

Status of this Memo

Copyright Notice

Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method for transporting multi-protocol datagrams over point-to-point links.

The PPP Encryption Control Protocol (ECP) [2] provides a method to negotiate and utilize encryption protocols over PPP encapsulated links.

This document provides specific details for the use of the DES standard [5, 6] for encrypting PPP encapsulated packets.

Acknowledgements

The authors extend hearty thanks to Fred Baker of Cisco, Philip Rakity of Flowpoint, and William Simpson of Daydreamer for helpful improvements to the clarity and correctness of the document.

Table of Contents

1. Introduction

1.1. Motivation

The purpose of this memo is two-fold: to show how one specifies the necessary details of a "data" or "bearer" protocol given the context of the generic PPP Encryption Control Protocol, and also to provide at least one commonly-understood means of secure data transmission between PPP implementations.

The DES encryption algorithm is a well studied, understood and widely implemented encryption algorithm. The DES cipher was designed for efficient implementation in hardware, and consequently may be relatively expensive to implement in software. However, its pervasiveness makes it seem like a reasonable choice for a "model" encryption protocol.

Source code implementing DES in the "Electronic Code Book Mode" can be found in [7]. US export laws forbid the inclusion of compilation-ready source code in this document.

1.2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [8].

2. General Overview

The purpose of encrypting packets exchanged between two PPP implementations is to attempt to insure the privacy of communication conducted via the two implementations. The encryption process depends on the specification of an encryption algorithm and a shared

secret (usually involving at least a key) between the sender and receiver.

Generally, the encryptor will take a PPP packet including the protocol field, apply the chosen encryption algorithm, place the resulting cipher text (and in this specification, an explicit sequence number) in the information field of another PPP packet. The decryptor will apply the inverse algorithm and interpret the resulting plain text as if it were a PPP packet which had arrived directly on the interface.

The means by which the secret becomes known to both communicating elements is beyond the scope of this document; usually some form of manual configuration is involved. Implementations might make use of PPP authentication, or the EndPoint Identifier Option described in PPP Multilink [3], as factors in selecting the shared secret. If the secret can be deduced by analysis of the communication between the two parties, then no privacy is guaranteed.

While the US Data Encryption Standard (DES) algorithm [5, 6] provides multiple modes of use, this specification selects the use of only one mode in conjunction with the PPP Encryption Control Protocol (ECP): the Cipher Block Chaining (CBC) mode. In addition to the US Government publications cited above, the CBC mode is also discussed in [7], although no C source code is provided for it per se.

The initialization vector for this mode is deduced from an explicit 64-bit nonce, which is exchanged in the clear during the negotiation phase. The 56-bit key required by all DES modes is established as a shared secret between the implementations.

One reason for choosing the chaining mode is that it is generally thought to require more computation resources to deduce a 64 bit key used for DES encryption by analysis of the encrypted communication stream when chaining mode is used, compared with the situation where each block is encrypted separately with no chaining. Certainly, identical sequences of plaintext will produce different ciphers when chaining mode is in effect, thus complicating analysis.

However, if chaining is to extend beyond packet boundaries, both the sender and receiver must agree on the order the packets were encrypted. Thus, this specification provides for an explicit 16 bit sequence number to sequence decryption of the packets. This mode of operation even allows recovery from occasional packet loss; details are also given below.

3. Structure of This Specification

The PPP Encryption Control Protocol (ECP), provides a framework for negotiating parameters associated with encryption, such as choosing the algorithm. It specifies the assigned numbers to be used as PPP protocol numbers for the "data packets" to be carried as the associated "data protocol", and describes the state machine.

Thus, a specification for use in that matrix need only describe any additional configuration options required to specify a particular algorithm, and the process by which one encrypts/decrypts the information once the Opened state has been achieved.

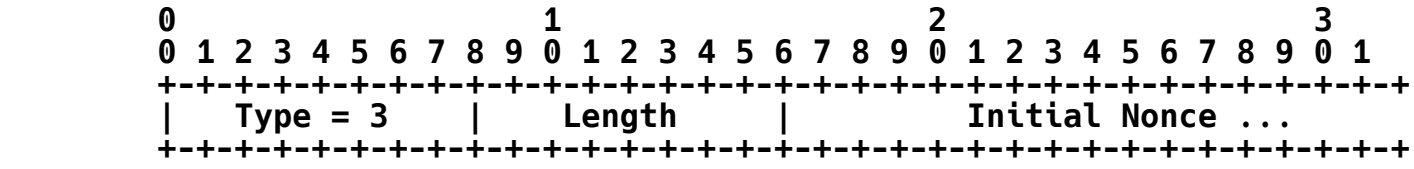
4. DESE Configuration Option for ECP

Description

The ECP DESE Configuration Option indicates that the issuing implementation is offering to employ this specification for decrypting communications on the link, and may be thought of as a request for its peer to encrypt packets in this manner.

The ECP DESE Configuration Option has the following fields, which are transmitted from left to right:

Figure 1: ECP DESE Configuration Option



Type

Type = 3, to indicate the DESE-bis protocol. The former value 1 indicating the previous DESE specification is deprecated, i.e. systems implementing this specification MUST NOT offer the former value 1 in a configure-request and MUST configure-reject the former value on receipt of a configure-request containing it.

Length

10

Initial Nonce

This field is an 8 byte quantity which is used by the peer implementation to encrypt the first packet transmitted after the sender reaches the opened state.

To guard against replay attacks, the implementation SHOULD offer a different value during each ECP negotiation. An example might be to use the number of seconds since Jan 1st, 1970 (GMT/UT) in the upper 32 bits, and the current number of nanoseconds relative to the last second mark in the lower 32 bits.

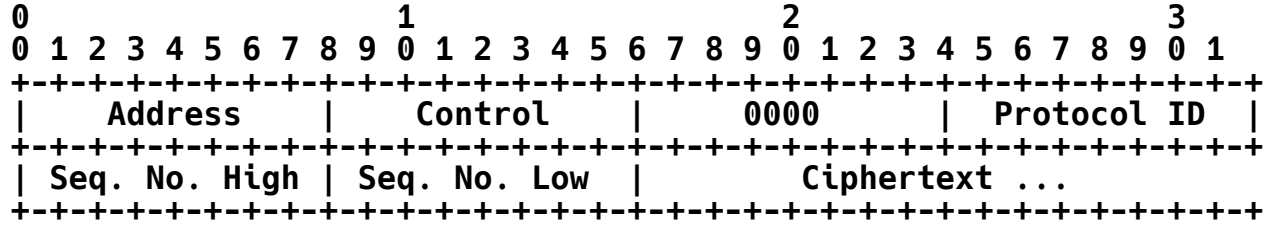
Its formulaic role is described in the Encryption section below.

5. Packet Format for DESE

Description

The DESE packets themselves have the following fields:

Figure 2: DES Encryption Protocol Packet Format



Address and Control

These fields MUST be present unless the PPP Address and Control Field Compression option (ACFC) has been negotiated.

Protocol ID

The value of this field is 0x53 or 0x55; the latter indicates that ciphertext includes headers for the Multilink Protocol, and REQUIRES that the Individual Link Encryption Control Protocol has reached the opened state. The leading zero MAY be absent if the PPP Protocol Field Compression option (PFC) has been negotiated.

Sequence Number

These 16-bit numbers are assigned by the encryptor sequentially starting with 0 (for the first packet transmitted once ECP has reached the opened state.

Ciphertext

The generation of this data is described in the next section.

6. Encryption

Once the ECP has reached the Opened state, the sender MUST NOT apply the encryption procedure to LCP packets nor ECP packets.

If the async control character map option has been negotiated on the link, the sender applies mapping after the encryption algorithm has been run.

The encryption algorithm is generally to pad the Protocol and Information fields of a PPP packet to some multiple of 8 bytes, and apply DES in Chaining Block Cipher mode with a 56-bit key K.

There are a lot of details concerning what constitutes the Protocol and Information fields, in the presence or non-presence of Multilink, and whether the ACFC and PFC options have been negotiated, and the sort of padding chosen.

Regardless of whether ACFC has been negotiated on the link, the sender applies the encryption procedure to only that portion of the packet excluding the address and control field.

If the Multilink Protocol has been negotiated and encryption is to be construed as being applied to each link separately, then the encryption procedure is to be applied to the (possibly extended) protocol and information fields of the packet in the Multilink Protocol.

If the Multilink Protocol has been negotiated and encryption is to be construed as being applied to the bundle, then the multilink procedure is to be applied to the resulting DESE packets.

6.1. Padding Considerations

Since the DES algorithm operates on blocks of 8 octets, plain text packets which are of length not a multiple of 8 octets must be padded. This can be injurious to the interpretation of some protocols which do not contain an explicit length field in their protocol headers.

Since there is no standard directory of protocols which are susceptible to corruption through padding, this can lead to confusion over which protocols should be protected against padding-induced corruption. Consequently, this specification requires that the unambiguous technique described below MUST be applied to ALL plain text packets.

The method of padding is based on that described for the LCP Self-Describing-Padding (SDP) option (as defined in RFC 1570 [4]), but differs in two respects: first, maximum-pad value is fixed to be 8, and second, the method is to be applied to ALL packets, not just "specifically identified protocols".

Plain text which is not a multiple of 8 octets long MUST be padded prior to encrypting the plain text with sufficient octets in the sequence of octets 1, 2, 3 ... 7 to make the plain text a multiple of 8 octets.

Plain text which is already a multiple of 8 octets may require padding with a further 8 octets (1, 2, 3 ... 8). These additional octets MUST be appended prior to encrypting the plain text if the last octet of the plain text has a value of 1 through 8, inclusive.

After the peer has decrypted the cipher text, it strips off the Self-Describing-Padding octets, to recreate the original plain text.

Note that after decrypting, only the content of the last octet need be examined to determine how many pad bytes should be removed. However, the peer SHOULD discard the frame if all the octets forming the padding do not match the scheme just described.

The padding operation described above is performed independently of whether or not the LCP Self-Describing-Padding (SDP) option has been negotiated. If it has, SDP would be applied to the packet as a whole after it had been ciphered and after the Encryption Protocol Identifiers had been prepended.

6.2. Generation of the Ciphertext

In this discussion, E[k] will denote the basic DES cipher determined by a 56-bit key k acting on 64 bit blocks. and D[k] will denote the corresponding decryption mechanism. The padded plaintext described in the previous section then becomes a sequence of 64 bit blocks P[i] (where i ranges from 1 to n). The circumflex character (^) represents the bit-wise exclusive-or operation applied to 64-bit blocks.

When encrypting the first packet to be transmitted in the opened state let C[0] be the result of applying E[k] to the Initial Nonce received in the peer's ECP DESE option; otherwise let C[0] be the final block of the previously transmitted packet.

The ciphertext for the packet is generated by the iterative process

$$C[i] = E[k](P[i] \text{ ^ } C[i-1])$$

for i running between 1 and n.

6.3. Retrieval of the Plaintext

When decrypting the first packet received in the opened state, let C[0] be the result of applying E[k] to the Initial Nonce transmitted in the ECP DESE option. The first packet will have sequence number zero. For subsequent packets, let C[0] be the final block of the previous packet in sequence space. Decryption is then accomplished by

$$P[i] = C[i-1] \text{ ^ } D[k](C[i]),$$

for i running between 1 and n.

6.4. Recovery after Packet Loss

Packet loss is detected when there is a discontinuity in the sequence numbers of consecutive packets. Suppose packet number N - 1 has an unrecoverable error or is otherwise lost, but packets N and N + 1 are received correctly.

Since the algorithm in the previous section requires C[0] for packet N to be C[last] for packet N - 1, it will be impossible to decode packet N. However, all packets N + 1 and following can be decoded in the usual way, since all that is required is the last block of ciphertext of the previous packet (in this case packet N, which WAS received).

7. MRU Considerations

Because padding can occur, and because there is an additional protocol field in effect, implementations should take into account the growth of the packets. As an example, if PFC had been negotiated, and if the MRU before had been exactly a multiple of 8, then the plaintext resulting combining a full sized data packets with a one byte protocol field would require an additional 7 bytes of padding, and the sequence number would be an additional 2 bytes so that the information field in the DESE protocol is now 10 bytes larger than that in the original packet. Because the convention is that PPP options are independent of each other, negotiation of DESE does not, by itself, automatically increase the MRU value.

8. Differences from RFC 1969

8.1. When to Pad

In RFC 1969, the method of Self-Describing Padding was not applied to all packets transmitted using DESE. Following the method of the SDP option itself, only "specifically identified protocols", were to be padded. Protocols with an explicit length identifier were exempt. (Examples included non-VJ-compressed IP, XNS, CLNP).

In this speficiation, the method is applied to ALL packets.

Secondly, this specification is clarified as being completely independent of the Self-Describing-Padding option for PPP, and fixes the maximum number of padding octets as 8.

8.2. Assigned Numbers

Since this specification could theoretically cause misinterpretation of a packet transmitted according to the previous specification, a new type field number has been assigned for the DESE-bis protocol

8.3. Minor Editorial Changes

This specification has been designated a standards track document. Some other language has been changed for greater clarity.

9. Security Considerations

This proposal is concerned with providing confidentiality solely. It does not describe any mechanisms for integrity, authentication or nonrepudiation. It does not guarantee that any message received has not been modified in transit through replay, cut-and-paste or active

tampering. It does not provide authentication of the source of any packet received, or protect against the sender of any packet denying its authorship.

This proposal relies on exterior and unspecified methods for authentication and retrieval of shared secrets. It proposes no new technology for privacy, but merely describes a convention for the application of the DES cipher to data transmission between PPP implementation.

Any methodology for the protection and retrieval of shared secrets, and any limitations of the DES cipher are relevant to the use described here.

10. References

[1] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[2] Meyer, G., "The PPP Encryption Protocol (ECP)", RFC 1968, June 1996.

[3] Sklower, K., Lloyd, B., McGregor, G., Carr, D., and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.

[4] Simpson, W., Editor, "PPP LCP Extensions", RFC 1570, January 1994.

[5] National Bureau of Standards, "Data Encryption Standard", FIPS PUB 46 (January 1977).

[6] National Bureau of Standards, "DES Modes of Operation", FIPS PUB 81 (December 1980).

[7] Schneier, B., "Applied Cryptography - Protocols Algorithms, and source code in C", John Wiley & Sons, Inc. 1994. There is an errata associated with the book, and people can get a copy by sending e-mail to schneier@counterpane.com.

[8] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11. Authors' Addresses

Keith Sklower
Computer Science Department
339 Soda Hall, Mail Stop 1776
University of California
Berkeley, CA 94720-1776

Phone: (510) 642-9587
EMail: sklower@CS.Berkeley.EDU

Gerry M. Meyer
Cisco Systems Ltd.
Bothwell House, Pochard Way,
Strathclyde Business Park,
Bellshill, ML4 3HB
Scotland, UK

Phone: (UK) (pending)
Fax: (UK) (pending)
Email: gemeyer@cisco.com

12. Full Copyright Statement