Network Working Group
Request for Comments: 1969
Category: Informational

K. Sklower
University of California, Berkeley
G. Meyer
Spider Systems

June 1996

The PPP DES Encryption Protocol (DESE)

Status of This Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

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 for helpful improvements to the clarity 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 following language conventions are used in the items of specification in this document:

- MUST, SHALL or MANDATORY -- the item is an absolute requirement of the specification.
- SHOULD or RECOMMENDED -- the item should generally be followed for all but exceptional circumstances.
- MAY or OPTIONAL -- the item is truly optional and may be followed or ignored according to the needs of the implementor.

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 Protol (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. Further, if chaining is not used, even if the key is never deduced, the communication may be subject to replay attacks.

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.

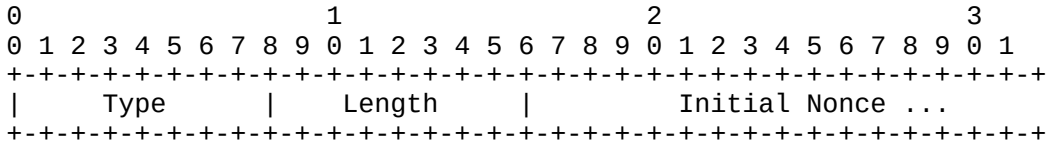
4. DESE Configuration Option for ECP

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

1, to indicate the DESE protocol.

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.

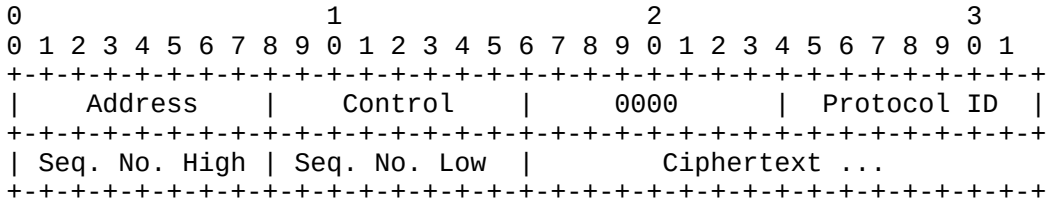
Its formulaic role is described in the Encryption section below.

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, 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. (Additional padding of the ciphered packet for the purposes of transmission by HDLC hardware which requires an even number of bytes should not be necessary since the information field will now be of length a multiple of 8, and whether or not the packet is of even length can be forced by use or absence of a leading zero in the

protocol field).

For protocols which do have an explicit length field, such as IP, IPX, XNS, and CLNP, then padding may be accomplished by adding random trailing garbage. Even when performing the Multilink protocol, if it is only being applied to packets with explicit length fields, and if care is taken so that all non-terminating fragments (i.e., those not bearing the (E)nd bit) are of lengths divisible by 8; then no ill effects will happen if garbage padding is applied only to terminating fragments.

For certain cases, such as the PPP bridging protocol when the trailing CRC is forwarded or when any bridging is being applied to protocols not having explicit length fields, adding garbage changes the interpretation of the packet. The self-describing padding option [4] permits unambiguous removal of padded bytes; although it should only be used when absolutely necessary as it may inadvertently require adding as many as 8 octets to packets that could otherwise be left unaltered.

> Consider a packet, which by unlucky circumstance is already a multiple of 8 octets, but terminates in the sequence 0x1, 0x2. Self-describing padding would otherwise remove the trailing two bytes. For purposes of coexistence with archaic HDLC chips where it is necessary to transmit packets of even length, one would normally only have to add an additional two octets (0x1, 0x2), which could then be removed. However, since the packet was initially a multiple of 8 bytes, an additional 8 bytes would need to be added.

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

$$C[i] = E[k](P[i] \wedge C[i-1])$$

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

$$P[i] = C[i-1] \wedge D[k](C[i]),$$

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. Security Considerations

Security issues are the primary subject of this memo. 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.

9. References

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

[2] Meyer, G., "The PPP Encryption Protocol", RFC 1968, Spider Systems, June 1996.

[3] Sklower, K., Lloyd, B., McGregor, G., and D. Carr, "The PPP Multilink Protocol (MP)", RFC 1717, UC Berkeley, November 1994.

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

10. Authors' Addresses

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

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

Gerry M. Meyer
Spider Systems
Stanwell Street
Edinburgh EH6 5NG
Scotland, UK

Phone: (UK) 131 554 9424
Fax: (UK) 131 554 0649
EMail: gerry@spider.co.uk