             The PPP Triple-DES Encryption Protocol (3DESE)

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 Triple-DES
   standard (3DES) [6] for encrypting PPP encapsulated packets.

Table of Contents

1. Introduction

   The purpose of encrypting packets exchanged between two PPP
   implementations is to attempt to insure the privacy of communication
   conducted via the two implementations. There exists a large variety
   of encryption algorithms, where one is the DES algorithm. The DES
   encryption algorithm is a well studied, understood and widely
   implemented encryption algorithm. Triple-DES means that this
   algorithm is applied three times on the data to be encrypted before
   it is sent over the line. The variant used is the DES-EDE3-CBC, which
   is described in more detail in the text. It was also chosen to be
   applied in the security section of IP [5].

   This document shows how to send via the Triple-DES algorithm
   encrypted packets over a point-to-point-link. It lies in the context
   of the generic PPP Encryption Control Protocol [2].

   Because of the use of the CBC-mode a sequence number is provided to
   ensure the right order of transmitted packets. So lost packets can be
   detected.

   The padding section reflects the result of the discussion on this
   topic on the ppp mailing list.

   In this document, the key words "MUST", "SHOULD", and "recommended"
   are to be interpreted as described in [3].

1.1 Algorithm

   The DES-EDE3-CBC algorithm is a simple variant of the DES-CBC
   algorithm. In DES-EDE3-CBC, an Initialization Vector (IV) is XOR'd
   with the first 64-bit (8 octet) plaintext block (P1). The keyed DES
   function is iterated three times, an encryption (E) followed by a
   decryption (D) followed by an encryption (E), 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 8-octet plaintext block (Pi). The keyed DES-EDE3
   encryption function generates the ciphertext (Ci) for that block.

```
                P1                 P2                 Pi
                |                  |                  |
         IV--->(X)     +------>(X)        +-------->(X)
                v      |        v         |          v
             +-----+   |     +-----+      |       +-----+
         k1->|  E  |   | k1->|  E  |      :   k1->|  E  |
             +-----+   |     +-----+      :       +-----+
                |      |        |         :          |
                v      |        v         :          v
             +-----+   ^     +-----+      ^       +-----+
         k2->|  D  |   |  k2->|  D  |     |   k2->|  D  |
             +-----+   |     +-----+      |       +-----+
                |      |        |         |          |
                v      |        v         |          v
             +-----+   |     +-----+      |       +-----+
         k3->|  E  |   | k3->|  E  |      |   k3->|  E  |
             +-----+   |     +-----+      |       +-----+
                |      |        |         |          |
             +---->+       +------>+          +---->
                |                  |          |
                C1                 C2         Ci
```
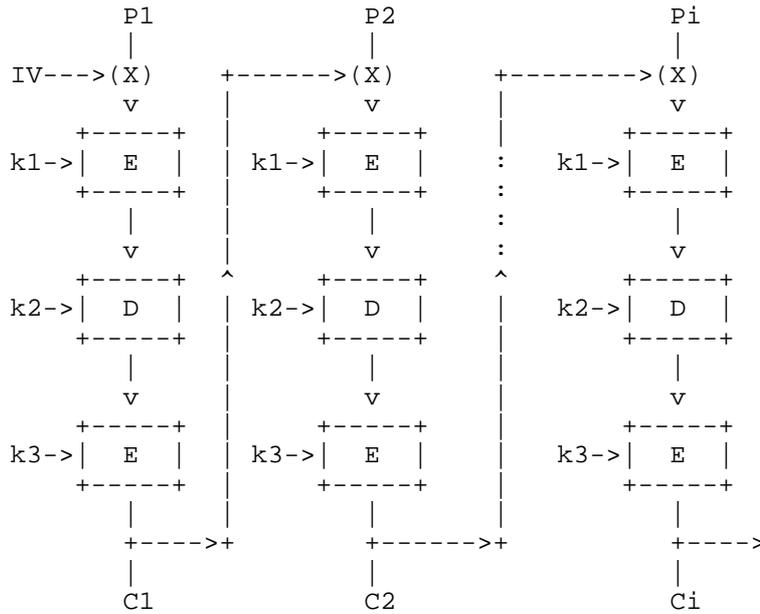
   To decrypt, the order of the functions is reversed: decrypt with k3,
   encrypt with k2, decrypt with k1, and XOR with the previous cipher-
   text block.

   When all three keys (k1, k2 and k3) are the same, DES-EDE3-CBC is
   equivalent to DES-CBC.

1.2 Keys

   The secret DES-EDE3 key shared between the communicating parties is
   effectively 168-bits long. This key consists of three independent
   56-bit quantities used by the DES algorithm. Each of the three 56-
   bit subkeys is stored as a 64-bit (8 octet) quantity, with the least
   significant bit of each octet used as a parity bit.

   When configuring keys for 3DESE those with incorrect parity or so-
   called weak keys [6] SHOULD be rejected.
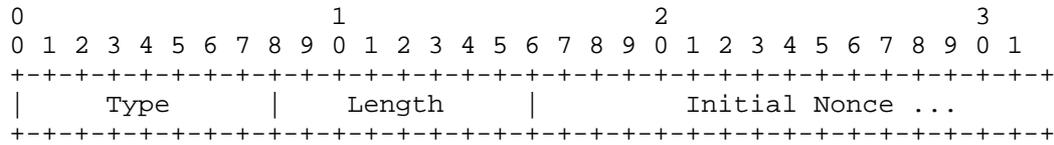
2. 3DESE Configuration Option for ECP

   Description

        The ECP 3DESE 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 3DESE Configuration Option has the following fields, which
are transmitted from left to right:

Figure 1: ECP 3DESE Configuration Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |        Initial Nonce ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

     2, to indicate the 3DESE protocol.

Length

     10
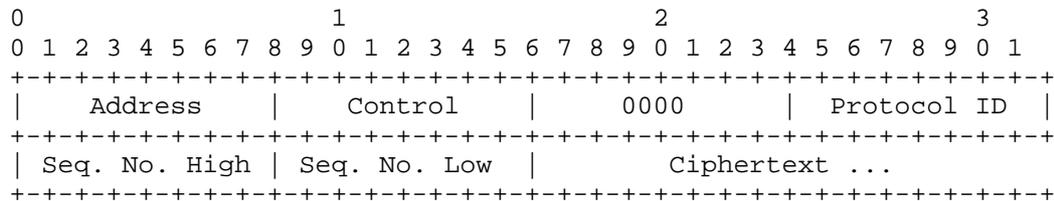
Initial Nonce

          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.

3. Packet format for 3DESE

   Description

          The 3DESE packets that contain the encrypted payload have the
          following fields:

          Figure 2: 3DESE Encryption Protocol Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Address    |    Control    |     0000      |  Protocol ID  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Seq. No. High | Seq. No. Low  |        Ciphertext ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   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 the use of the Individual Link Encryption
Control Protocol and that the ciphertext contains a
Multilink fragment. Protocol Field Compression MAY be
applied to the leading zero if 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.

4. 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 3DES as described in section 1.1 with the three 56-bit keys k1,
k2 and k3.

The encryption procedure is only applied to that portion of the
packet excluding the address and control field.

When encrypting the first packet after ECP stepped into opened state
the Initial Nonce is encrypted via 3DES-algorithm before its use.

4.1 Padding

Since the 3DES algorithm operates on blocks of 8 octets, plain text
packets which are of length not a multiple of 8 octets must be padded
prior to encrypting. If this padding is not removed after decryption
this can be injurious to the interpretation of some protocols which
do not contain an explicit length field in their protocol headers.

Therefore all packets not already a multiple of eight bytes in length
MUST be padded prior to encrypting using the unambiguous technique of
Self Describing Padding with a Maximum Pad Value (MPV) of 8. This
means that the plain text is padded with the sequence of octets 1, 2,
3, .. 7 since its length is a multiple of 8 octets. Negotiation of
SDP is not needed. Negotiation of the LCP Self Describing Option may
be negotiated independently to solve an orthogonal problem.

Plain text which length is already a multiple of 8 octets may require
padding with a further 8 octets (1, 2, 3 ... 8). These additional
octets MUST only be appended, if the last octet of the plain text had
a value of 8 or less.

When using Multilink and encrypting on individual links it is
recommended that all non-terminating fragments have a length
divisible by 8. So no additional padding is needed on those
fragments.

After the peer has decrypted the ciphertext, it strips off the Self
Describing Padding octets to recreate the original plain text. The
peer SHOULD discard the frame if the octets forming the padding do
not match the Self Describing Padding scheme just described.

Note that after decrypting, only the content of the last byte needs
to be examined to determine the presence or absence of a Self
Described Pad.

4.2 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 previously described algorithm requires the last C_i of
packet N - 1 to decrypt C1 of packet N, it will be impossible to
decrypt packet N. However, all packets N + 1 and following can be
decrypted 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).

5. 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.

Security issues are the primary subject of this memo. This proposal
relies on exterior and unspecified methods for retrieval of shared
secrets. It proposes no new technology for privacy, but merely
describes a convention for the application of the 3DES cipher to data
transmission between PPP implementations. Any methodology for the
protection and retrieval of shared secrets, and any limitations of
the 3DES cipher are relevant to the use described here.

6. References

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

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

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

   [4] Sklower, K., and G. Meyer, "The PPP DES Encryption Protocol,
        Version 2 (DESE-bis)", RFC 2419, September 1998.

   [5] Doraswamy, N., Metzger, P., Simpson, W., "The ESP Triple DES
        Transform", Work in Progress, June 1997.

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

7. Acknowledgements

   Many portions of this document were taken from [4] and [5]. Bill
   Simpson gave useful hints on the initial revision.

8. Author's Address

   Holger Kummert
   Nentec Gesellschaft fuer Netzwerktechnologie
   76227 Karlsruhe, Killisfeldstr. 64, Germany

   Phone: +49 721 9495 0
   EMail: kummert@nentec.de

9. Full Copyright Statement

   Copyright (C) The Internet Society (1998). All Rights Reserved.

   This document and translations of it may be copied and furnished to
   others, and derivative works that comment on or otherwise explain it
   or assist in its implementation may be prepared, copied, published
   and distributed, in whole or in part, without restriction of any
   kind, provided that the above copyright notice and this paragraph are
   included on all such copies and derivative works. However, this
   document itself may not be modified in any way, such as by removing
   the copyright notice or references to the Internet Society or other
   Internet organizations, except as needed for the purpose of
   developing Internet standards in which case the procedures for
   copyrights defined in the Internet Standards process must be
   followed, or as required to translate it into languages other than
   English.

   The limited permissions granted above are perpetual and will not be
   revoked by the Internet Society or its successors or assigns.

   This document and the information contained herein is provided on an
   "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
   TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
   BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
   HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
   MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.