Network Working Group
Request for Comments: 1968
Category: Standards Track

G. Meyer
Spider Systems

June 1996

The PPP Encryption Control Protocol (ECP)

Status of this Memo

Abstract

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

This document defines a method for negotiating data encryption over PPP links.

Conventions

The following language conventions are used in the items of specification in this document:

- MUST -- the item is an absolute requirement of the specification. MUST is only used where it is actually required for interoperation, not to try to impose a particular method on implementors where not required for interoperability.

- SHOULD -- the item should 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.

  The words "should" and "may" are also used, in lower case, in their more ordinary senses.

Table of Contents

1. Introduction

In order to establish communications over a PPP link, each end of the link must first send LCP packets to configure and test the data link during Link Establishment phase. After the link has been established, optional facilities may be negotiated as needed.

One such facility is data encryption. A wide variety of encryption methods may be negotiated, although typically only one method is used in each direction of the link.

A different encryption algorithm may be negotiated in each direction, for speed, cost, memory or other considerations.

2. Encryption Control Protocol (ECP)

The Encryption Control Protocol (ECP) is responsible for configuring and enabling data encryption algorithms on both ends of the point-to-point link.

ECP uses the same packet exchange mechanism as the Link Control Protocol (LCP). ECP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. ECP packets received before this phase is reached should be silently discarded.

The Encryption Control Protocol is exactly the same as LCP [1] with the following exceptions:

Frame Modifications

The packet may utilise any modifications to the basic frame format which have been negotiated during the Link Establishment phase.

Data Link Layer Protocol Field

Exactly one ECP packet is encapsulated in the PPP Information field, where the PPP Protocol field indicates type hex 8053 (Encryption Control Protocol).

When individual link data encryption is used in a multiple link connection to a single destination [2], the PPP Protocol field indicates type hex 8055 (Individual link Encryption Control Protocol).

Code field

ECP uses (decimal) codes 1 through 7 (Configure-Request, Configure-Ack, Configure-Nak, Configure-Reject, Terminate-Request, Terminate-Ack and Code-Reject); And may also use code 14 (Reset-Request) and code 15 (Reset-Ack). Other codes should be treated as unrecognised and should result in Code-Rejects.

Negotiation

ECP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. An implementation should be prepared to wait for Authentication and Link Quality Determination to finish before timing out waiting for a Configure-Ack or other response.

An implementation MUST NOT transmit data until ECP negotiation has completed successfully. If ECP negotiation is not successful the link SHOULD be brought down.

Configuration Option Types

ECP has a distinct set of Configuration Options.

2.1 Sending Encrypted Datagrams

Before any encrypted packets may be communicated, PPP must reach the Network-Layer Protocol phase, and the Encryption Control Protocol must reach the Opened state.

An encrypted packet is encapsulated in the PPP Information field, where the PPP Protocol field indicates type hex 0053 (Encrypted datagram).

When using multiple PPP links to a single destination [2], there are two methods of employing data encryption:

- The first method is to encrypt the data prior to sending it out through the multiple links.

  The PPP Protocol field MUST indicate type hex 0053.

- The second is to treat each link as a separate connection, that may or may not have encryption enabled.

  On links which have negotiated encryption, the PPP Protocol field MUST be type hex 0055 (Individual link encrypted datagram).

Only one encryption algorithm in each direction is in use at a time, and that is negotiated prior to sending the first encrypted frame. The PPP Protocol field of the encrypted datagram indicates that the frame is encrypted, but not the algorithm with which it was encrypted.

The maximum length of an encrypted packet transmitted over a PPP link is the same as the maximum length of the Information field of a PPP encapsulated packet. If the encryption algorithm is likely to increase the size of the message beyond that, multilink should also be negotiated to allow fragmentation of the frames (even if only using a single link).

If the encryption algorithm carries history between frames, the encryption algorithm must supply a way of determining if it is passing data reliably, or it must require the use of a reliable transport such as LAPB [3].

Compression may also be negotiated using the Compression Control Protocol [5]. To ensure interoperability, plain text MUST be:

- First compressed.

- Then encrypted.

This order has been chosen since it should result in smaller output and more secure encryption.

3. Additional Packets

The Packet format and basic facilities are already defined for LCP [1].

Up-to-date values of the ECP Code field are specified in the most recent "Assigned Numbers" RFC [4]. This specification concerns the following values:

14 Reset-Request
15 Reset-Ack

3.1 Reset-Request and Reset-Ack

Description

ECP includes Reset-Request and Reset-Ack Codes in order to provide a mechanism for indicating a decryption failure in one direction of a decrypted link without affecting traffic in the other direction. Some encryption algorithms may not require this mechanism.

Individual algorithms need to specify a mechanism for determining how to detect a decryption failure. On initial detection of a decryption failure, an ECP implementation SHOULD transmit an ECP packet with the Code field set to 14 (Reset-Request). The Data field may be filled with any desired data.

Once a Reset-Request has been sent, any encrypted packets received are discarded. Further Reset-Requests MAY be sent with the same Identifier, until a valid Reset-Ack is received.

When the link is busy, one decryption error is usually followed by several more before the Reset-Ack can be received. It is undesirable to transmit Reset-Requests more frequently than the round-trip-time of the link, since this will result in redundant Reset-Requests and Reset-Acks being transmitted and processed. The receiver MAY elect to limit transmission of Reset-Requests (to say one per second) while a Reset-Ack is outstanding.

Upon reception of a Reset-Request, the transmitting encrypter is reset to an initial state. An ECP packet MUST be transmitted with the Code field set to 15 (Reset-Ack), the Identifier field copied from the Reset-Request packet, and the Data field filled with any desired data.

On receipt of a Reset-Ack, the receiving decrypter is reset to an initial state. Since there may be several Reset-Acks in the pipe, the decrypter MUST be reset for each Reset-Ack which matches the currently expected identifier.

A summary of the Reset-Request and Reset-Ack packet formats is shown below. The fields are transmitted from left to right.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.