PPP Protocol Modules for TTCN-3 Toolset with TITAN, Function Specification

Contents

[1 Introduction 2](#_Toc365900051)

[1.1 Revision History 2](#_Toc365900052)

[1.2 How to Read this Document 2](#_Toc365900053)

[1.3 Scope 2](#_Toc365900054)

[1.4 References 2](#_Toc365900055)

[1.5 Abbreviations 3](#_Toc365900056)

[1.6 Terminology 3](#_Toc365900057)

[2 General 3](#_Toc365900058)

[3 Functional Specification 4](#_Toc365900059)

[3.1 Protocol Version Implemented 4](#_Toc365900060)

[3.2 Modifications/deviations Related to the Protocol Specification 4](#_Toc365900061)

[3.2.1 Implemented messages 4](#_Toc365900062)

[3.2.2 Protocol Modifications/Deviations 4](#_Toc365900063)

[3.3 Encoding/Decoding and Other Related Functions 4](#_Toc365900064)

# Introduction

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2008-06-06 | PA1 | First draft version | ETHEKR |
| 2010-06-01 | PB1 | EAP, IP has been added | ETMEMOD |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## How to Read this Document

This is the Function Specification for the set of PPP protocol modules. PPP protocol modules are developed for the TTCN-3 Toolset with TITAN. This document should be read together with the Product Revision Information [3].

## Scope

The purpose of this document is to specify the content of the PPP protocol modules.

## References

1. 2/198 17-CRL 113 200/3 Uen  
   Programmer’s Technical Reference for the TITAN TTCN-3 Test Executor
2. ETSI ES 201 873-1 v.3.2.1 (2007-02)  
   The Testing and Test Control Notation version 3. Part 1: Core Language
3. 109 21-CNL 113 599-1 Uen  
   PPP Protocol Modules for TTCN-3 Toolset with TITAN, Product Revision Information
4. IETF RFC 1661  
   The Point-to-Point Protocol
5. IETF RFC 1332  
   The PPP Internet Protocol Control Protocol (IPCP)
6. IETF RFC 1877  
   PPP Internet Protocol Control Protocol Extensions for   
   Name Server Address
7. IETF RFC 1994  
   PPP Challenge Handshake Authentication Protocol (CHAP)
8. IETF RFC 1334  
   PPP Authentication Protocols
9. IETF RFC 1662  
   PPP in HDLC-like Framing
10. IETF RFC 3748  
    Extensible Authentication Protocol (EAP)
11. Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM)  
    draft-haverinen-pppext-eap-sim-16.txt  
    (2004-12)
12. Extensible Authentication Protocol Method for 3rd Generation  
    Authentication and Key Agreement (EAP-AKA)  
    draft-arkko-pppext-eap-aka-15.txt  
    (2004-12)

## Abbreviations

CHAP PPP Challenge Handshake Authentication Protocol

IETF Internet Engineering Task Force

IP Internet Protocol

IPCP PPP Internet Protocol Control Protocol

PAP PPP Authentication Protocols

PPP Point-to-Point Protocol

EAP Extensible Authentication Protocol

RFC Request for Comments

TTCN-3 Testing and Test Control Notation version 3

## Terminology

TITAN TTCN-3 Test Executor.

# General

Protocol modules implement the message structures of the related protocol in a formalized way, using the standard specification language TTCN-3. This allows defining of test data (templates) in the TTCN-3 language and correctly encoding/decoding messages when executing test suites using the TITAN TTCN-3 test environment.

Protocol modules are using TITAN’s RAW encoding attributes [1] and hence are usable with the TITAN test toolset only.

# Functional Specification

## Protocol Version Implemented

This protocol module contains the protocol messages and elements of PPP [4] and its associated protocols IP [5], IPCP [5], PPP Extensions for   
Name Server Address [6], CHAP [7], PAP [8], EAP [10-12] and the Address and Control fields are defined in [9].

## Modifications/deviations Related to the Protocol Specification

### Implemented messages

All message types listed in protocol descriptions are implemented.

### Protocol Modifications/Deviations

None

## Encoding/Decoding and Other Related Functions

This product also contains encoding/decoding functions that provide for the correct encoding of messages when sent from TITAN and correct decoding of messages when received by TITAN. Implemented encoding/decoding functions:

Name Type of formal parameters Type of return value  
**enc\_PDU\_PPP PDU\_PPP octetstring  
dec\_PDU\_PPP octetstring PDU\_PPP**

Note that the Address and Control fields defined in [9] are treated as a single optional field in the beginning of PDU\_PPP.

Implemented PPP EAP functions (useful in RADIUS Protocol Module Generator):

Name Type of parameters Type of return value

**f\_enc\_PDU\_EAP PDU\_EAP octetstring**

**f\_dec\_PDU\_EAP octetstring PDU\_EAP**

**f\_enc\_PDU\_EAP\_list PDU\_EAP\_list octetstring**

**f\_dec\_PDU\_EAP\_list octetstring PDU\_EAP\_list**

**f\_enc\_eap\_sim\_attrib\_list eap\_sim\_attrib\_list octetstring**

**f\_dec\_eap\_sim\_attrib\_list octetstring eap\_sim\_attrib\_list**

**f\_enc\_eap\_aka\_attrib\_list eap\_aka\_attrib\_list octetstring**

**f\_dec\_eap\_aka\_attrib\_list octetstring eap\_aka\_attrib\_list**

**f\_calc\_HMAC octetstring, octetstring, integer octetstring**

**f\_initEAPPortDescriptor EAP\_port\_descriptor** (inout)

**f\_get\_EAP\_parameters octetstring** (inout)**,   
 EAP\_port\_descriptor** (inout)**, Boolean**

**f\_set\_Ki integer, octetstring,   
 EAP\_port\_descriptor** (inout)

**f\_set\_K integer, octetstring,   
 EAP\_port\_descriptor** (inout)

**f\_set\_SQN integer, octetstring,  
 EAP\_port\_descriptor** (inout)

**f\_set\_SQN\_MS integer, octetstring,   
 EAP\_port\_descriptor** (inout)

**f\_set\_AMF integer, octetstring,   
 EAP\_port\_descriptor** (inout)

**f\_calc\_AKA\_Keys octetstring, octetstring, octetstring  
 octetstring, octetstring** (inout)**,  
 octetstring** (inout)**, octetstring** (inout)

**f\_calc\_A3A8 octetstring, octetstring octetstring**

**f\_calc\_SRES octetstring, octetstring octetstring**

**f\_calc\_Kaut octetstring, octetstring octetstring**

**f\_encrypt\_at\_encr octetstring, octetstring  
 octetstring, boolean octetstring**

**f\_crypt\_atSimEncrData at\_sim\_encr\_data  
 octetstring, octetstring,  
 boolean at\_sim\_encr\_data**

**f\_crypt\_atAKAEncrData at\_aka\_encr\_data  
 octetstring, octetstring,  
 boolean at\_aka\_encr\_data**