Network Working Group
Internet Draft
expires in six months

William A. Nace(NSA)
James E. Zmuda(SPYRUS)
November 21st, 1997

PPP EAP DSS Public Key Authentication Protocol
<draft-ietf-pppext-eapdss-01.txt>

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, which allows negotiation of an Authentication Protocol for authentication of its peer before allowing Network Layer protocols to transmit over the link.

PPP Extensible Authentication Protocol (EAP) [2] provides for a

number of authentication mechanisms. This document specifies yet another authentication mechanism that may be used within the EAP framework. This document defines the DSS Public Key Authentication Protocol within PPP EAP.

1. Introduction

In order to establish communications over a point-to- point link, each end of the PPP link must first send LCP packets to configure the data link during Link Establishment phase. After the link has been established, PPP provides for an optional Authentication phase before proceeding to the Network- Layer Protocol phase.

By default, authentication is not mandatory. If authentication of the link is desired, an implementation MUST specify the Authentication-Protocol Configuration Option during Link Establishment phase.

PPP Extensible Authentication Protocol (EAP) [2] allows for a number of authentication protocols including DSS Public Key Authentication Protocol.

This document defines the PPP EAP DSS Public Key Authentication Protocol. The Link Establishment and Authentication phases, and the Authentication-Protocol Configuration Option are defined in The Point-to-Point Protocol (PPP) [1]. The Extensible Authentication protocol is defined in PPP Extensible Authentication Protocol (EAP) [2].

1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

MUST — This word, or the adjective required, means that the definition is an absolute requirement of the specification.

MUST NOT — This phrase means that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective recommended, means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighed before choosing a different course.

MAY This word, or the adjective optional, means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

1.2. Terminology

This document frequently uses the following terms:

authenticator The end of the link requiring the authentication. The authenticator specifies use of DSS Authentication in the EAP-Request during Authentication phase.

certificate A certificate consists of the binding together of one or more public key values and an identity. This binding is effected through a digital signature which is computed over the data containing both the public key and the identity. This signature is applied by a "certification authority" who is recognized as issuing this certificate on behalf of the entity identified in the certificate. In this manner a recipient of this certificate can determine the recognized public key of the particular entity identified in the certificate. This requires the recipient to, either directly or indirectly, trust the authority who has issued this certificate.

certification authority (CA)
An authority trusted by one or more users to create and assign certificates. [3].

digital signature
In the DSS, a digital signature is produced by performing the DSA signing operation with a private key

on the SHA-1 Hash value computed over the original data to be signed. The verification of this digital signature requires the verifier to obtain the original message, and the signature value, and the proper public key value that is associated with the signer (see certificates below). The verifier then also computes the SHA-1 Hash of the message data, and then perform a computation whose inputs include this hash value, the public key, and the signature value. If the output of this computation matches a particular part of the signature value produced by the signer, then the signature is verified.

DSA Digital Signature Algorithm

DSS Digital Signature Standard

DSS key pair A pair of keys, one of which, the private key, can be used to produce a "signature". The other, or public, key can be used only to verify that a digital signature has been produced by the private key it is associated with, when acting on a particular piece of data. Under the DSA these two keys do not form an encryption/decryption pair, however.

distinguished name
A unique heirarchical name. Used in the certificate's "subject" field to denote the entity associated with the public key value(s) in the certificate[2]. Also used in the certificate's "issuer" field to denote the entity that issued this certificate.

peer The other end of the point-to-point link; the end which is being authenticated by the authenticator.

private key That key of a key pair which is known only by that user [3].

public key That key of a key pair which is publicly known [3].

SHA-1 Secure Hash Algorithm revision one.

2. PPP EAP DSS Public Key Authentication

The PPP Extensible Authentication Protocol is a general protocol for PPP authentication which supports multiple authentication mechanisms. EAP MAY be negotiated at Link Control Phase. EAP MAY then be used to select the DSS Public Key Authentication mechanisms at the Authentication Phase.

The DSS Public Key Authentication Protocol is a challenge- response protocol based on unilateral two pass authentication as described in NIST FIPS PUB 196 "Standard for Public Key Cryptographic Entity Authentication Mechanisms" [4]. The authenticator issues a challenge in the form of a Request packet. The peer MUST formulate a Response packet based on information in the Request packet as well as information only the peer knows (the peer's private key). The peer MUST also provide in its response a reference (i.e. the subject Distinguished Name in the Certificate) to its own certificate (the certificate containing the peer's public key), as well as proof that it

knows the corresponding private key. The peer's certificate is assumed to have been obtained through other means. One such means is the use of the Certificate Exchange Protocol. The Certificate Exchange Protocol is defined as an extension to the PPP protocol suite. It is suggested as occurring during a new phase in between Link Control and Authentication. The Certificate Exchange Protocol is defined in [5].

In detail, the steps in EAP DSS are:

1. After the Link Establishment phase is complete and Extensible Authentication Protocol is negotiated, the authenticator sends a Request packet to authenticate the peer. The Request packet has a type field specifying DSS Public Key Authentication plus some random data produced by the authenticator.

2. The peer sends a Response packet in reply to the Request. The response contains the digital signature computed by the peer over the concatenation of the challenge, the timestamp, and the peer's distinguished name.

3. Based on information contained in the Response packet, the authenticator ends the authentication phase with either a Success packet or a Failure packet. These packets are defined in PPP Extensible Authentication Protocol (EAP) [2].

3. PPP EAP DSS Public Key Authentication Packet Format

DSS Unilateral authentication is performed using a derivative of the FIPS PUB 196 mechanism as defined below. The FIPS PUB 196 verifier corresponds to the EAP authenticator, while the claimant has a similar relation to the EAP authenticatee.

In keeping with FIPS PUB 196 notation, the authenticator is identified as "B"; the authenticatee as "A". Two packets are exchanged in order to perform the authentication, first from B to A, and then from A to B.

Both the EAP Response and Request packets for the DSS Unilateral Type have the following 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
  Public Key Authentication ProtocolNovember 1997
```

NOT imply that we will use ASN.1 to represent the contents of TokenBA or TokenAB in the EAP DSS Request and Response packets. This is rather just a list of the information found in the EAP DSS packets.]

Token BA1 is profiled from FIPS PUB 196 Appendix A as:

```
TokenBA ::= SEQUENCE {
    ranB            RandomNumber,
    timestampB      TimeStamp
}
```

TokenAB is then profiled as:

```
  TokenABU ::= SEQUENCE {
      ranA            RandomNumber,   -- unused
      entityA         EntityName,
      certA           Certificate,    -- from X.509 -- unused
      signature       SigDataABU
  }

  SigDataABU ::= SIGNATURE SEQUENCE {
      ranA            RandomNumber,   -- unused
      ranB            RandomNumber,   -- as sent in TokenBA
      entityA         EntityName
  }

  RandomNumber ::=        INTEGER
```

EntityName is a CHOICE and for this specification, the Name CHOICE is the only one acceptable. EmailName may not be used.

The following sections define the format of the request and response.

3.1. EAP DSS Public Key Request Packet

The DSS Unilateral Request packet Type Data field contains the data from the FIPS PUB 196 Token BA1.

This information is formatted in a length-value format. No explicit type field is necessary because all fields are required and are in a determinate order. In this one case the last element includes a length field also, even though its length can be determined from the overall length. This allows for easy expansion in this case. The EAP DSS Request packet has the following overall format:

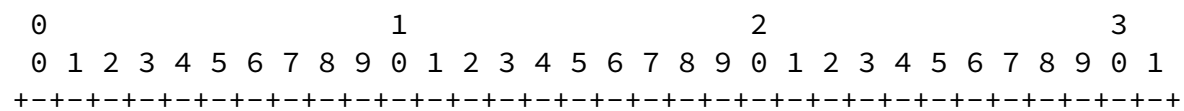


Figure 3.0-2 - EAP DSS Request Packet format

Code

1 (Request)

Identifier

The Identifier field is one octet and serves the same purpose as the TokenID field in FIPS PUB 196, namely disambiguating between multiple outstanding Requests and Responses. Handling of the Identifier field with respect to time-outs, new Requests, and duplicate Responses is as specified in EAP.

Length

The Length field is two octets and indicates the length of the EAP Request and Response packets including the Code, Identifier, Length, Type, timeStampLen, timeStamp, ranB Length, and ranB fields. Octets in the packet outside the range of the Length field should be treated as Data Link Layer padding and should be ignored on reception.

Type

The Type field in the Request will carry the value 10 (DSS Unilateral).

timeStampLen

The Length of the timeStamp field in bytes is specified here. A single byte is used to represent this length.

For the current version this value is 4.

timestampB

This value is a monotonically increasing (aside from wraparound) four byte integer in network byte order (Big Endian).

ranB Length

The Length of the ranB field in bytes is specified here. A single byte is used to represent this length. For the Fortezza version this value is 20.

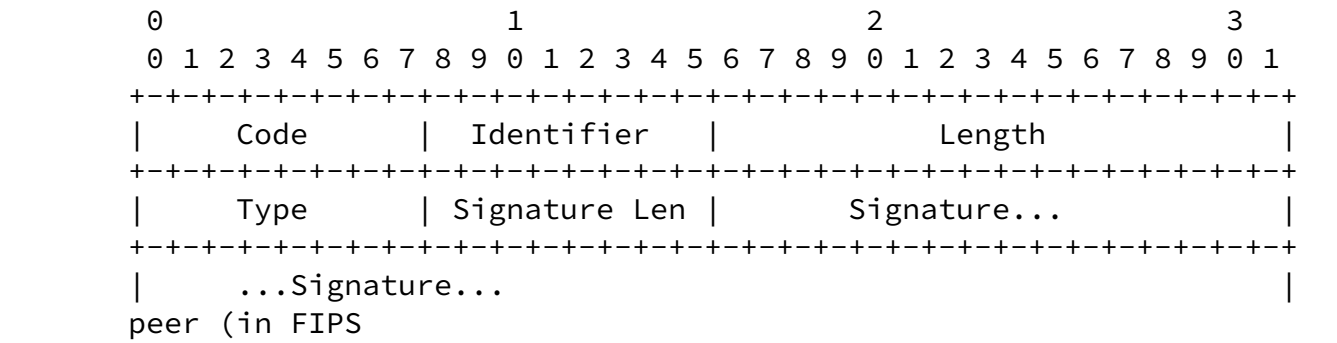
ranB

The value of the random challenge from the initiator to the responder. This value cannot exceed 255 bytes in length. For the Fortezza version the length of this field is 20 bytes.

3.2. EAP DSS Public Key Response Packet

The DSS Unilateral Response packet Type Data field contains the data from the FIPS PUB 196 Token AB1.

This information is formatted in a length-value format. No explicit type field is necessary because all fields are required and are in a determinate order. The last element does not include a length field because its length can be determined from the overall length. The EAP DSS Response packet has the following overall format:



peer (in FIPS 196 terms entity A). The peer signs the concatenation of the random challenge sent to it in the EAP DSS Request, the timestamp sent to it, and its own entity name from the certificate whose public key corresponds to the private key used in forming the signature. The entity name is the DER-encoded form of the Distinguished Name contained in the subject field of the certificate. The signature is computed as described under "digital signature" in section 1.2. This value cannot exceed 255 bytes in length.

DName

The DER-encoded form of the subject field in the X.509 certificate whose public key corresponds to the private key used by the entity to produce the signature value.

4. PPP EAP DSS Public Key Authentication Processing

If TokenAB is successfully verified by B and B is willing to operate a PPP link with A then B shall transmit an EAP Success packet. Otherwise, B may transmit an EAP Failure packet, and shall in all cases transmit an LCP Terminate-Request.

Figure 4.0-1 depicts the operation of the EAP Unilateral authentication protocol with DSS. In this and the following figures depicting PDU exchanges, the curly braces ({, }) denote items in Length-Value representation.

```
   Side:       B                  A

   Authenticator           Authenticatee

   EAP Request (ID1, DSS Unilateral, {ranB, timestampB }) =>

               <= EAPResponse(ID1,DSSUnilateral,
                      {SIGNA({ranB, timeStampB, entityA}), entityA)

   EAP Success Packet( ID2) =>
```

Figure 4.0-1 DSS Unilateral Authentication processing

Security Considerations

This memo defines a method for using EAP to perform Strong authentication of a peer using the DSS signature algorithm.

References:

[1] Simpson, W. A., 'The Point to Point Protocol (PPP)', July 1994, RFC 1661.

[2] Blunk, L. J. & Vollbrecht, J. R., 'PPP Extensible Authentication Protocol (EAP)', June 1996, work in progress.

[3] CCITT Recommendation X.509, 'The Directory - Authentication Framework', 1988.

[4] Federal Information Processing Standards Publication,

FIPS Pub 196, 'Entity Authentication using Public Key Cryptography', February 18, 1997.

[5] Zmuda, J., 'The PPP Certificate Exchange Protocol', July 1997, work in progress.

Acknowledgements:

This work is based largely on EAP. The authors would like to thank John Vollbrecht of Merit specifically for his help in understanding the intention of the EAP Internet Draft. The authors would also like to thank Paul Amaranth of Oakland University for his EAP implementation. Thanks also are due to Bill Whelan of Network Express for his Internet Draft showing a worked example of the use of EAP for public

key based authentication. Also both Peter Yee and Russ Housley provided helpful comments on earlier versions of this Memo. And thanks finally to Bill Simpson for the standard PPP spec boilerplate from which we have borrowed heavily.

Chair's Address:

The working group can be contacted via the current chair:

Karl Fox
Ascend Communications, Inc.

Email: karl@ascend.com

Author's Address:

Questions about this memo can also be directed to:

DIRNSA
Attn: X22 (W. Nace)
9800 Savage Road
Fort Meade, MD 20755-6000
USA

Phone: +1 410 859-4464

Email: WANace@missi.ncsc.mil

James E. Zmuda
SPYRUS
2460 N. First Street
Suite 100
San Jose, CA 95131-1023
USA

Phone: +1 408 432-8180
Email: jzmuda@spyrus.com