PPP EAP RSA Public Key Authentication Protocol
<draft-ietf-pppext-eaprsa-00.txt>

Status of this Memo

This document is a submission to the Point-to-Point Protocol Extensions Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the ietf-ppp@merit.edu mailing list.

Distribution of this memo is unlimited.

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as ‘work in progress.’

To learn the current status of any Internet-Draft, please check the ‘1id-abstracts.txt’ listing contained in the Internet-Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

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 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. One of these is RSA Public Key Authentication. This document defines RSA 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 RSA Public Key Authentication Protocol.

This document defines the PPP EAP RSA 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 RSA Public Key Authentication in the Configure-Request during Authentication phase.

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

RSA key pair A pair of keys, each of which can be
used to encode data or to reverse the encoding performed by the other. The two keys in the pair are known as the public and private keys.

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

**private key (secret key)**  That key of a users key pair which is known only by that user [3].

**digital signature**  Encipherment, by the originators secret key, of a compressed string of the relevant data to be transferred. The digital signature together with the plain data is sent to the recipient. This message is processed by the recipient to prove integrity. The digital signature mechanism also proves the authenticity of the originator and the unambiguous relationship between the originator and the data that was transferred [3].

A unique digital pattern produced by encoding a message digest of digital document using the signatorys private key. The signature can be produced only by someone holding the private key and cannot be applied to any but the intended digital document. The signature can be verified by anyone who has the signatorys public key.

**certificate**  The public keys of a user, together with some other information, rendered unforgeable by encipherment with the secret key of the certification authority which issued it [3].

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

2. **PPP EAP RSA 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 RSA Public Key Authentication at Authentication Phase.

RSA Public Key Authentication Protocol is a challenge-response protocol based on unilateral two pass authentication as described in ISO/IEC 9798-3 [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 peers
private key). The peer MUST also provide in its response its own public key as well as proof that it knows the corresponding private key. The Response MUST also contain the peers certificate produced by a trusted Certification Authority.

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 RSA Public Key Authentication plus some random data produced by the authenticator.

2. The peer sends a Response packet in reply to the Request. The exact contents of the Response is partially dependent upon the random data sent by the authenticator and partially dependent upon random data produced by the peer itself.

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].

PPP EAP RSA Public Key Authentication Packet Format

A summary of the PPP EAP RSA Public Key Authentication Request/Response packet format is shown below. The fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------+-----------+-----------+---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Identifier</td>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>+---------------------------------+-----------+-----------+---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Data ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------+-----------+-----------+---</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Code**: 1 - Request, 2 - Response
- **Identifier**: The identifier field is one octet and aids in matching responses with requests.
- **Length**: The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, and Data fields. Octets outside the range of the Length field should be treated as Data Link Layer padding and should be ignored on reception.
- **Type**: 9 - RSA Public Key Authentication
- **Data**: The format of the Data field is determined by the
3.1 RSA Public Key Request Packet

A summary of the PPP EAP RSA Public Key Authentication Request packet format is shown below. The fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>Code</th>
<th>Identifier</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>ChallengeVal...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChallengeVal...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChallengeVal...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChallengeVal...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChallengeVal...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Code

1

Identifier

The identifier field is one octet and aids in matching responses with requests. The identifier field MUST be changed on each Request packet containing a different ChallengeVal.

Length

21

Type

9

ChallengeVal

The ChallengeVal is sixteen octets of data. It SHOULD be generated by the authenticator in such a way as to assure that it cannot be predicted by an attacker using a playback attack. The ChallengeVal will affect the Response packet sent by the peer. The ChallengeVal SHOULD be changed each time a Request is sent. This includes retransmitted Requests in instances where a previously transmitted Request or corresponding Response is lost. Changing the ChallengeVal on retransmissions is recommended so as to make a replay attack more difficult.

3.2 RSA Public Key Response Packet

A summary of the PPP EAP RSA Public Key Authentication Response packet format is shown below. The fields are transmitted from left to right.
<table>
<thead>
<tr>
<th>Code</th>
<th>Identifier</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Cert Type</td>
<td>Certificate...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResponseVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ResponseVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ResponseVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ResponseVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ResponseVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ChallengeVal...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ChallengeVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ChallengeVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...ChallengeVal...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Signature Len</td>
<td>Signature...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Code**

2

**Identifier**

The identifier field is one octet and MUST match the Identifier field from the corresponding request.

**Length**

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, Certificate, Random Data, Echo Value, Signature Len, and Signature fields.

**Type**

9

**Cert Type**

This field identifies the type of certificate the peer is presenting. This is further defined in the following section.

**Certificate**

The peers certificate. This is further defined in the following section.

**ResponseVal**

The ResponseVal is a sixteen octet field. It SHOULD be generated by the peer in such a way as to assure that it cannot be predicted by an attacker.

**ChallengeVal**

The ChallengeVal is a sixteen octet field which MUST match the ChallengeVal which appeared in the corresponding Request packet.
Signature Length
The length in octets of the following signature.

Signature
The signature of the peer applied to the combination of ChallengeVal and ResponseVal. The peer produces this signature by taking the thirty-two octets formed by the ChallengeVal followed by the ResponseVal and producing an MD5 message digest. The 128 bit message digest is then encrypted by the peer using the peer’s private key. To verify this signature, the authenticator will decrypt the peer’s signature using the peer’s public key. The authenticator would also produce an MD5 message digest using the ChallengeVal followed by the ResponseVal. The two results would then be compared for equality.

Certificates
For the purposes of this document, it is necessary that a certificate provide the RSA public key of the owner as well as some identification of the certificate owner. These information must be signed by a mutually trusted certifying authority.

Other applications may require additional information. It is desirable to be able to use any certificate formats providing the necessary information so as to not require owners to possess different certificate formats for different applications.

Since it is impossible to predict in advance all the possible information one may ever want in a certificate, it was decided to provide for the support of multiple certificate types. One certificate type is described here. Compliance with PPP EAP RSA Public Key Authentication requires support of this certificate type. Support for other certificate types is optional.

Certificate Type
1 - Mandatory

Certificate Format:

<table>
<thead>
<tr>
<th>Certificate Length</th>
<th>Identifier Len</th>
<th>Identifier Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key Exp Length</td>
<td>Public Key Exponent...</td>
<td></td>
</tr>
<tr>
<td>Key Mod Length</td>
<td>Public Key Modulus...</td>
<td></td>
</tr>
<tr>
<td>Signature Len</td>
<td>Signature...</td>
<td></td>
</tr>
<tr>
<td>Certificate Length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Certificate Length
The length in octets of the entire certificate including all contiguous fields from the Certificate Length through the Signature fields.

Identifier Len
The total length in octets of the Identifier Type and Identification fields.

Identifier Type
This refers to the type of identification provided by the Identification field.

1 - Identification field contains a name.

Identification
Format depends upon the value in the Identifier Type. For Identifier Type = 1, this will be a name. The length of the name will depend upon the Identifier Len. The name will not be null terminated.

Key Exp Len
Length in octets of the Public Key Exponent field.

Public Key Exponent
The value of the RSA public key exponent in network byte order.

Key Mod Len
Length in octets of the Public Key Modulus field.

Public Key Modulus
The value of the RSA public key modulus in network byte order.

Signature Len
Length in octets of the Signature field.

Signature
This field contains the signature of a certifying authority applied to all contiguous fields within the certificate from the Identifier Len through the Public Key Modulus. To produce this signature, the certifying authority will produce an MD5 message digest of these fields then encrypt the result using its RSA private key.

Security Considerations

RSA Public Key Authentication is designed to allow an authenticator to obtain the public key from a peer and to authenticate the peer by requiring the peer to corroborate its identity by demonstrating its knowledge of its private key. The process is based on the unilateral two pass authentication described in ISO/IEC 9798-3 [4].

The use of random data fields within this protocol has been defined to prevent replay or interleaving attacks. Each entity (authenticator and peer) SHOULD provide
unpredictable random data so as to assure an adequate level of security.

RSA Public Key Authentication MUST provide for the following requirements:

1. The peer MUST possess a valid certificate signed by a mutually recognized certifying authority.

2. The peer MUST possess a valid private key corresponding to the public key in the peers certificate.

To assure these requirements are met, upon receipt of the EAP Response packet, the authenticator SHOULD verify the certification authority's signature within the certificate by decrypting using the certification authority's public key and compare the resulting information to the rest of the certificate. The authenticator SHOULD then obtain the peers public key from the peers certificate and use it to decrypt the peers signature to verify it matches the two sets of random data.

The set of recognized certifying authorities is implementation dependent. As a minimum, an authenticator MUST recognize any certifying authority which signed the authenticators certificate. Implementations MAY recognize certifying authorities from a hierarchical chain.

If the peers EAP Response satisfies all requirements, the authenticator MUST send an EAP Success packet. If the peers EAP Response does not satisfy all requirements, the authenticator MUST send an EAP Failure packet and MAY take down the link.

References


Acknowledgments
Dave Carrell, Jeff Schiller, and Fred Baker provided valuable feedback on earlier versions of this draft.

Chair’s Address

The working group can be contacted via the current chair:

Fred Baker  
Cisco Systems, Inc.  

Email: fbaker@cisco.com

Author’s Addresses

Questions about this memo can also be directed to:

William T. Whelan  
bwhelan@nei.com