PPP EAP SRP-SHA1 Authentication Protocol

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

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 inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This document is the product of 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.

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 Authentication Protocol (EAP) [2] framework to allow the use of multiple authentication mechanisms without fixing the mechanism to be used during initial link negotiation. This document defines an authentication mechanism for EAP called SRP-SHA1, and allows the use of the Secure Remote Password (SRP) [3] protocol as a PPP link authentication method.
1. Introduction

PPP-EAP allows the use of multiple authentication mechanisms within a single negotiation framework. This design overcomes the chief design limitation of the original PPP authentication mechanisms, PAP [4] and CHAP [5], which require that the protocol to be used for authentication be chosen before the user’s identity is known. EAP also simplifies the process of adding authentication mechanisms to PPP.

SRP is an open source protocol that provides strong, password-based authentication based on cryptographic hashing. Unlike PAP, the password never appears on the wire. Unlike CHAP (and variants MS-CHAPv1 [6] and MS-CHAPv2 [7]), access to the cleartext password is not required for the authenticator. Unlike all of these authentication protocols, SRP is resistant to dictionary attacks against the over-the-wire information. SRP is also resistant to eavesdropping and active attacks. As a side-effect, SRP also creates a session key that is resistant to man-in-the-middle attacks and can be used for data encryption.

SHA-1 [8] is a message digest algorithm that can be used as a hashing mechanism for SRP, producing an SRP variant known as SRP-SHA1. For
calculation of the shared key in SRP, SHA-1 is used in an interleaved form to produce 40 octet hashes. See [3] for details.

This document describes the message exchanges REQUIRED for one PPP peer to authenticate the other using EAP and SRP-SHA1. As with all PPP authentication mechanisms, the peers are independent and equal, and either or both may demand authentication from the other, and different protocols may be used independently in each direction.

When the PPP Encryption Control Protocol (ECP) [9] is used along with EAP SRP-SHA1, this document describes methods that SHOULD be used to generate the necessary shared keys from the SRP-SHA1 session key. Because authentication necessarily requires prior arrangement between the peers, pre-configured keys MAY be used in place of the SRP-SHA1 session key, and any such selection is outside the scope of this document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [10].

2. Detailed Description of EAP SRP-SHA1 Authentication

Unlike CHAP, SRP-SHA1 requires more than one exchange to authenticate the peer. For this reason, three message subtypes are defined.

With SRP, the authenticator must communicate at least three values to the authenticatee, referred to as 's' (the password salt), 'B' (an ephemeral public key), and 'M2' (a hash value). The authenticatee must communicate at least three values to the authenticator, referred to as 'C' (the client name), 'A' (an ephemeral public key), and 'M1' (a hash value).

(The value 'u' passed from authenticator to authenticatee in the general SRP algorithm is derived from the first 32 bits of a SHA-1 hash of the 'B' value itself in the SRP-SHA1 algorithm. Thus, it does not need to be sent explicitly.)

In order to send its last value (M1), the authenticatee needs A, B, and u. However, in order to guarantee that the authenticatee’s value chosen for A isn’t a rogue value (see section 3.2.4 of the SRP description [11]), the value of u must not be sent until the authenticatee reveals A. This implies that there are at least three steps -- authenticatee sends A, authenticator sends B, authenticatee sends M1.

The adaptation described in this document recommends the use of the
EAP Identity Request/Response mechanism to obtain C from the peer. It then uses a new message type, called EAP SRP-SHA1, with three subtypes to handle the SRP authentication. The Subtype 1 Request ("Challenge") message contains s, g, and N. The Subtype 1 Response ("Client Key") contains A. The Subtype 2 Request ("Server Key") continues with B and the Subtype 2 Response ("Client Validator") contains M1. Finally, the Subtype 3 Request ("Server Validator") contains the M2 value and the Subtype 3 response is an acknowledgement containing only the Subtype number and no data.

2.1. EAP SRP-SHA1 Packet Formats

All SRP authentication is driven by the authenticator (server). The authenticatee (client) MUST NOT retransmit Response messages, but SHOULD terminate the link if a Request is not received within a reasonable time period. The authenticator SHOULD silently ignore unexpected Response messages. The authenticatee MUST respond using EAP Nak if any unknown Subtype or otherwise unacceptable message is received.

A summary of the PPP EAP SRP-SHA1 Request/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>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Subtype-Data...</th>
</tr>
</thead>
</table>

Code

1 - Request
2 - Response

Identifier

The Identifier field is one octet and aids in matching responses with requests. The Identifier SHOULD be changed for each Request message sent. The Identifier in the Response message MUST match the corresponding Request. The authenticator MUST discard non-matching Response messages.
Length

The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, Subtype, and Subtype-Data fields. Octets outside the range of the Length field should be treated as Data Link Layer padding and should be ignored on reception. Truncated packets (with Length greater than the link layer reported length) MUST be silently discarded.

Type

19 - EAP SRP-SHA1

Subtype

1 - Challenge / Client Key
2 - Server Key / Client Validator
3 - Server Validator

The Subtype field is one octet and must contain the value 1, 2, or 3. If any other subtype value is encountered in an EAP SRP-SHA1 Request message, then the authenticatee SHOULD return an EAP Response with the Type field set to Nak. In EAP SRP-SHA1 Response messages, the Subtype value must be copied from the corresponding Request message. The authenticator should treat unknown Subtype values in Response messages as malformed packets and silently discard.

Subtype-Data

The format of the Subtype-Data field is determined by the Code and Subtype fields as described in the sections below.

2.2. EAP SRP-SHA1 Challenge Subtype-Data

The PPP EAP SRP-SHA1 Challenge (Subtype 1 Request) packet MUST be sent after the peer's identity has been obtained; use of the EAP Identity Request/Response packet as described in [2] is RECOMMENDED. Using the peer's unauthenticated identity, the authenticator MUST look up the password salt, verifier ('v'), prime modulus, and generator values in an implementation-dependent manner. Use of EAP Identity is not required by this specification, and determination of salt, verifier, prime modulus, and generator may be done in any convenient manner.

A summary of the PPP EAP SRP-SHA1 Challenge Subtype-Data format is shown below. Code (1), Identifier, Length, Type (19), and Subtype
(1) fields are as described above. The fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>Hash Type</th>
<th>Salt Length</th>
<th>Salt ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prime Modulus Length</td>
<td>Prime Modulus ...</td>
<td></td>
</tr>
<tr>
<td>Generator ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hash Type

A single octet specifying the type of hash algorithm to be used with SRP. For SHA-1 based SRP as described in this document, this value MUST be set to 1. An authenticatee that does not support the hash algorithm specified by the authenticator MUST reply with EAP Nak to disable EAP SRP authentication.

Salt Length

A single octet giving the length of the Salt field in octets. This MUST be at least 4 octets and MAY be any length up to 255 octets.

Salt

Password salt string; may contain any values and be of any length, subject to link MTU restrictions. This field is not padded.

Prime Modulus Length

Length of the Prime Modulus field in octets. This value MAY be zero to select the 2048 bit value for N listed in Appendix A and select g=2. In this case, the Prime Modulus and the Generator field MUST NOT be present in the message.

If the Prime Modulus Length field is non-zero, then it SHOULD be at least 64 octets (512 bits). Longer values are RECOMMENDED.

Prime Modulus

Prime Modulus value. This value (called ’N’ in the SRP documentation) is in network byte order and has a length equal to the Prime Modulus Length field.
Generator

The Generator value. This value (called ‘g’ in the SRP documentation) is in network byte order and fills the rest of the message without padding. If the Generator field is omitted, then g is set to 2.

2.3. EAP SRP-SHA1 Client Key Subtype-Data

The PPP EAP SRP-SHA1 Client Key (Subtype 1) Response message MUST be sent in reply to an EAP SRP-SHA1 Subtype 1 Request message.

A summary of the PPP EAP SRP-SHA1 Client Key Subtype-Data format is shown below. Code (2), Identifier, Length, Type (19), and Subtype (1) fields are as described above. The fields are transmitted from left to right.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------+
| Value A ...                                   |
+---------------------------------------------+
```

Value A

The result of \((g^a) \mod N\), where ‘a’ is a randomly chosen number in the range 1 .. N (exclusive). The randomly chosen number is the authenticatee’s private key, and the Value A field is the corresponding public key. This field is in network byte order and is not padded.

The A value MUST NOT be zero. If the authenticator receives zero for this field, it SHOULD take action to disconnect the link.

2.4. EAP SRP-SHA1 Server Key Subtype-Data

The PPP EAP SRP-SHA1 Server Key (Subtype 2 Request) message MUST NOT be sent until a valid Client Key Response message has been received. The Subtype 2 Request message SHOULD be retransmitted on time-out.

A summary of the PPP EAP SRP-SHA1 Server Key Subtype-Data format is shown below. Code (1), Identifier, Length, Type (19), and Subtype (2) fields are as described above. The fields are transmitted from left to right.
The result of \( (v + g^b) \mod N \), where ‘\( b \)’ is a randomly chosen number in the range 1 .. N (exclusive), and \( v \) is the stored verifier from the authentication database. The randomly chosen number is the authenticator’s private key, and the Value B field is the corresponding public key. This field is in network byte order and is not padded.

The B value MUST NOT be zero. If the authenticatee receives zero for this field, it SHOULD take action to disconnect the link.

### 2.5. EAP SRP-SHA1 Client Validator Subtype-Data

The PPP EAP SRP-SHA1 Client Validator (Subtype 2 Response) message MUST be sent by the authenticatee in response to a valid EAP SRP-SHA1 Subtype 2 Request. The authenticator validates this Response message by calculating the M1 value from its own copies of A, B, and K, and compares the result. If the values match, then the authentication is successful, and EAP SRP-SHA1 Server Validator Request SHOULD be sent. If the values do not match, then EAP Failure SHOULD be returned and the link terminated.

A summary of the PPP EAP SRP-SHA1 Client Validator Subtype-Data format is shown below. Code (2), Identifier, Length, Type (19), and Subtype (2) fields are as described above. The fields are transmitted from left to right.
Value M1

The 20 octet result of SHA1(SHA1(N) xor SHA1(g), SHA1(peername), s, A, B, K).

As described in SRP [3], the authenticatee computes \( x = \text{SHA1}(s, \text{SHA1}(\text{peername}, ":", \text{password})) \), and then computes \( K = \text{SHA1}_\text{Interleave}((B - g^x)^(a + u*x) \mod N) \). The authenticator computes \( K = \text{SHA1}_\text{Interleave}((A * v^u)^b \mod N) \). The calculated K values MAY be used with ECP (see section 3). Finally, M1 is then computed as SHA1(SHA1(N) xor SHA1(g), SHA1(peername), s, A, B, K).

(Note that reference [11] gives different definitions of these values; the [3] document should be treated as the normative reference.)

The above described SHA1 hash to produce the long-term private key x, as described in [3], SHOULD be used by default in EAP SRP-SHA1. As an implementation open, however, the x value used above MAY instead be derived from any mutually-chosen hashing algorithm, provided that the security of that algorithm is acceptable to both authentication parties.

2.6. EAP SRP-SHA1 Server Validator Subtype-Data

The PPP EAP SRP-SHA1 Server Validator (Subtype 3 Request) message MUST be sent by the authenticator after reception of a valid EAP SRP-SHA1 Subtype 2 Response. If the authenticatee receives a Server Validator message with invalid contents, it MUST terminate the link.

A summary of the PPP EAP SRP-SHA1 Server Validator Subtype-Data format is shown below. Code (1), Identifier, Length, Type (19), and Subtype (3) fields are as described above. 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>Value M2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td>M2 (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td>M2 (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td>M2 (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td>M2 (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td>M2 (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------------------------+</td>
<td>M2 (continued)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Value M2

The 20 octet result of SHA1(A, M1, K).

2.7. EAP SRP-SHA1 Subtype 3 Response

The authenticatee MUST transmit a PPP EAP SRP-SHA1 Subtype 3 Response message in reply to a valid Server Validator message from the peer. This Response message has no Subtype-Data field. The Code (2), Identifier, Length (6), Type (19), and Subtype (3) fields are as described above.

3. Use with ECP

The 40 octet value K calculated by the SRP-SHA1 authentication procedure SHOULD be used to form encryption keys if PPP encryption is in use and K is available (see section 4). For the DESEbis [12] algorithm, a shared 56 bit key is necessary, and for the 3DES [13] algorithm, a shared 168 bit key is required. Complicating this, ECP may be negotiated in only one direction or both. In addition, PPP authentication may be performed one-way (the most common case) or mutually, and when mutual authentication is chosen, different authentication schemes may be used to authenticate each peer. The effects of these details are described below.

The "decryptor" is the peer that sends ECP Configure-Request suggesting an algorithm and receives a corresponding ECP Configure-Ack. The "encryptor" is the sender of the ECP Configure-Ack, and will transmit the encrypted data.

Rekeying can be accomplished by restarting EAP authentication.

3.1. One-Way Versus Bidirectional Encryption

When encryption is employed in only one direction, then the K value that is chosen for the shared key is the value associated with the EAP SRP-SHA1 authentication in which the decryptor is the authenticator. If the decryptor did not authenticate its peer with EAP SRP-SHA1, then the K value associated with the other authentication session is used instead.
3.2. One-Way Versus Mutual Authentication

If only one peer authenticates the other using SRP-SHA1, and the other either does not authenticate its peer at all or uses a method that does not result in encryption keys, then the one K value associated with this authentication SHOULD be used for both encryption sessions. The first 20 octets of K are used for the encryption of data sent by the authenticatee to the authenticator, and the second 20 octets of K are used for encryption of data in the opposite direction.

If mutual authentication with algorithms that produce encryption keys is done, such as SRP-SHA1, then two K values are produced. The K values are used so that the encryptor is the authenticatee for the corresponding authentication session, and the decryptor is the authenticator.

3.3. DESEbis Versus 3DES

For DESEbis, the first 8 octets of the key value are used. DES keys are constructed such that the most significant bit (MSB) of each octet is reserved for parity.

For 3DES, 24 octets of key value are used by most implementations. As for DESEbis, the MSBs are discarded. If the 40 octet K value has been split into two 20 octet values because one-way authentication is in use, then an extra 4 octets are needed for each key. The SHA-1 algorithm is run again over the 40 octet K value to produce a 20 octet hash. This hash is split into two 10 octet values that are then appended to the two 20 octet values from the split K value. The first 24 octets of each 30 octet value is used as the 3DES key.

4. Use with AAA

The SRP exchange uses the PPP peername as part of the calculation, and thus depends on leaving this string unmodified between the authenticatee and authenticator. If a mechanism, such as a AAA protocol, is used to perform the SRP authentication on behalf of the PPP "server," then the peername MUST be identical on both ends. In particular, if a Network Access Identifier [14] is used with a proxy authentication server, then the implementor MUST take steps to preserve the peername string end-to-end.

Use of a AAA protocol also makes the use of the session key (K) for ECP keying problematic, because the contents of this key are available only at the authentication endpoints (where SRP is run), and not
the link layer endpoints (where ECP is run). Use of a directory access protocol for the hash values, rather than a AAA protocol, would solve this problem.

5. Security Considerations

The security of SRP on the wire relies on having a prime modulus that is large enough to make brute force attack of the key exchange infeasible. A length of at least 1024 bits is recommended. In addition, the SRP document [3] describes tests that MUST be performed on the chosen modulus and generator values and random numbers. SRP is a significant improvement over the situation with PAP, which has no on-the-wire security, and with CHAP, which is vulnerable to dictionary attacks against a captured Challenge/Response pair.

The security of the credentials stored in the authenticator’s database relies on the entropy in the chosen password, the difficulty of hash inversion of a salted string, and the security of the system containing the database. For this reason, the chosen password SHOULD be restricted to limit the effectiveness of dictionary attacks against a disclosed database. However, since typical passwords have only a few bits of entropy, the database itself MUST be protected against disclosure.

6. Intellectual Property Rights Notices

6.1. Patent Issues

The SRP key-exchange protocol described in this document is available worldwide on a royalty-free basis for commercial and non-commercial uses.

"The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this
specification can be obtained from the IETF Secretariat."

"The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director."

6.2. Full Copyright Statement

"Copyright (C) The Internet Society 2001. 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 DISCLAIM 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."

7. References


8. Acknowledgements

The author is grateful for the many critiques and ideas offered on the IETF PPP Extensions mailing list and by private mail. In particular, I thank Bernard Aboba, Jacques Caron, and Vernon Schryver.
9. Author’s Address

James Carlson
Sun Microsystems
1 Network Drive MS UBUR02-212
Burlington MA  01803-2757

Phone:  +1 781 442 2084        Email:  james.d.carlson@sun.com
Fax:    +1 781 442 1677

10. Appendix A - Well-Known Prime Modulus

This 2048 bit prime modulus (and corresponding generator value 2) are borrowed from the SRP GSS-API mechanism, an IETF work in progress.

0xAC, 0x6B, 0xDB, 0x41, 0x32, 0x4A, 0x9A, 0x9B,
0xF1, 0x66, 0xDE, 0x5E, 0x13, 0x89, 0x58, 0x2F,
0xAF, 0x72, 0xB6, 0x65, 0x19, 0x87, 0xEE, 0x07,
0xFC, 0x31, 0x92, 0x94, 0x3D, 0xB5, 0x60, 0x50,
0xA3, 0x73, 0x29, 0xCB, 0xB5, 0x99, 0xED,
0x81, 0x93, 0xE0, 0x75, 0x4E, 0x53, 0x29,
0x5E, 0x27, 0x90, 0x04, 0x1C, 0x7B, 0xC3, 0x08,
0x71, 0x35, 0xF8, 0x2F, 0x8F, 0x56, 0x98, 0xA8,
0xA3, 0x73, 0x29, 0xCB, 0xB5, 0x99, 0xED,
0x81, 0x93, 0xE0, 0x75, 0x4E, 0x53, 0x29,