[SRP-PROTO]:
Secure Remote Password Protocol

Revision History

Revision summary			
Author	**Date**	**Revision history**	**Comments**
Marc-André Moreau	07/01/2019	1.0	Initial draft

Release: August 1st, 2019

Contents

1 Introduction

The Secure Remote Password (SRP) protocol allows mutual authentication between a client and a server that both know the same password without ever exchanging it directly over the network. Since it is a Password-Authenticated Key Agreement (PAKE) protocol, a secret session key is generated as a result of this exchange.

2 Protocol Details

This section describes the Secure Remote Password (SRP) protocol details.

2.1 Common Details

The extended SRP protocol is an SRP-6a implementation compliant with RFC5054 and RFC2945. The default hashing algorithm is SHA256, and the default prime size is 2048 bits.

3 Messages

This section describes the SRP protocol messages.

3.1 Transport

The SRP protocol is meant to be transport agnostic, but adaptable to different use cases.

3.1.1 Hypertext Transfer Protocol (HTTP)

The HTTP authentication scheme name for SRP is "SRP".

While HTTPS is recommended, SRP can be used with an insecure transport such as HTTP since it provides its own layer of security.

SRP is a multilegged authentication protocol, meaning that it requires the exchange of a series of messages to be completed. Since HTTP is inherently stateless, the client and server have no standard to associate messages belonging to the same authentication sequence. To solve this problem, the solution documented in [draft-montenegro-httpbis-multilegged-auth] is recommended.

SRP HTTP authentication works as follows:

The client sends an HTTP GET request.

The server responds with an HTTP 401 Unauthorized response with the following header fields:

- WWW-Authenticate: SRP
- Auth-ID: <auth-id-token>

The client sends a new HTTP GET request with the following header fields:

- Authorization: SRP <srp-msg1-base64>
- Auth-ID: <auth-id-token>

The server responds with an HTTP 401 Unauthorized response with the following header fields:

- WWW-Authenticate: SRP <srp-msg2-base64>
- Auth-ID: <auth-id-token>

The client sends a new HTTP GET request with the following header fields:

- Authorization: SRP <srp-msg2-base64>
- Auth-ID: <auth-id-token>

The server responds with an HTTP 401 Unauthorized response with the following header fields:

- WWW-Authenticate: SRP <srp-msg3-base64>
- Auth-ID: <auth-id-token>

The client sends a new HTTP GET request with the following header fields:

- Authorization: SRP <srp-msg4-base64>
- Auth-ID: <auth-id-token>

If authentication is successful, the server responds with an HTTP 200 OK response with the following header fields:

- Auth-ID: <auth-id-token>

The authentication context associated with the Auth-ID field is no longer required after this step.

If authentication fails, the server responds with an HTTP 403 Forbidden response and the following header fields:

- Auth-ID: <auth-id-token>

This response can be sent by the server at any time. The authentication context associated with the Auth-ID field is deleted after this step.

3.2 Message Syntax

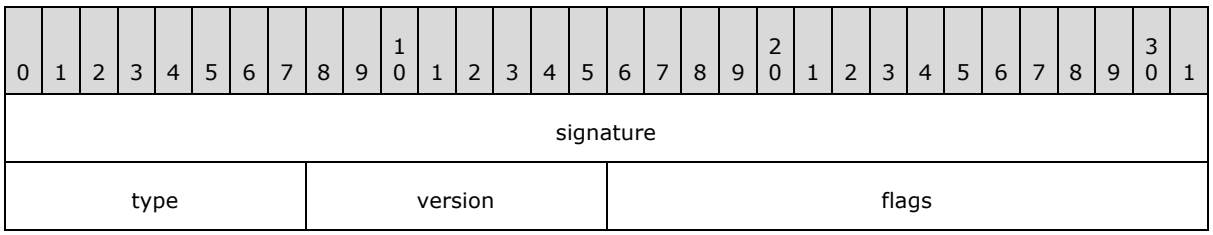
This section describes the protocol message encoding.

3.2.1 Protocol Messages

This section describes the encoding of protocol messages.

3.2.1.1 SRP_HEADER

The SRP_HEADER structure is shared by all SRP messages.



signature (4 bytes): The SRP message signature. This field MUST contain the null-terminated 4-byte string "SRP". As a 32-bit litte-endian unsigned integer, the signature is equal to 0x00505253.

type (1 byte): The SRP message type.

Value	Meaning
SRP_INITIATE_MSG_ID 0x01	SRP_INITIATE_MSG
SRP_OFFER_MSG_ID 0x02	SRP_OFFER_MSG
SRP_ACCEPT_MSG_ID 0x03	SRP_ACCEPT_MSG

Value	Meaning
SRP_CONFIRM_MSG_ID 0x04	SRP_CONFIRM_MSG

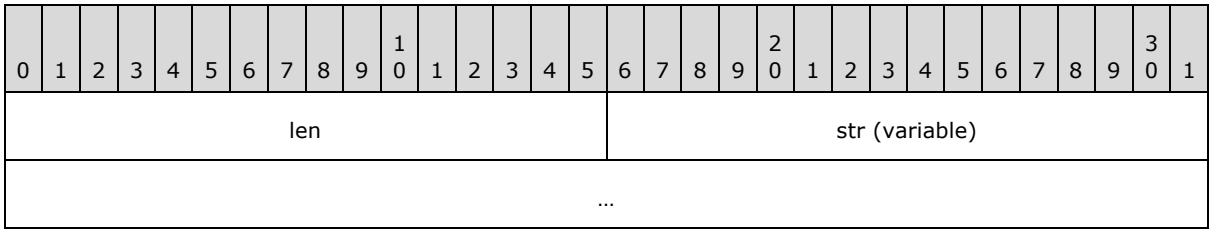
version (1 byte): The SRP protocol version, MUST be set to 6.

flags (2 bytes): The SRP message flags.

Flag	Meaning
SRP_FLAG_MAC 0x0001	A 32-byte Message Authentication Code (MAC) is present at the end of this message.

3.2.1.2 SRP_STRING

The SRP_STRING structure is used to represent variable-length strings.

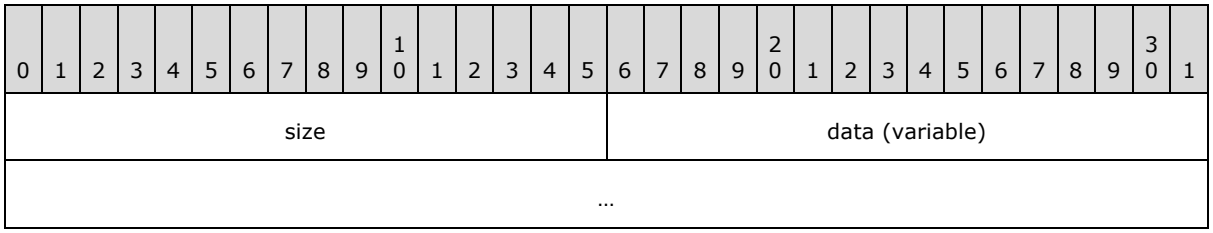


len (2 bytes): An unsigned 16-bit number containing the string length, excluding the null terminator.

str (variable): The UTF-8 encoded string including the null terminator, meaning the actual size of the str field is (len + 1).

3.2.1.3 SRP_BUFFER

The SRP_BUFFER structure is used to represent variable-length buffers.

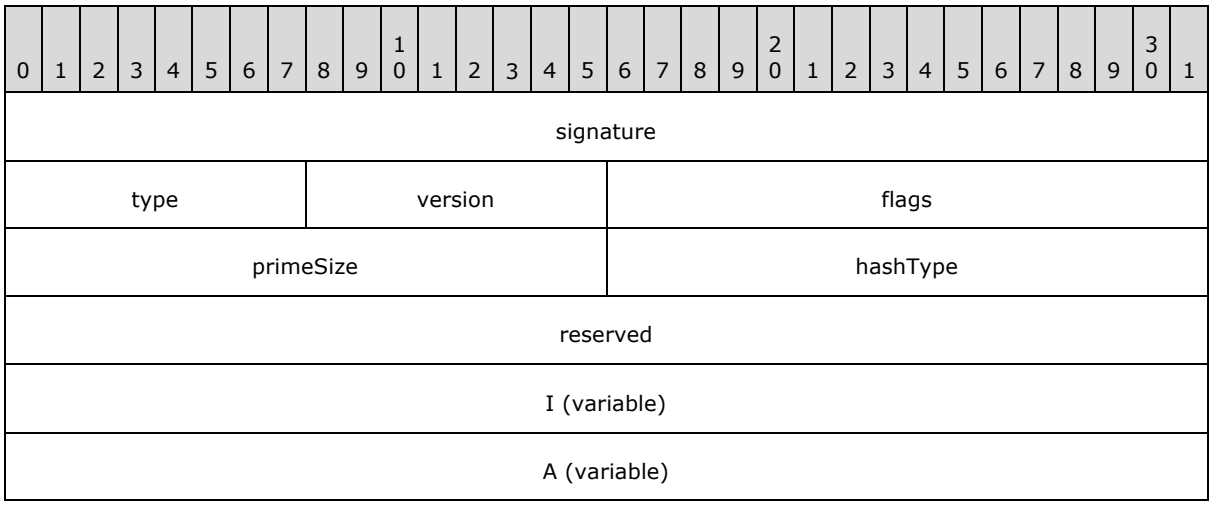


size (2 bytes): An unsigned 16-bit number containing the buffer size.

data (variable): The buffer data, whose size is given by the size field.

3.2.1.4 SRP_INITIATE_MSG

The SRP_INITIATE_MSG structure is sent client-to-server and is the first message of the authentication sequence.



signature (4 bytes): The SRP message signature.

type (1 byte): The SRP message type, MUST be set to 1.

version (1 byte): The SRP protocol version, MUST be set to 6.

flags (2 bytes): The SRP message flags.

primeSize (2 bytes): The prime size, in bytes. This field MUST be set to one of the following values: 256 (2048 bits), 512 (4096 bits) or 1024 (8192 bits). Values smaller than 256 (2048 bits) are not supported because they are considered too weak for modern use. The prime and generator values are not transmitted over the network, and the SRP groups are the ones from RFC5054.

hashType (2 bytes): The hashing function type. The field contains a multihash code as defined in the multicodec table (https://github.com/multiformats/multicodec). The value SHOULD be set to 0x12 (SHA256) because it is the only currently supported algorithm.

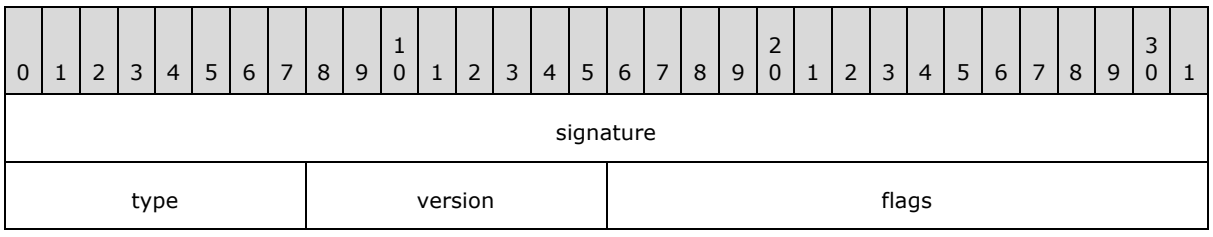
reserved (4 bytes): This field is reserved for future use and MUST be set to zero.

I (variable): An SRP_STRING structure containing the identifying username for authentication. Hashing MUST be performed using the string length, not its size, which means the null terminator is excluded even if it is encoded over the network.

A (variable): An SRP_BUFFER structure containing the SRP 'A' buffer, the user ephemeral key.

3.2.1.5 SRP_OFFER_MSG

The SRP_OFFER_MSG structure is sent server-to-client and is the second message of the authentication sequence.



primeSize	hashType
reserved	
s (variable)	
B (variable)	

signature (4 bytes): The SRP message signature.

type (1 byte): The SRP message type, MUST be set to 2.

version (1 byte): The SRP protocol version, MUST be set to 6.

flags (2 bytes): The SRP message flags.

primeSize (2 bytes): The negotiated prime size.

hashType (2 bytes): The negotiated hash function type.

reserved (4 bytes): This field is reserved for future use and MUST be set to zero.

s (variable): An SRP_BUFFER structure containing the SRP 's' buffer, the host salt.

B (variable): An SRP_BUFFER structure containing the SRP 'B' buffer, the host ephemeral key.

3.2.1.6 SRP_ACCEPT_MSG

The SRP_ACCEPT_MSG structure is sent client-to-server and is the third message of the authentication sequence.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
signature																															
type								version								flags															
M (variable)																															

signature (4 bytes): The SRP message signature.

type (1 byte): The SRP message type, MUST be set to 3.

version (1 byte): The SRP protocol version, MUST be set to 6.

flags (2 bytes): The SRP message flags. The SRP_FLAG_MAC flag MUST be set.

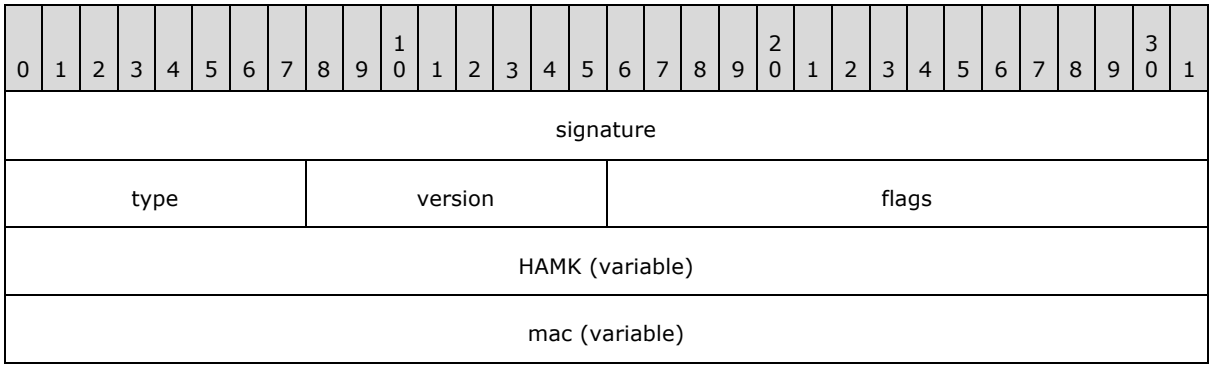
M (variable): An SRP_BUFFER structure containing the SRP 'M' buffer.

mac (variable): A trailing buffer containing the message authentication code (MAC). The size of this buffer MUST correspond to the negotiated hashing function, such as 32 bytes for SHA256. The MAC is the HMAC of all SRP messages in the sequence, including the current one, but excluding all mac fields.

The server MUST compute the MAC independently and abort the authentication sequence if the value doesn't match.

3.2.1.7 SRP_CONFIRM_MSG

The SRP_CONFIRM_MSG structure is sent server-to-client and is the fourth message of the authentication sequence.



signature (4 bytes): The SRP message signature.

type (1 byte): The SRP message type, MUST be set to 4.

version (1 byte): The SRP protocol version, MUST be set to 6.

flags (2 bytes): The SRP message flags. The SRP_FLAG_MAC flag MUST be set.

HAMK (variable): An SRP_BUFFER structure containing the SRP 'HAMK' buffer.

mac (variable): A trailing buffer containing the message authentication code (MAC). The size of this buffer MUST correspond to the negotiated hashing function, such as 32 bytes for SHA256. The MAC is the HMAC of all SRP messages in the sequence, including the current one, but excluding all mac fields. The client MUST compute the MAC independently and abort the authentication sequence if the value doesn't match.