SIPCORE Working Group
Internet-Draft
Intended Status: Standards Track
Expires: January 7, 2017

Fuwen Liu
Minpeng Qi
Zuo Min
Chinamobile
July 7, 2016

SIP Authentication using the EC-SRP5 Protocol
draft-liu-sipcore-ec-srp5-03

Abstract

This document specifies how the elliptic curve secure remote protocol (EC-SRP) is applied to SIP authentication. SIP Client and server perform mutual authenticate by using the modern 'zero knowledge' method without disclosing the password in the process. It has low computation complexity and low bandwidth consumption due to the use of elliptical curve cryptography. This makes it more suitable for resource-constrained environments,e.g. wireless network. The security of the scheme is based on the computational intractability of the elliptic curve discrete logarithm problem. It is resilient to various kinds of attacks, including off-line dictionary attacks.

Status of this Memo

Table of Contents

1 Introduction

SIP [1] applies HTTP digest authentication [2] [19] by default to performing user authentication. It is designed on the basis of the challenge-response mechanism, where the server presents the client a challenge (randomly-generated number), and the client responses with a valid answer which is generated by hashing the challenge in conjunction with the password. This is a weak authentication because it is possible for an attacker to recover the used password. Although passwords are not transmitted in a clear form over the insecure network, an adversary is still able to acquire the correct password by using a special variant of the brute-force attack: the off-line dictionary attack [3]. This results from the low entropy of a human-chosen password. The length of passwords mostly used in practice is rarely longer than 8 characters. It has merely about 30 bits of entropy (2^30) if the password is chosen by a human [4].

A Key-Derivation Authentication Scheme [5] has been proposed for SIP authentication. It creates a master-key by using a key-derivation function (KDF), whose inputs include a password,a salt, a key length, and an iteration count. A good example of KDF is HMAC [6]. The major difference between the HTTP digest authentication and the Key-Derivation Authentication is that the former performs the HMAC computation only once, while the latter computes HMAC n times, where n is the iteration count whose default value is 1000. This method could slow down the speed of off-line dictionary attacks. But it is not cryptographically secure as it needs just more 999 HMAC computations compared to the HTTP digest authentication when checking the correctness of a guessed password. Accordingly, the attacker can recover the password in a reasonable time using the off-line dictionary attacks.

Bellovin and Merritt first introduced an innovative password-based protocol, called DH-EKE (Diffie-Hellman Encrypted Key Exchange) protocol [7], to foil off-line dictionary attacks. Its basic idea is that two parties exchange ephemeral DH public keys encrypted with a shared password. Only the parties who know the password are able to authenticate each other and agree upon a session key for securing the communication. The computation complexity of off-line dictionary attacks on the DH-EKE protocol is equal to that of solving the discrete logarithm problem because the password is entangled with the ephemeral DH public key.

Inspired by the DH-EKE protocol, numerous password-based authentication key agreement protocols have been developed. Typical examples are the PAK (Password Authenticated Key exchange) protocol [8], SPEKE (Secure Password Exponential Key Exchange) protocol [9],AMP (Authentication via Memorable Password) protocol [10], and

SRP (Secure Remote Password) protocol [11]. They have been adopted as the standards by the IEEE computer society, including the elliptical curve (EC) variants of these protocols[12]. The EC-SRP5 protocol [13], a EC variant of the SRP protocol is chosen for SIP authentication in the demo, since it is much more efficient than the original SRP protocol regarding the computation and bandwidth consumption due to the use of elliptic curve cryptography.

Elliptic curve cryptography systems [14] are constructed by using elliptic curve over finite fields, which supersedes the conventional asymmetrical cryptographic algorithms, e.g., RSA and DH, in terms of computational and communicational burdens. ECC-based cryptographic systems require shorter key length and less computing power than conventional systems based on discrete logarithm problem (DLP). This is because the algorithms to solve the ECDLP run in a fully exponential time, while the sub-exponential time algorithms are available to address the discrete logarithm problem. The following table gives approximate comparable key sizes for symmetric- and asymmetric-key crypto systems based on the best-known algorithms for attacking them [15].

Symmetric	ECC	DH/DSA/RSA
112	224- 255	2048
128	256-383	3072
192	384-511	7680
256	511+	15360

Table 1: Comparable Key Sizes (in bits)

As shown in Table 1, compared to currently prevalent crypto systems such as RSA, ECC offers equivalent security with smaller key sizes. Smaller key sizes result in savings for power, memory, bandwidth, and computational cost that make ECC especially attractive for constrained environments, such as wireless environments. Another advantage of ECC is that some elliptic curves such as Curve25519 and Curve448[22]are resistant to a wide range of side-channel attacks, since they use constant-time implementation and an exception-free scalar multiplication.

1.1 Terminology

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 RFC 2119 [RFC2119].

2 EC-SRP5 Protocol in SIP Authentication

2.1 Notation

The terms used in the document are listed as follows:

ECI: elliptic curve index
G: a base point (xG, yG) on an elliptic curve
s:salt
Tc: client's temporary private key
Ts: server's temporary private key
Wc: client's public key
Ws: server's public key
Cc: client's confirmation value
Cs: server's confirmation value
Pw: password
v: password verifier
Z: shared secret between client and server
SIP-URI: Uniform Resource Identifier for SIP
         containing user name and domain name

The | symbol denotes string concatenation, the * operator is the scalar point multiplication operation in an EC group, and the . operator is the integer multiplication.

2.2 Password Verifier

The password verifier is computed based on the salt s, uniform resource identifier SIP-URI, password Pw, and elliptic curve index ECI. The SHA-256 hash algorithm[18] is used as the hash function.

```
i=OS2IP(SHA-256(s|SHA-256(SIP-URI|":"|Pw|ECI)))
v=i*G
```

where OS2IP means octet string to integer conversion primitive, the derived password verifier v is actually a point on the elliptic curve indicated by the ECI.

The server then stores the following information in the database for each user:

o SIP-URI
o salt s
o elliptic curve index ECI
o password verifier v

2.3 Protocol Overview

The following flows describe the EC-SRP5 based SIP authentication mechanism at a high-level. The four messages are exchanged between the client and the server during the authentication procedure, which are Initial Request, Response, Request, and Confirmation, respectively.

```
   Client                                                       Server

     | 1) Initial Request                                        |
     |       SIP-URI                                             |
     |---------------------------------------------------------->|
     |                                                           |
     |                             +-------------------------------+
     |                             |lookup  (SIP-URI,v,s, ECI),   |
     |                             |generate  a password-entangled|
     |                             |public key Ws.                |
     |                             +-------------------------------+
     |                                                           |
     |                                                           |
     |                 2)Response                                |
     |                               s, ECI, Ws                  |
     |<----------------------------------------------------------|
     |                                                           |
 +-------------------------------------+                         |
 | generate  public key Wc,            |                         |
 | compute  the shared secret Z,       |                         |
 | compute  the confirmation value Cc.|                         |
 +-------------------------------------+                         |
     |                                                           |
     | 3) Request                                                |
     |             Wc, Cc                                        |
     |---------------------------------------------------------->|
     |                                                           |
     |                         +------------------------------------+
     |                         |compute  the shared secret Z,      |
     |                         |verify   the value Cc,             |
     |                         |compute  the confirmation value Cs|
     |                         +------------------------------------+
     |                                                           |
     |               4) Confirmation                             |
     |                                Cs                         |
     |<----------------------------------------------------------|
     |                                                           |
 +--------------------------+
 | verify    the value Cs  |
 +--------------------------+
```

Fig.1 SIP Authentication procedure based on EC-SRP5 protocol

2.3.1 Initial Request

Authentication is initiated by the client to send the Initial Request message to the server, which contains the SIP-URI.

2.3.2 Response

When receiving the Initial Request from the client, the server lookups the database using the SIP-URI as the key for search, and fetches the password verifier v, salt s, and elliptic curve index ECI associated with the SIP-URI. The server generates a temporary private key Ts by randomly selecting an integer in the range [1, r-1], where r is the order of the the base point G. After that, a password-entangled public key Ws is computed by using the algorithm ECPEPKGP-SRP5-SERVER (Ts, v), which is specified in Appendix A. The server then responds the Initial Request with the Response message containing the salt s, elliptic curve index ECI, and the public key Ws.

2.3.3 Request

When obtaining the Response message from the server, the client generates a temporary private key Tc by randomly choosing an integer from the interval [1, r-1]. The client’s public key is created by calculating Wc= Tc*G. Then client checks the password-entangled public key Ws to validate whether it is a non-identity element of the parent group. This serves to prevent the simple substitution attacks[20]. If it is not true, the client MUST stop the authentication process. Otherwise the client derives the shared secret Z using the algorithm ECSVDP-SRP5-CLIENT (Tc, Pw, Wc, Ws,s) specified in Appendix B. The password verifier v is created as specified in Section 2.2 by using the password Pw, salt s, and elliptic curve index ECI. Then the client creates the confirmation value Cc by computing Cc=SHA-256(hex(04), Wc, Ws, Z, v), in order to confirm the possession of the shared secret Z to the server. Then the client acknowledges the server with the Request message containing the public key Wc and confirmation value Cc.

2.3.4 Confirmation

When receiving the Request message from the client, the server verifies whether the public key Wc is a non-identity element of the parent group. If it is false, the server MUST abort the authentication. Otherwise the server computes the shared secret Z by applying the algorithm ECSDVP-SRP5-SERVER(Ts, v, Wc, Ws),which is detailed in Appendix C. Then the server calculates the expected confirmation value Cc’ with respect to the client using Cc’=SHA-256 (hex(04), Wc, Ws, Z, v). If the expected confirmation value Cc’ is not identical to the received confirmation value Cc, the server MUST terminates the authentication process. Otherwise the client is successfully authenticated by the server. Then server generated the confirmation value Cs by computing Cs=SHA-256 (hex(03), Wc, Ws, Z, v), and sends it to the client. This serves to confirm the possession of the shared secret Z to the client.

Once obtaining the Confirmation message from the server, the client computes the expected confirmation value Cs’ using Cs’=SHA-256(hex(03), Wc, Ws, Z, v) to verify the received confirmation value Cs. If the expected confirmation value Cs’ is not identical to the received confirmation value Cs, the client MUST abort the authentication. Otherwise the client authenticates the server successfully. At this point, the client and the server have completed the mutual authentication.

3 Security Considerations

3.1 Off-line dictionary attack resistance

The messages exchanged in the protocol are usually available to an eavesdropper. The message related to the password information is just the server’s password-entangled public key Ws. An attacker, however, is not able to derive the password from the message Ws. This is because Ws is the addition of the two points in the group, i.e. Ws=TS*G+e1, see Appendix A. Password verifier is used as input selector value to choose a pseudo-random element e1 of a group. The element e1 is shadowed by adding the point Ts*G, which has high-grade entropy as Ts is the order of base point G which usually exceeds 192bits long. In this way, an attacker can not access the sensitive password information from the eavesdropped message Ws.

3.2 On-line dictionary attack resistance

An adversary launches on-line dictionary attacks by running the protocol with an honest party using a guessed password. Each time the protocol abandons the active adversary can eliminate one password. The attack itself is not a great threat to the use of the protocol, since this attack is trivial to detect in the sever by checking the confirmation value Cc. To prevent an attacker from guessing more passwords, the server usually blocks the user authentication when the times of authentication failure reach the default value set in advance.

3.3 Man-in-the middle attack resistance

The man-in-the middle (MITM) attack is that an adversary replaces the exchanged public keys Ws and Wc with its own public keys Ws’ and Wc’ in the middle, respectively. The object of the MITM attack is to fool the client and the server to believe that they communicate with each other using the shared secret Z. Actually the client talks to the adversary with the shared secret Z’, and the server talks to the adversary with the shared secret Z’’. The EC-SRP5 protocol thwarts the MITM attacks by generating and verifying the confirmation value Cc and Cs in the client’s side and server’s side, respectively. In both client and server, the received confirmation value will not equal to the computed confirmation value when the MITM attack is launched, because the client and server do not have the shared secret Z.

3.4 Replay attack resistance The replay attack is that an adversary simply takes a previously sent message, and resends it later in an attempt to gain access to a network or resource. Provided that an adversary replays the messages sent by the server before, which are the Response message (s, ECI, Ws) and the Confirmation message (Cs). The shared secret Z is computed based on the client’s temporary private key Tc. In each authentication process the client will randomly generate a private key, which is completely different from the private keys used in past authentication processes. This implies that each authentication session has its unique shared secret Z. The confirmation value Cs thus will vary with the authentication session. As a result, the client can detect the replay attack by comparing Cs with the expected confirmation value Cs’.

4 Elliptic Curve Index

It is RECOMMEMDED that the following elliptic curves are used, which are specified in [16] as well as in [21].

Description	ECI
secp224k1	1.3.132.0.32
secp224r1	1.3.132.0.33
secp256k1	1.3.132.0.10
secp256r1	1.2.840.10045.3.1.7
secp384r1	1.3.132.0.34
secp521k1	1.3.132.0.35
brainpoolP256r1	1.3.132.0.26
brainpoolP384r1	1.3.132.0.27
brainpoolP512r1	1.3.132.0.28

The elliptic curve identifier (ECI) is the string in the second column of the table, the ASCII representation of the object identifier (OID) of the curve.

5. Acknowledgments

The authors would like to thank Paul Kyzivat for his insight reviews and constructive suggestions, thanks also go to Xiaojun Zhuang, Ivy Guo, and Cathy Wang for their useful comments and suggestions.

6 References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[2] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart: HTTP Authentication: Basic and Digest Access Authentication. IETF RFC 2617, June 1999.

[3] T. Wu: A Real-World Analysis of Kerberos Password Security. Proceedings of the ISOC Symposium on Network and Distributed System Security, 1999.

[4] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A.Nabb : Electronic Authentication Guideline. NIST Special publication 800-63-2, August 2013.

[5] R. Shekh-Yusef: Key-Derivation Authentication Scheme. IETF draft draft-yusef-sipcore-key-derivation-00, October 10, 2014.

[6] H. Krawczyk, M. Bellare, and R. Canetti: HMAC: Keyed-Hashing for Message Authentication, RFC 2104, Febr. 1997.

[7] S. Bellovin and M. Merritt: Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks. Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1992.

[8] P. MacKenzie: The PAK Suite: Protocols for Password-Authenticated Key Exchange. DIMACS Technical Report 002-46, October 2002.

[9] D. Jablon: Strong Password-Only Authenticated Key Exchange. Computer Communication Review, ACM SIGCOMM 26 (1996) 5: 5-26.

[10] T. Kwon: Authentication and Key Agreement via Memorable Password. NDSS 2001 Symposium Conference Proceedings, February 2001.

[11] T. Wu: The Secure Remote Password Protocol. Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium, San Diego, March 1998, pp. 97-111.

[12] Wang, Y., "IEEE P1363.2 Submission / D2001-06-29," A contribution by Yongge Wang for P1363.2 giving an elliptic curve version of the SRP protocol, June 29, 2001.

[13] IEEE P1363.2: Password-Based Public-Key Cryptography. September 2008.

[14] D. Hankerson, A. Menezes, S. Vanstone: Guide to Elliptic Curve Cryptography. Springer, 2003.

[15] NIST: Recommendation on Key Management, SP 800-57,August 2005.

[16] SEC 2: Recommended Elliptic Curve Domain Parameters. Version 2.0, Jan. 2010.

[17] IEEE 1363-2000:IEEE Standard Specifications for Public-Key Cryptography.

[18] FIPS PUB 180-4: Secure Hash Standard. March 2012.

[19] J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart: An Extension to HTTP : Digest Access Authentication. IETF RFC 2069, January 1997.

[20] J. F. Raymond1 and A. Stiglic: Security Issues in the Diffie-Hellman Key Agreement Protocol. http://crypto.cs.mcgill.ca/˜stiglic/

[21] J. Merkle and M. Lochter: Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS). RFC 7027, October 2013.

[22] A. Langley, M. Hamburg and S. Turner: Elliptic Curves for Security. RFC 7748, January 2016.

Authors' Addresses

Fuwen Liu
China Mobile
32 Xuanwumenxi Ave,Xicheng District
Beijing 100032
China

Email: liufuwen@chinamobile.com

Minpeng Qi
China Mobile
32 Xuanwumenxi Ave,Xicheng District
Beijing 100032
China

Email: qiminpeng@chinamobile.com

Min Zuo
China Mobile
32 Xuanwumenxi Ave, Xicheng District
Beijing 100032
China

Email: zuomin@chinamobile.com

Appendix A: Algorithm ECPEPKGP-SRP5-SERVER

ECPEPKGP-SRP5-SERVER is Elliptic Curve Password-Entangled Public Key Generation Primitive for server. The algorithm ECPEPKGP-SRP5-SERVER (Ts, v) is used to generate a elliptic curve password-entangled public key Ws, where the inputs are server’s temporary private key Ts and password verifier v. The following steps are needed to compute the public key Ws:

(1) Compute octet string o1=GE2OSP-X(v)
(2) Compute group element e1=ECREDP(o1)
(3) Compute group element Ws=Ts*G+e1
(4) Output Ws as the password-entangled public key

Where GE2OSP-X is used to convert group elements into octet strings. ECREDP is Elliptic Curve Random Element Derivation Primitive [17]. The primitive uses a hash function of a password-based input selector value to select a pseudo-random element of a group to be used in the password-based authenticated key agreement scheme, in order to prevent collisions and obscure exponential relationships of output values.

Appendix B: Algorithm ECSVDP-SRP5-CLIENT

ECSVDP-SRP5-CLIENT is Elliptic Curve Password-Entangled Secret Value Derivation Primitive for client. The algorithm ECSVDP-SRP5-CLIENT (Tc, Pw, Wc, Ws, s) derives a shared secret value Z from the temporary private key Tc, password Pw, the client’s public key Wc, server’s password entangled public key Ws, and salt s. It has the following sequence of steps:

(1) Compute octet string o1=GE2OSP-X(Wc)
(2) Compute octet string o2=GE2OSP-X(Ws)
(3) Compute octet string o3=SHA-256(o1|o2)
(4) compute an integer i2=OS2IP(o3)
(5) Compute octet string o4=GE2OSP-X(v)
(6) Compute group element e1=ECREDP(o4)
(7) Compute group element e2=Ws-e1
(8) Compute i3=OS2IP(SHA-256(s|SHA-256(SIP-URI|":"|Pw|ECI)))
(9) Compute group element zg= (Tc+(i2.i3))*e2
(10) Compute field element z= GE2SVFEP (zg)
(11) Compute shared secret value Z=FE2OSP (z)
(12) Output Z

Where GE2SVFEP is the primitive for group element to secret value field element conversion, FE2OSP is field element to octet string conversion primitive.

Appendix C: Algorithm ECSDVP-SRP5-SERVER

ECSDVP-SRP5-SERVER is Elliptic Curve Password-Entangled Secret Value Derivation Primitive for server. The algorithm ECSDVP-SRP5-SERVER (Ts, v, Wc, Ws) derives a shared secret value Z from the temporary private key Ts, password verifier v, the client’s public key Wc, and server’s password entangled public key Ws. It has the following sequence of steps:

(1) Compute octet string o1=GE2OSP-X(Wc)
(2) Compute octet string o2=GE2OSP-X(Ws)
(3) Compute octet string o3=SHA-256(o1|o2)
(4) compute an integer i2=OS2IP(o3)
(5) Compute group element zg= Ts*(Wc+i2*v))
(6) Compute field element z= GE2SVFEP (zg)
(7) Compute shared secret value Z=FE2OSP (z)
(8) Output Z