Workgroup: Internet Engineering Task Force
Internet-Draft: draft-leggett-spkac-00
Published: 6 March 2020
Intended Status: Informational
Expires: 7 September 2020
Authors: G. Leggett, Ed. D.W. van Gulik
Pepperpot Media WebWeaving Internet Engineering

Signed Public Key and Challenge

Abstract

This memo describes the Signed Public Key and Challenge (SPKAC), a syntax to provide Proof-of-Possession of a Public Key to support federated (client) certificate enrolment.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on 7 September 2020.

Table of Contents

1. Introduction

During a certificate enrollment process between a client (browser) and a certificate authority, the certificate authority requires that the client provide proof-of-possession of the public key of the certificate that will be signed by the certificate authority.

The Signed Public Key and Challenge consists of a public key and an optional challenge, collectively signed by the private key of the end entity requesting certification.

1.1. Requirements Language

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].

1.2. Historical

The SPKAC protocol was originally used by the Netscape web browser as part of their implementation of what eventually became the HTML5 [W3C.REC-html5-20141028] keygen tag. The keygen tag allowed a web browser to request a (client) certificate from a certificate authority over the world wide web, and the SPKAC protocol ensured the web browser possessed the key being signed by the certificate authority. Storage of the private key would typically be in a file based keystore; or through a PKCS interface on a hardware token (which may, or may not, have generated the private key and signed the SPAC inside that hardware enclave).

For a long time the Signed Public Key and Challenge was a de facto standard widely implemented but not standardised. The purpose of this RFC is to document the existing use of the protocol, address security implementation weaknesses in common implementations, and formalise the protocol into a standard.

Note that, in 2015, Google unilaterally decided to retire keygen tag support from the Chrome web browser. Prior to this; SPKAC was widely used by both centralised certificate authorities (that would issue personal digital x509 certificates) as well as in local enterprise and federated settings. This removal has left the web community with no standard way, de facto or otherwise, to distribute soft and hard tokens to clients.

2. Signed Public Key and Challenge Profile

The parts that make up the Signed Public Key and Challenge are encoded using the ASN.1 distinguished encoding rules (DER) [X.690], and are defined below.

2.1. spki

The spki is a SubjectPublicKeyInfo as defined in RFC 5912 [RFC5912], and consists of an ASN.1 sequence containing the algorithm used by the public key, and the public key itself.

2.2. challenge

The challenge is an ASN.1 IA5String, and MUST consist of a value provided by the certificate authority that is difficult to predict. This value will be encoded into the SPKAC by the end entity, signed by the private key corresponding to the public key, and returned to the certificate authority.

2.3. publicKeyAndChallenge

The publicKeyAndChallenge is an ASN.1 sequence of the spki and challenge defined above. This value is signed using the signatureAlgorithm and public key to produce the signature below.

2.4. signatureAlgorithm

The signatureAlgorithm is an AlgorithmIdentifier defined in RFC 5911 [RFC5911], and represents the algorithm used to sign the publicKeyAndChallenge.

2.5. signature

The signature is an ASN.1 bit string containing the signature of the ASN.1 DER encoded publicKeyAndChallenge, using the algorithm specified by signatureAlgorithm.

3. ASN.1 Module SPKAC

This appendix includes all of the ASN.1 type and value definitions contained in this document in the form of the ASN.1 module SPKAC.

```
SPKAC-Schema DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
  IMPORTS

  AlgorithmIdentifier{}, SIGNATURE-ALGORITHM
  FROM AlgorithmInformation-2009
      {iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-algorithmInformation-02(58)}

  SubjectPublicKeyInfo, SignatureAlgorithms
  FROM PKIX1Explicit-2009
      {iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkix1-explicit-02(51)};

  PublicKeyAndChallenge ::= SEQUENCE
  {
    spki SubjectPublicKeyInfo,
    challenge IA5String
  }
  SignedPublicKeyAndChallenge ::= SEQUENCE
  {
    publicKeyAndChallenge PublicKeyAndChallenge,
    signatureAlgorithm AlgorithmIdentifier{SIGNATURE-ALGORITHM,
                                {SignatureAlgorithms}},
    signature BIT STRING
  }

END
```

4. Example

The following example consists of a Base64 [RFC4648] encoded SPKAC message signed with an RSA key [RFC8017] using the SHA256 message-digest [RFC4634] algorithm.

MIIESTCCAjEwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC9wmyQidBwnIN3
08UwRlTX2mB9g1a05IR6l9TOGmSz6BV5YYbavXdq38EA7dw/44U/P9edRNIwFTMPLWIb
hkDYMZkuziyovhBji8c5gUb09Flplc1aWO8V4u5kjEY6clusYVSxL+j0GU3NXW2k2AC4
+Ts8Y/sE9kfqWb5QzTI2Tz1UqRr6oE2G65LhFhgkg/yafnvOsh+B8nNwdcPwbqzPW8qN
FrDqppDg9sm5PbrfGY9xTeBYKq0AVv//qyG5YgqjNZPIPC4mRQfx/IPbcMJXNIf6iRQi
LniAxFuBmO3Ohog8rufUezWrA5d3f1sorTozkKxRECwnzgMcKmzcyZtENbkrNuL1BbDq
hwDs3xf9ilbwkiRx1BOWrPCEXZneS96iFRkEMkw2AcIQIqiDNSStpOt8jyKX4sRdUO3t
fagCZ1QcL2Dmyab5aypTr/eVx/xj3sZsrQCy89BlOsSN9GTBbIQeui97PiA6hmVFUWbG
La2TdZH+Bnjjc8OiZPF5YH9uQo1Z9xD+fxcL6xOQW35/JyJ7AtGa1SIut0q3WQ4S93wA
B/DOADc9GWl4mWB/2goLDsCUWXSHYs2czS/2a/Lh9MKQsDKLQbEMncxaq3TVBG7Urq9S
RnmemguaWjvoni0o725nWi/j5H6AtOZYoFGmccjxraSAVGWS4r+0X3qSZQIDAQABFglj
aGFsbGVuZ2UwDQYJKoZIhvcNAQELBQADggIBAJJgmBFMD+2AqkQpD/2AgCcoKNkRmcD8
EkHQY+5WN80+ogaWf5VcDU3ycP2554x95EPLhclqrX9xbzUemuUoNiR/sPyhxl0Pr7OP
tKqulW6QvT+YCcyrbILR1jE8lhVSENkT/fL6U9J4NPd7qGB0OFiTv9tAT1huzzuXgx67
6T8Y5mb9XVk0C6CCGEoDxSKI2n3/nbkdyyXlq1uFphwVCXEBUvrndD8y3vKd8rhtGyvz
8cTg2q/mHmSHldwwmfksaRNwh9mxOKerLUQ5pFM68HODOnJHFs/D26GQlwINfVqrVnI+
oCA/VFFz/QO4minT7zuDSGa/cFdiPWj3d//Gz02ppUIHk8RVKrdGgTf/efQmbP2zLEfa
AfTUlSjVliVDqw5SRG6QJYrvz80pfZcz13BY3pkN5lnAcuA8Ld5Gb/YVfiJkiefvMt9t
753pe9Yxv8iU6PKfQO8UbiGbPfEDP5bQ1EJPX0rdmvX7T85hwR7LXC5iUBs2xdahTfDg
oZTZ/12fSoNwkdgmYURmy/fAEOnVHIn5Gj/LKu8ii2UOzWktbAnz4f30MeuFeaBx5h9v
e/nELQnvsPiZgIDFdKYdXb8yJRTgg9ahYdPhEC/u1RIJFxs4sRmRfZwY7qATssLhnL9Z
DtDuuZxJft+sn5swpiepSiekGvw20fsP6tRD4nu0

The following section shows the decoded version of the above SKPAC message.

```
Netscape SPKI:
  Public Key Algorithm: rsaEncryption
    RSA Public-Key: (4096 bit)
    Modulus:
        00:bd:c2:6c:90:89:d0:70:9c:83:77:d3:c5:30:46:
        54:d7:da:60:7d:83:56:b4:e4:84:7a:97:d4:ce:1a:
        64:b3:e8:15:79:61:86:da:bd:77:6a:df:c1:00:ed:
        dc:3f:e3:85:3f:3f:d7:9d:44:d2:30:15:33:0f:2d:
        62:1b:86:40:d8:31:99:2e:ce:2c:a8:be:10:63:8b:
        c7:39:81:46:f4:f4:59:69:95:cd:5a:58:ef:15:e2:
        ee:64:8c:46:3a:72:5b:ac:61:54:b1:2f:e8:f4:19:
        4d:cd:5d:6d:a4:d8:00:b8:f9:3b:3c:63:fb:04:f6:
        47:ea:59:be:50:cd:32:36:4f:3d:54:a9:1a:fa:a0:
        4d:86:eb:92:e1:16:18:24:83:fc:9a:7e:7b:ce:b2:
        1f:81:f2:73:70:75:c3:f0:6e:ac:cf:5b:ca:8d:16:
        b0:ea:a6:90:e0:f6:c9:b9:3d:ba:df:19:8f:71:4d:
        e0:58:2a:ad:00:56:ff:ff:ab:21:b9:62:0a:a3:35:
        93:c8:3c:2e:26:45:07:f1:fc:83:db:70:c2:57:34:
        87:fa:89:14:22:2e:78:80:c4:5b:81:98:ed:ce:86:
        88:3c:ae:e7:d4:7b:35:ab:03:97:77:7f:5b:28:ad:
        3a:33:90:ac:51:10:2c:27:ce:03:1c:2a:6c:dc:c9:
        9b:44:35:b9:2b:36:e2:f5:05:b0:ea:87:00:ec:df:
        17:fd:8a:56:f0:92:24:71:d4:13:96:ac:f0:84:5d:
        99:de:4b:de:a2:15:19:04:32:4c:36:01:c2:10:22:
        a8:83:35:24:ad:a4:eb:7c:8f:22:97:e2:c4:5d:50:
        ed:ed:7d:a8:02:67:54:1c:2f:60:e6:c9:a6:f9:6b:
        2a:53:af:f7:95:c7:fc:63:de:c6:6c:ad:00:b2:f3:
        d0:65:3a:c4:8d:f4:64:c1:6c:84:1e:ba:2f:7b:3e:
        20:3a:86:65:45:51:66:c6:2d:ad:93:75:91:fe:06:
        78:e3:73:c3:a2:64:f1:79:60:7f:6e:42:8d:59:f7:
        10:fe:7f:17:0b:eb:13:90:5b:7e:7f:27:22:7b:02:
        d1:9a:d5:22:2e:b7:4a:b7:59:0e:12:f7:7c:00:07:
        f0:ce:00:37:3d:19:69:78:99:60:7f:da:0a:0b:0e:
        c0:94:59:74:87:62:cd:9c:cd:2f:f6:6b:f2:e1:f4:
        c2:90:b0:32:8b:41:b1:0c:9d:cc:5a:ab:74:d5:04:
        6e:d4:ae:af:52:46:79:9e:9a:0b:9a:5a:3b:e8:9e:
        2d:28:ef:6e:67:5a:2f:e3:e4:7e:80:b4:e6:58:a0:
        51:a6:71:c8:f1:ad:a4:80:54:65:92:e2:bf:b4:5f:
        7a:92:65
    Exponent: 65537 (0x10001)
  Challenge String: challenge
  Signature Algorithm: sha256WithRSAEncryption
      92:60:98:11:4c:0f:ed:80:aa:44:29:0f:fd:80:80:27:28:28:
      d9:11:99:c0:fc:12:41:d0:63:ee:56:37:cd:3e:a2:06:96:7f:
      95:5c:0d:4d:f2:70:fd:b9:e7:8c:7d:e4:43:cb:85:c9:6a:ad:
      7f:71:6f:35:1e:9a:e5:28:36:24:7f:b0:fc:a1:c6:5d:0f:af:
      b3:8f:b4:aa:ae:95:6e:90:bd:3f:98:09:cc:ab:6c:82:d1:d6:
      31:3c:96:15:52:10:d9:13:fd:f2:fa:53:d2:78:34:f7:7b:a8:
      60:74:38:58:93:bf:db:40:4f:58:6e:cf:3b:97:83:1e:bb:e9:
```

3f:18:e6:66:fd:5d:59:34:0b:a0:82:18:4a:03:c5:22:88:da:
7d:ff:9d:b9:1d:cb:25:e5:ab:5b:85:a6:1c:15:09:71:01:52:
fa:e7:74:3f:32:de:f2:9d:f2:b8:6d:1b:2b:f3:f1:c4:e0:da:
af:e6:1e:64:87:95:dc:30:99:f9:2c:69:13:70:87:d9:b1:38:
a7:ab:2d:44:39:a4:53:3a:f0:73:83:3a:72:47:16:cf:c3:db:
a1:90:97:02:0d:7d:5a:ab:56:72:3e:a0:20:3f:54:51:73:fd:
03:b8:9a:29:d3:ef:3b:83:48:66:bf:70:57:62:3d:68:f7:77:
ff:c6:cf:4d:a9:a5:42:07:93:c4:55:2a:b7:46:81:37:ff:79:
f4:26:6c:fd:b3:2c:47:da:01:f4:d4:95:28:d5:96:25:43:ab:
0e:52:44:6e:90:25:8a:ef:cf:cd:29:7d:97:33:d7:70:58:de:
99:0d:e6:59:c0:72:e0:3c:2d:de:46:6f:f6:15:7e:22:64:89:
e7:ef:32:df:6d:ef:9d:e9:7b:d6:31:bf:c8:94:e8:f2:9f:40:
ef:14:6e:21:9b:3d:f1:03:3f:96:d0:d4:42:4f:5f:4a:dd:9a:
f5:fb:4f:ce:61:c1:1e:cb:5c:2e:62:50:1b:36:c5:d6:a1:4d:
f0:e0:a1:94:d9:ff:5d:9f:4a:83:70:91:d8:26:61:44:66:cb:
f7:c0:10:e9:d5:1c:89:f9:1a:3f:cb:2a:ef:22:8b:65:0e:cd:
69:2d:6c:09:f3:e1:fd:f4:31:eb:85:79:a0:71:e6:1f:6f:7b:
f9:c4:2d:09:ef:b0:f8:99:80:80:c5:74:a6:1d:5d:bf:32:25:
14:e0:83:d6:a1:61:d3:e1:10:2f:ee:d5:12:09:17:1b:38:b1:
19:91:7d:9c:18:ee:a0:13:b2:c2:e1:9c:bf:59:0e:d0:ee:b9:
9c:49:7e:df:ac:9f:9b:30:a6:27:a9:4a:27:a4:1a:fc:36:d1:
fb:0f:ea:d4:43:e2:7b:b4

5. IANA Considerations

IANA is asked to assign the value "spkac" below { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) } as per https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-26 for the identifier of the ASN.1 SPKAC schema, and to add this to the ASN.1 definition in this specification.

All drafts are required to have an IANA considerations section (see Guidelines for Writing an IANA Considerations Section in RFCs [RFC5226] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

6. Security Considerations

The aim of SPKAC is that no adversary can convince a certificate authority to sign a certificate using the public key other than that intended. An adversary is any entity other than the end entity and the certificate authority attempting to establish proof-of-possession.

6.1. Use of the MD5 Message-Digest Algorithm

Historically the formal definition of the HTML keygen tag specified that the MD5 message-digest algorithm be used within SPKAC requests.

As defined in Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms [RFC6151] MD5 must not be used for digital signatures.

New protocols using the SPKAC protocol MUST NOT mandate the use of a fixed message-digest algorithm, and existing protocols using the SPKAC protocol SHOULD be updated to ensure the message-digest used is not fixed to a given digest.

6.2. Clear Text Challenge and Public Key

Given that both the Challenge and the Public Key are encoded within the SPKAC message in clear text, to ensure privacy of the data in transit additional steps SHOULD be taken to ensure that SPKAC message is delivered over a secure transport, such as TLS [RFC8446].

6.3. UI/UX Denial of Service Design Issues

When the generation of an SPKAC message is triggered by a remote entity, such as a certificate authority triggering the generation of an SPKAC message in a browser as part of a certificate request, the

user interfaces in the client (browser) should take care to not allow (rogue) webpages or javascript to generate a very large number of keygen requests; as this is not only somewhat resource intensive; but may also deplete cryptographic quality random generator pools (historically a concern). This is especially important as most implementations will generally keep the cryptographic code and (private) key storage outside the sandbox in which the DOM and Javascript is handled.

Likewise - clients (browsers) should be particularly careful when handling solicited (and unsolicited and maliciously repeated/high-volume) responses to a SPKAC submission when storing certificates and recombining certificates with keys in the key store. Especially as (historically) it was common for such request to be handled asynchronously; with the user receiving an email after, for example human approval, to pick up the signed certificate at a certain URL.

Clients SHOULD make a request to the user for consent for the client to generate the SPKAC message in a clear and easy to understand manner, with cancel being the default choice should the user not understand the request.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4634] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, DOI 10.17487/RFC4634, July 2006, <https://www.rfc-editor.org/info/rfc4634>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC5911] Hoffman, P. and J. Schaad, "New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S/MIME", RFC 5911, DOI 10.17487/RFC5911, June 2010, <https://www.rfc-editor.org/info/rfc5911>.

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/info/rfc5912>.

[RFC6151] Turner, S., "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, March 2011, <https://www.rfc-editor.org/info/rfc6151>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[W3C.REC-html5-20141028] Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, T., and S. Pfeiffer, "HTML5", World Wide Web Consortium Recommendation REC-html5-20141028, 28 October 2014, <http://www.w3.org/TR/2014/REC-html5-20141028>.

[X.690] authSurName, authInitials., "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).", ITU-T Recommendation X.690 (2002) ISO/IEC 8825-1:2002, 2002.

7.2. Informative References

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

Authors' Addresses

Graham Leggett (editor)
Pepperpot Media
London
United Kingdom

Email: minfrin@sharp.fm

Dirk-Willem van Gulik
WebWeaving Internet Engineering
Leiden
Netherlands

Email: dirkx@webweaving.org