INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: April 5, 2014

M. Peck
The MITRE Corporation

October 2, 2013

Elliptic Curve Diffie-Hellman Proof-of-Possession Method
draft-peck-ecdhpop-01

Abstract

This document specifies a proof-of-possession mechanism for requesting Elliptic Curve Diffie-Hellman X.509 public key certificates using the PKCS #10 and CRMF syntaxes.

Status of this Memo

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

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/1id-abstracts.html

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

Copyright and License Notice

Table of Contents

1. Introduction

PKCS #10 [RFC2986] and the Certificate Request Message Format (CRMF) [RFC4211] are two syntaxes for requesting X.509 public key certificates.

Public Key Infrastructure (PKI) policies commonly require that certificate requesters prove possession of the private key corresponding to the public key in the request.

In the case of a signature certificate request, the requester typically provides a digital signature, computed using the private key corresponding to the public key in the certificate request, as proof of possession.

In the case of a key agreement certificate request, it is possible to provide proof of possession of the private key by performing a key agreement operation. However, doing so adds complexity. A simpler method of providing proof of possession is to perform a one-time digital signature using the private key.

[RFC6955] defines mechanisms to perform proof-of-possession using a key agreement operation for Diffie-Hellman and Elliptic Curve Diffie-Hellman keys, and a mechanism to perform proof-of-possession using a digital signature for Diffie-Hellman keys. However, it does not define a mechanism to perform proof-of-possession using a digital signature algorithm for Elliptic Curve Diffie-Hellman keys.

This document defines a mechanism to perform proof of possession for Elliptic Curve Diffie-Hellman certificate requests using the Elliptic Curve Digital Signature Algorithm (ECDSA). This document also defines how to transmit the proof of possession value using both the PKCS #10 and CRMF syntaxes.

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 [KEYWORDS].

2. Elliptic Curve Diffie-Hellman

When using the proof of possession mechanism specified in this section, Elliptic Curve Diffie-Hellman domain parameters MUST be selected from a named curve. [FIPS186] names fifteen curves, and other documents specify other curves.

PKCS #10 and CRMF both use a data element of type

SubjectPublicKeyInfo to convey the requested public key. The X.509 certificate, if issued, will contain the public key in a field of the same type.

[RFC5480] Section 2.1.1 specifies how to represent the Elliptic Curve Diffie-Hellman domain parameters and public key within the SubjectPublicKeyInfo data element.

The Elliptic Curve Digital Signature Algorithm is specified in [FIPS186]. [RFC6090] specifies a signature algorithm called KT-I that for many hash functions is mathematically and functionally equivalent to ECDSA. An ECDSA digital signature is generated using a set of domain parameters (a curve), a private key, a per-message secret number, a hash function, and the data to be signed.

To use ECDSA to perform proof of possession of an Elliptic Curve Diffie-Hellman private key:

- Set the ECDSA curve to the same as the ECDH curve.
- Set the ECDSA private key to the same as the ECDH private key.
- Set the ECDSA per-message secret according to the guidance of [FIPS186].
- Set the hash function depending on the selected curve. The security strength of the chosen hash function MUST be equal to or greater than the security strength associated with the bit length of the domain parameter n. The security considerations section of

[RFC5480] provides recommended hash functions to use in conjunction with the NIST curves.

o Set the data to be signed according to the guidance provided by PKCS #10 or CRMF.

For PKCS #10, set the signatureAlgorithm field to the appropriate ECDSA signature algorithm object identifier depending on the hash function chosen above, and place the generated ECDSA signature in the signature field, following the guidance of [RFC3279] Section 2.2.3.

For CRMF, include the popo field within CertReqMsg using the signature (POPOSigningKey) proof-of-possession choice. Set the POPOSigningKey algorithmIdentifier to the appropriate ECDSA signature algorithm object identifier depending on the hash function chosen above, and place the generated ECDSA signature in the POPOSigningKey signature field, following the guidance of [RFC3279] Section 2.2.3.

[RFC3279] and [RFC5758] provide ECDSA signature algorithm identifiers

paired with various hash functions. Additional ECDSA signature algorithm identifiers may be found in other sources.

3. Security Considerations

This document specifies proof-of-possession of a key agreement private key by performing a digital signature. Except for one-time proof-of-possession, a single key pair SHOULD NOT be used for both signature and key agreement. NIST Special Publication 800-57 Part 1 [SP80057] Section 8.1.5.1.1.2 permits use of a key agreement private key to perform a digital signature for proof-of-possession purposes.

This specification requires implementations to generate key pairs and other random values. The use of inadequate pseudo-random number generators (PRNGs) can result in little or no security. The generation of quality random numbers is difficult. [FIPS186] and [RFC4086] offer random number generation guidance.

In a substitution attack, as described in [RFC5272] Section 6.3, an attacker may attempt to take a PKCS #10 or CRMF certificate request and change the context in which it is presented to the Certification Authority in order to cause a certificate with incorrect identity

information to be generated. In order to thwart this class of attack, the proof-of-possession signature should cover not only the public key itself but also on the requested identity or other information used by the public key infrastructure to assign an identity to the issued certificate. For example, CMC [RFC5272] provides a mechanism to cryptographically bind information from the outer Full PKI Request into the inner PKCS #10 or CRMF message where it is covered by the proof-of-possession signature. The EST protocol [est] provides a similar mechanism to cryptographically bind information from the TLS session into the inner PKCS #10 or CRMF message where it is covered by the proof-of-possession signature.

4. IANA Considerations

This document has no IANA actions.

5. References

5.1. Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[FIPS186] National Institute of Standards and Technology, FIPS Publication 186-4: "Digital Signature Standard (DSS)", July 2013.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, November 2000.

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, September 2005.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, March 2009.

5.2. Informative References

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, June 2008.

[RFC5758] Dang, Q., Santesson, S., Moriarty, K., Brown, D., and T. Polk, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", RFC 5758, January 2010.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, February 2011.

[RFC6955] Schaad, J., and H. Prafullchandra, "Diffie-Hellman Proof-of-Possession Algorithms", RFC 6955, May 2013.

[SP80057] National Institute of Standards and Technology, Special Publication 800-57 Part 1: "Recommendation for Key Management - Part 1: General (Revision 3)", July 2012.

[est] Pritikin, M., Yee, P., and D. Harkins, "Enrollment over Secure Transport", draft-ietf-pkix-est-09, Work in Progress, August 2013.

Authors' Addresses

Michael Peck
The MITRE Corporation

EMail: mpeck@mitre.org