

Elliptic Curve Private Key Structure

Abstract

This document specifies the syntax and semantics for conveying Elliptic Curve (EC) private key information. The syntax and semantics defined herein are based on similar syntax and semantics defined by the Standards for Efficient Cryptography Group (SECG).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc5915>.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

This document specifies a syntax and semantics for Elliptic Curve (EC) private key information. EC private key information includes a private key and parameters. Additionally, it may include the corresponding public key. The syntax and semantics defined herein are based on similar syntax and semantics defined by the Standards for Efficient Cryptography Group (SECG) [SECG1].

Most Public Key Infrastructures (PKIs) mandate local key generation; however, there are some PKIs that also support centralized key generation (e.g., the public-private key pair is generated by a Certification Authority). The structure defined in this document allows the entity that generates the private and public keys to distribute the key pair and the associated domain parameters.

This syntax is useful when distributing EC private keys using PrivateKeyInfo, as defined in PKCS #8 [RFC5208]. Distributing an EC private key with PKCS#8 [RFC5208] involves including:

- a) id-ecPublicKey, id-ecDH, or id-ecMQV (from [RFC5480]) with the namedCurve as the parameters in the privateKeyAlgorithm field; and
- b) ECPrivateKey in the PrivateKey field, which is an OCTET STRING.

When an EC public key is included in the distributed PrivateKeyInfo, the publicKey field in ECPrivateKey is used.

2. 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 [RFC2119].

3. Elliptic Curve Private Key Format

This section gives the syntax for an EC private key. Computationally, an EC private key is an unsigned integer, but for representation, EC private key information SHALL have ASN.1 type ECPrivateKey:

```
ECPrivateKey ::= SEQUENCE {  
    version          INTEGER { ecPrivkeyVer1(1) } (ecPrivkeyVer1),  
    privateKey       OCTET STRING,  
    parameters [0] ECPParameters {{ NamedCurve }} OPTIONAL,  
    publicKey  [1] BIT STRING OPTIONAL  
}
```

The fields of type `ECPrivateKey` have the following meanings:

- o `version` specifies the syntax version number of the elliptic curve private key structure. For this version of the document, it SHALL be set to `ecPrivkeyVer1`, which is of type `INTEGER` and whose value is one (1).
- o `privateKey` is the private key. It is an octet string of length ceiling ($\log_2(n)/8$) (where n is the order of the curve) obtained from the unsigned integer via the Integer-to-Octet-String-Primitive (`I2OSP`) defined in [RFC3447].
- o `parameters` specifies the elliptic curve domain parameters associated to the private key. The type `ECPParameters` is discussed in [RFC5480]. As specified in [RFC5480], only the `namedCurve` CHOICE is permitted. `namedCurve` is an object identifier that fully identifies the required values for a particular set of elliptic curve domain parameters. Though the ASN.1 indicates that the `parameters` field is OPTIONAL, implementations that conform to this document MUST always include the `parameters` field.
- o `publicKey` contains the elliptic curve public key associated with the private key in question. The format of the public key is specified in Section 2.2 of [RFC5480]. Though the ASN.1 indicates `publicKey` is OPTIONAL, implementations that conform to this document SHOULD always include the `publicKey` field. The `publicKey` field can be omitted when the public key has been distributed via another mechanism, which is beyond the scope of this document. Given the private key and the parameters, the public key can always be recomputed; this field exists as a convenience to the consumer.

4. Other Considerations

When generating a transfer encoding, generators SHOULD use Distinguished Encoding Rules (DER) [X.690] and receivers SHOULD be prepared to handle Basic Encoding Rules (BER) [X.690] and DER [X.690].

Section 1 described a format for transporting EC private keys (i.e., converting `ECPrivateKey` to `PrivateKeyInfo` [PKCS#8]); however, this format can also be used for local storage.

Local storage of an unencrypted `ECPrivateKey` object is out of scope of this document. However, one popular format uses the `.pem` file extension. It is the PEM encoding, which is the Base64 encoding (see Section 4 of [RFC4648]), of the DER-encoded `ECPrivateKey` object that is sandwiched between:

```
-----BEGIN EC PRIVATE KEY-----  
-----END EC PRIVATE KEY-----
```

Another local storage format uses the .der file extension. In this case, it is a DER [X.690] encoding of the ECPrivateKey object.

Local storage of an encrypted ECPrivateKey object is out of scope of this document. However, ECPrivateKey should be the format for the plaintext key being encrypted. DER [X.690] encoding the ECPrivateKey will promote interoperability if the key is encrypted for transport to another party. PEM encoding the DER-encoded ECPrivateKey is common; "Proc-Type:" and "DEK-INFO:" fields [RFC1421] followed by the DER-encoded ECPrivateKey are sandwiched between:

```
-----BEGIN EC PRIVATE KEY-----  
-----END EC PRIVATE KEY-----
```

5. Security Considerations

This structure does not protect the EC private key information in any way. This structure should be combined with a security protocol to protect it.

Protection of the private key information is vital to public key cryptography. The consequences of disclosure depend on the purpose of the private key. If a private key is used for signature, then the disclosure allows unauthorized signing. If a private key is used for key management, then disclosure allows unauthorized parties to access the managed keying material. The encryption algorithm used in the encryption process must be as 'strong' as the key it is protecting.

6. References

6.1. Normative References

- [RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, February 1993.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

- [RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", [RFC 5480](#), March 2009.
- [RFC5912] Schaad, J. and P. Hoffman, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)" [RFC 5912](#), June 2010.
- [SECG1] Standards for Efficient Cryptography Group (SECG), "SEC 1: Elliptic Curve Cryptography", Version 2.0, May 2009.
- [X.680] ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, Information Technology - Abstract Syntax Notation One.
- [X.681] ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002, Information Technology - Abstract Syntax Notation One: Information Object Specification.
- [X.682] ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002, Information Technology - Abstract Syntax Notation One: Constraint Specification.
- [X.683] ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002, Information Technology - Abstract Syntax Notation One: Parameterization of ASN.1 Specifications, 2002.
- [X.690] ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

7.2. Informative References

- [RFC5208] Kaliski, B., "Public-Key Cryptography Standards (PKCS) #8: Private-Key Information Syntax Specification Version 1.2", [RFC 5208](#), May 2008.

Appendix A. ASN.1 Module

This appendix provides ASN.1 definitions for the structures described in this specification using ASN.1 as defined in [X.680], [X.681], [X.682], and [X.683] for compilers that support the 2002 ASN.1.

```
ECPrivateKey { iso(1) identified-organization(3) dod(6)
  internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
  id-mod-ecprivateKey(65) }

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL;

IMPORTS

-- FROM New PKIX ASN.1 [RFC5912]

ECPParameters, NamedCurve
  FROM PKIXAlgs-2009
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-pkix1-algorithms2008-02(56) }

;

ECPrivateKey ::= SEQUENCE {
  version          INTEGER { ecPrivkeyVer1(1) } (ecPrivkeyVer1),
  privateKey       OCTET STRING,
  parameters [0] ECPParameters {{ NamedCurve }} OPTIONAL,
  publicKey  [1] BIT STRING OPTIONAL
}

END
```

Appendix B. Differences with SECG1

This appendix lists the differences between this document and [SECG1]:

1. This document uses the I2OSP routine defined in [RFC3447] while SECG1 defines its own routine. The two routines result in the same output.

2. SECG1 constrains its parameters (i.e., the curves) to SECGCurveNames. This document constrains the parameters to NamedCurve from [RFC5480].
3. This document requires parameters be present while SECG1 does not.
4. This document specifies requirements for encoding rules while SECG1 did not.

Acknowledgements

The authors would like to thank Simon Blake-Wilson and John O. Goyo for their work on defining the structure in [SECG1]. The authors would also like to thank Pasi Eronen, Alfred Hoenes, Joel Jaeggli, Avshalom Houri, Russ Housley, Jim Schaad, and Carl Wallace for their comments.

Authors' Addresses

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

E-Mail: turners@ieca.com

Daniel R. L. Brown
Certicom Corp
5520 Explorer Drive #400
Mississauga, ON L4W 5L1
Canada

E-Mail: dbrown@certicom.com