Network Working Group
Request for Comments: 5349
Category: Informational
September 2008

L. Zhu
K. Jaganathan
K. Lauter
Microsoft Corporation

# Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)

## Status of This Memo



## Abstract

This document describes the use of Elliptic Curve certificates, Elliptic Curve signature schemes and Elliptic Curve Diffie-Hellman (ECDH) key agreement within the framework of PKINIT -- the Kerberos Version 5 extension that provides for the use of public key cryptography.

## Table of Contents

## 1. Introduction

Elliptic Curve Cryptography (ECC) is emerging as an attractive public-key cryptosystem that provides security equivalent to currently popular public-key mechanisms such as RSA and DSA with smaller key sizes [LENSTRA] [NISTSP80057].

Currently, [RFC4556] permits the use of ECC algorithms but it does not specify how ECC parameters are chosen or how to derive the shared key for key delivery using Elliptic Curve Diffie-Hellman (ECDH) [IEEE1363] [X9.63].

This document describes how to use Elliptic Curve certificates, Elliptic Curve signature schemes, and ECDH with [RFC4556]. However, it should be noted that there is no syntactic or semantic change to the existing [RFC4556] messages. Both the client and the Key Distribution Center (KDC) contribute one ECDH key pair using the key agreement protocol described in this document.

## 2. Conventions Used in This Document

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. Using Elliptic Curve Certificates and Elliptic Curve Signature Schemes

ECC certificates and signature schemes can be used in the Cryptographic Message Syntax (CMS) [RFC3852] [RFC3278] content type 'SignedData'.

X.509 certificates [RFC5280] that contain ECC public keys or are signed using ECC signature schemes MUST comply with [RFC3279].

The signatureAlgorithm field of the CMS data type 'SignerInfo' can contain one of the following ECC signature algorithm identifiers:

```
ecdsa-with-Sha1   [RFC3279]
ecdsa-with-Sha256 [X9.62]
ecdsa-with-Sha384 [X9.62]
ecdsa-with-Sha512 [X9.62]
```

The corresponding digestAlgorithm field contains one of the following hash algorithm identifiers respectively:

```
id-sha1          [RFC3279]
id-sha256        [X9.62]
id-sha384        [X9.62]
id-sha512        [X9.62]
```

Namely, id-sha1 MUST be used in conjunction with ecdsa-with-Sha1, id-sha256 MUST be used in conjunction with ecdsa-with-Sha256, id-sha384 MUST be used in conjunction with ecdsa-with-Sha384, and id-sha512 MUST be used in conjunction with ecdsa-with-Sha512.

Implementations of this specification MUST support ecdsa-with-Sha256 and SHOULD support ecdsa-with-Sha1.

## 4. Using the ECDH Key Exchange

This section describes how ECDH can be used as the Authentication Service (AS) reply key delivery method [RFC4556]. Note that the protocol description here is similar to that of Modular Exponential Diffie-Hellman (MODP DH), as described in [RFC4556].

If the client wishes to use the ECDH key agreement method, it encodes its ECDH public key value and the key's domain parameters [IEEE1363] [X9.63] in clientPublicValue of the PA-PK-AS-REQ message [RFC4556].

As described in [RFC4556], the ECDH domain parameters for the client's public key are specified in the algorithm field of the type SubjectPublicKeyInfo [RFC3279] and the client's ECDH public key value is mapped to a subjectPublicKey (a BIT STRING) according to [RFC3279].

The following algorithm identifier is used to identify the client's choice of the ECDH key agreement method for key delivery.

```
id-ecPublicKey  (Elliptic Curve Diffie-Hellman [RFC3279])
```

If the domain parameters are not accepted by the KDC, the KDC sends back an error message [RFC4120] with the code KDC_ERR_DH_KEY_PARAMETERS_NOT_ACCEPTED [RFC4556]. This error message contains the list of domain parameters acceptable to the KDC. This list is encoded as TD-DH-PARAMETERS [RFC4556], and it is in the KDC's decreasing preference order. The client can then pick a set of domain parameters from the list and retry the authentication.

Both the client and the KDC MUST have local policy that specifies which set of domain parameters are acceptable if they do not have a priori knowledge of the chosen domain parameters. The need for such local policy is explained in Section 7.

If the ECDH domain parameters are accepted by the KDC, the KDC sends back its ECDH public key value in the subjectPublicKey field of the PA-PK-AS-REP message [RFC4556].

As described in [RFC4556], the KDC's ECDH public key value is encoded as a BIT STRING according to [RFC3279].

Note that in the steps above, the client can indicate to the KDC that it wishes to reuse ECDH keys or it can allow the KDC to do so, by including the clientDHNonce field in the request [RFC4556]; the KDC can then reuse the ECDH keys and include the serverDHNonce field in the reply [RFC4556]. This logic is the same as that of the Modular Exponential Diffie-Hellman key agreement method [RFC4556].

If ECDH is negotiated as the key delivery method, then the PA-PK-AS-REP and AS reply key are generated as in Section 3.2.3.1 of [RFC4556] with the following difference: The ECDH shared secret value (an elliptic curve point) is calculated using operation ECSVDP-DH as described in Section 7.2.1 of [IEEE1363]. The x-coordinate of this point is converted to an octet string using operation FE2OSP as described in Section 5.5.4 of [IEEE1363]. This octet string is the DHSharedSecret.

Both the client and KDC then proceed as described in [RFC4556] and [RFC4120].

Lastly, it should be noted that ECDH can be used with any certificates and signature schemes. However, a significant advantage of using ECDH together with ECC certificates and signature schemes is that the ECC domain parameters in the client certificates or the KDC certificates can be used. This obviates the need of locally preconfigured domain parameters as described in Section 7.

## 5. Choosing the Domain Parameters and the Key Size

The domain parameters and the key size should be chosen so as to provide sufficient cryptographic security [RFC3766]. The following table, based on table 2 on page 63 of NIST SP800-57 part 1 [NISTSP80057], gives approximate comparable key sizes for symmetric- and asymmetric-key cryptosystems based on the best-known algorithms for attacking them.

| Symmetric | ECC | RSA |
|---|---|---|
| 80 | 160 - 223 | 1024 |
| 112 | 224 - 255 | 2048 |
| 128 | 256 - 383 | 3072 |
| 192 | 384 - 511 | 7680 |
| 256 | 512+ | 15360 |

Table 1: Comparable key sizes (in bits)

Thus, for example, when securing a 128-bit symmetric key, it is prudent to use 256-bit Elliptic Curve Cryptography (ECC), e.g., group P-256 (secp256r1) as described below.

A set of ECDH domain parameters is also known as a "curve". A curve is a "named curve" if the domain parameters are well known and can be identified by an Object Identifier; otherwise, it is called a "custom curve". [RFC4556] supports both named curves and custom curves, see Section 7 on the tradeoffs of choosing between named curves and custom curves.

The named curves recommended in this document are also recommended by the National Institute of Standards and Technology (NIST)[FIPS186-2]. These fifteen ECC curves are given in the following table [FIPS186-2] [SEC2].

| Description | | SEC 2 OID |
|---|---|---|
| ECPRGF192Random | group P-192 | secp192r1 |
| EC2NGF163Random | group B-163 | sect163r2 |
| EC2NGF163Koblitz | group K-163 | sect163k1 |
| ECPRGF224Random | group P-224 | secp224r1 |
| EC2NGF233Random | group B-233 | sect233r1 |
| EC2NGF233Koblitz | group K-233 | sect233k1 |
| ECPRGF256Random | group P-256 | secp256r1 |
| EC2NGF283Random | group B-283 | sect283r1 |
| EC2NGF283Koblitz | group K-283 | sect283k1 |
| ECPRGF384Random | group P-384 | secp384r1 |
| EC2NGF409Random | group B-409 | sect409r1 |
| EC2NGF409Koblitz | group K-409 | sect409k1 |
| ECPRGF521Random | group P-521 | secp521r1 |
| EC2NGF571Random | group B-571 | sect571r1 |
| EC2NGF571Koblitz | group K-571 | sect571k1 |

## 6. Interoperability Requirements

Implementations conforming to this specification MUST support curve P-256 and P-384.

## 7. Security Considerations

When using ECDH key agreement, the recipient of an elliptic curve public key should perform the checks described in IEEE P1363, Section A16.10 [IEEE1363]. It is especially important, if the recipient is using a long-term ECDH private key, to check that the sender's public key is a valid point on the correct elliptic curve; otherwise, information may be leaked about the recipient's private key, and iterating the attack will eventually completely expose the recipient's private key.

Kerberos error messages are not integrity protected; as a result, the domain parameters sent by the KDC as TD-DH-PARAMETERS can be tampered with by an attacker so that the set of domain parameters selected could be either weaker or not mutually preferred. Local policy can configure sets of domain parameters that are acceptable locally or can disallow the negotiation of ECDH domain parameters.

Beyond elliptic curve size, the main issue is elliptic curve structure. As a general principle, it is more conservative to use elliptic curves with as little algebraic structure as possible. Thus, random curves are more conservative than special curves (such as Koblitz curves), and curves over F_p with p random are more conservative than curves over F_p with p of a special form. (Also, curves over F_p with p random might be considered more conservative than curves over F_2^m, as there is no choice between multiple fields of similar size for characteristic 2.) Note, however, that algebraic structure can also lead to implementation efficiencies, and implementors and users may, therefore, need to balance conservatism against a need for efficiency. Concrete attacks are known against only very few special classes of curves, such as supersingular curves, and these classes are excluded from the ECC standards such as [IEEE1363] and [X9.62].

Another issue is the potential for catastrophic failures when a single elliptic curve is widely used. In this case, an attack on the elliptic curve might result in the compromise of a large number of keys. Again, this concern may need to be balanced against efficiency and interoperability improvements associated with widely used curves. Substantial additional information on elliptic curve choice can be found in [IEEE1363], [X9.62], and [FIPS186-2].

## 8. Acknowledgements

The following people have made significant contributions to this document: Paul Leach, Dan Simon, Kelvin Yiu, David Cross, Sam Hartman, Tolga Acar, and Stefan Santesson.

## 9. References

### 9.1. Normative References

[FIPS186-2] NIST, "Digital Signature Standard", FIPS 186-2, 2000.

[IEEE1363] IEEE, "Standard Specifications for Public Key Cryptography", IEEE 1363, 2000.

[NISTSP80057] NIST, "Recommendation on Key Management", SP 800-57, August 2005, <http://csrc.nist.gov/publications/nistpubs/>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3278] Blake-Wilson, S., Brown, D., and P. Lambert, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 3278, April 2002.

[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.

[RFC3766] Orman, H. and P. Hoffman, "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", BCP 86, RFC 3766, April 2004.

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[X9.62] ANSI, "Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI X9.62, 2005.

[X9.63] ANSI, "Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport using Elliptic Curve Cryptography", ANSI X9.63, 2001.

## 9.2. Informative References

[LENSTRA] Lenstra, A. and E. Verheul, "Selecting Cryptographic Key Sizes", Journal of Cryptography 14, 255-293, 2001.

[SEC2] Standards for Efficient Cryptography Group, "SEC 2 - Recommended Elliptic Curve Domain Parameters", Ver. 1.0, 2000, <http://www.secg.org>.

## Authors' Addresses

Larry Zhu
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

EMail: lzhu@microsoft.com

Karthik Jaganathan
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

EMail: karthikj@microsoft.com

Kristin Lauter
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

EMail: klauter@microsoft.com

## Full Copyright Statement



## Intellectual Property