Network Working Group
Request for Comments: 5114
Category: Informational

M. Lepinski
S. Kent
BBN Technologies

January 2008

Additional Diffie-Hellman Groups for Use with IETF Standards

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document describes eight Diffie-Hellman groups that can be used in conjunction with IETF protocols to provide security for Internet communications. The groups allow implementers to use the same groups with a variety of security protocols, e.g., SMIME, Secure SHell (SSH), Transport Layer Security (TLS), and Internet Key Exchange (IKE).

All of these groups comply in form and structure with relevant standards from ISO, ANSI, NIST, and the IEEE. These groups are compatible with all IETF standards that make use of Diffie-Hellman or Elliptic Curve Diffie-Hellman cryptography.

These groups and the associated test data are defined by NIST on their web site [EX80056A], but have not yet (as of this writing) been published in a formal NIST document. Publication of these groups and associated test data, as well as describing how to use Diffie-Hellman and Elliptic Curve Diffie-Hellman for key agreement in all of the protocols cited below, in one RFC, will facilitate development of interoperable implementations and support the Federal Information Processing Standard (FIPS) validation of implementations that make use of these groups.

Table of Contents

1. Introduction

This document provides parameters and test data for several Diffie-Hellman (D-H) groups that can be used with IETF protocols that employ D-H keys, (e.g., IKE, TLS, SSH, and SMIME) and with IETF standards, such as Public Key Infrastructure for X.509 Certificates (PKIX) (for certificates that carry D-H keys). These groups complement others already documented for the IETF, including the "Oakley" groups defined in RFC 2409 [RFC2409] for use with IKEv1, and several additional D-H groups defined later, e.g., [RFC3526] and [RFC4492].

The initial impetus for the definition of D-H groups (in the IETF) arose in the IPsec (IKE) context, because of the use of an ephemeral, unauthenticated D-H exchange as the starting point for that protocol. RFC 2409 defined five standard Oakley Groups: three modular exponentiation groups and two elliptic curve groups over GF[2^N]. One modular exponentiation group (768 bits - Oakley Group 1) was declared to be mandatory for all IKEv1 implementations to support, while the other four were optional. Sixteen additional groups subsequently have been defined and registered with IANA for use with IKEv1, including eight that have also been registered for use with IKEv2. All of these additional groups are optional in the IKE context. Of the twenty-one groups defined so far for use with IKE, eight are MODP groups (exponentiation groups modulo a prime), ten are EC2N groups (elliptic curve groups over GF[2^N]), and three are ECP groups (elliptic curve groups over GF[P]).

The purpose of this document is to provide the parameters and test data for eight additional groups, in a format consistent with existing RFCs along with instructions on how these groups can be used with IETF protocols such as SMIME, SSH, TLS, and IKE. Three of these groups were previously specified for use with IKE [RFC4753], and five of these groups were previously specified for use with TLS [RFC4492]. (The latter document does not provide or reference test data for the specified groups). By combining the specification of all eight groups with test data and instructions for use in a variety of protocols, this document serves as a resource for implementers who may wish to offer the same Diffie-Hellman groups for use with multiple IETF protocols.

All of these groups are compatible with applicable ISO [ISO-14888-3], ANSI [X9.62], and NIST [NIST80056A] standards for Diffie-Hellman key exchange. These groups and the associated test data are defined by NIST on their web site [EX80056A], but have not yet (as of this writing) been published in a formal NIST document. Publication of these groups with associated test data as an RFC will facilitate development of interoperable implementations and support FIPS validation of implementations that make use of these groups.

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. Additional Diffie-Hellman Groups

This section contains the specification for eight groups for use in IKE, TLS, SSH, etc. There are three standard prime modulus groups and five elliptic curve groups. All groups were taken from publications of the National Institute of Standards and Technology, specifically [DSS] and [NIST80056A]. Test data for each group is provided in Appendix A.

2.1. 1024-bit MODP Group with 160-bit Prime Order Subgroup

The hexadecimal value of the prime is:

```
p = B10B8F96 A080E01D DE92DE5E AE5D54EC 52C99FBC FB06A3C6
    9A6A9DCA 52D23B61 6073E286 75A23D18 9838EF1E 2EE652C0
    13ECB4AE A9061123 24975C3C D49B83BF ACCBDD7D 90C4BD70
    98488E9C 219A7372 4EFFD6FA E5644738 FAA31A4F F55BCCC0
    A151AF5F 0DC8B4BD 45BF37DF 365C1A65 E68CFDA7 6D4DA708
    DF1FB2BC 2E4A4371
```

The hexadecimal value of the generator is:

```
g = A4D1CBD5 C3FD3412 6765A442 EFB99905 F8104DD2 58AC507F
    D6406CFF 14266D31 266FEA1E 5C41564B 777E690F 5504F213
    160217B4 B01B886A 5E91547F 9E2749F4 D7FBD7D3 B9A92EE1
    909D0D22 63F80A76 A6A24C08 7A091F53 1DBF0A01 69B6A28A
    D662A4D1 8E73AFA3 2D779D59 18D08BC8 858F4DCE F97C2A24
    855E6EEB 22B3B2E5
```

The generator generates a prime-order subgroup of size:

```
q = F518AA87 81A8DF27 8ABA4E7D 64B7CB9D 49462353
```

2.2. 2048-bit MODP Group with 224-bit Prime Order Subgroup

The hexadecimal value of the prime is:

```
p =  AD107E1E 9123A9D0 D660FAA7 9559C51F A20D64E5 683B9FD1
     B54B1597 B61D0A75 E6FA141D F95A56DB AF9A3C40 7BA1DF15
     EB3D688A 309C180E 1DE6B85A 1274A0A6 6D3F8152 AD6AC212
     9037C9ED EFDA4DF8 D91E8FEF 55B7394B 7AD5B7D0 B6C12207
     C9F98D11 ED34DBF6 C6BA0B2C 8BBC27BE 6A00E0A0 B9C49708
     B3BF8A31 70918836 81286130 BC8985DB 1602E714 415D9330
     278273C7 DE31EFDC 7310F712 1FD5A074 15987D9A DC0A486D
     CDF93ACC 44328387 315D75E1 98C641A4 80CD86A1 B9E587E8
     BE60E69C C928B2B9 C52172E4 13042E9B 23F10B0E 16E79763
     C9B53DCF 4BA80A29 E3FB73C1 6B8E75B9 7EF363E2 FFA31F71
     CF9DE538 4E71B81C 0AC4DFFE 0C10E64F
```

The hexadecimal value of the generator is:

```
g =  AC4032EF 4F2D9AE3 9DF30B5C 8FFDAC50 6CDEBE7B 89998CAF
     74866A08 CFE4FFE3 A6824A4E 10B9A6F0 DD921F01 A70C4AFA
     AB739D77 00C29F52 C57DB17C 620A8652 BE5E9001 A8D66AD7
     C1766910 1999024A F4D02727 5AC1348B B8A762D0 521BC98A
     E2471504 22EA1ED4 09939D54 DA7460CD B5F6C6B2 50717CBE
     F180EB34 118E98D1 19529A45 D6F83456 6E3025E3 16A330EF
     BB77A86F 0C1AB15B 051AE3D4 28C8F8AC B70A8137 150B8EEB
     10E183ED D19963DD D9E263E4 770589EF 6AA21E7F 5F2FF381
     B539CCE3 409D13CD 566AFBB4 8D6C0191 81E1BCFE 94B30269
     EDFE72FE 9B6AA4BD 7B5A0F1C 71CFFF4C 19C418E1 F6EC0179
     81BC087F 2A7065B3 84B890D3 191F2BFA
```

The generator generates a prime-order subgroup of size:

```
q =  801C0D34 C58D93FE 99717710 1F80535A 4738CEBC BF389A99
     B36371EB
```

2.3. 2048-bit MODP Group with 256-bit Prime Order Subgroup

The hexadecimal value of the prime is:

```
p = 87A8E61D B4B6663C FFBBD19C 65195999 8CEEF608 660DD0F2
    5D2CEED4 435E3B00 E00DF8F1 D61957D4 FAF7DF45 61B2AA30
    16C3D911 34096FAA 3BF4296D 830E9A7C 209E0C64 97517ABD
    5A8A9D30 6BCF67ED 91F9E672 5B4758C0 22E0B1EF 4275BF7B
    6C5BFC11 D45F9088 B941F54E B1E59BB8 BC39A0BF 12307F5C
    4FDB70C5 81B23F76 B63ACAE1 CAA6B790 2D525267 35488A0E
    F13C6D9A 51BFA4AB 3AD83477 96524D8E F6A167B5 A41825D9
    67E144E5 14056425 1CCACB83 E6B486F6 B3CA3F79 71506026
    C0B857F6 89962856 DED4010A BD0BE621 C3A3960A 54E710C3
    75F26375 D7014103 A4B54330 C198AF12 6116D227 6E11715F
    693877FA D7EF09CA DB094AE9 1E1A1597
```

The hexadecimal value of the generator is:

```
g = 3FB32C9B 73134D0B 2E775066 60EDBD48 4CA7B18F 21EF2054
    07F4793A 1A0BA125 10DBC150 77BE463F FF4FED4A AC0BB555
    BE3A6C1B 0C6B47B1 BC3773BF 7E8C6F62 901228F8 C28CBB18
    A55AE313 41000A65 0196F931 C77A57F2 DDF463E5 E9EC144B
    777DE62A AAB8A862 8AC376D2 82D6ED38 64E67982 428EBC83
    1D14348F 6F2F9193 B5045AF2 767164E1 DFC967C1 FB3F2E55
    A4BD1BFF E83B9C80 D052B985 D182EA0A DB2A3B73 13D3FE14
    C8484B1E 052588B9 B7D2BBD2 DF016199 ECD06E15 57CD0915
    B3353BBB 64E0EC37 7FD02837 0DF92B52 C7891428 CDC67EB6
    184B523D 1DB246C3 2F630784 90F00EF8 D647D148 D4795451
    5E2327CF EF98C582 664B4C0F 6CC41659
```

The generator generates a prime-order subgroup of size:

```
q = 8CF83642 A709A097 B4479976 40129DA2 99B1A47D 1EB3750B
    A308B0FE 64F5FBD3
```

2.4. 192-bit Random ECP Group

The curve is based on the integers modulo the prime p given by:

$p = 2^{192} - 2^{64} - 1$

Group prime (in hexadecimal):

```
p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
```

The equation for the elliptic curve is:

$y^2 = x^3 + ax + b \pmod p$

Group curve parameter A (in hexadecimal):

```
a = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFC
```

Group curve parameter B (in hexadecimal):

```
b = 64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
```

The generator for this group is given by: g=(gx,gy) where

```
gx = 188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012

gy = 07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811
```

Group order (in hexadecimal):

```
n = FFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831
```

2.5. 224-bit Random ECP Group

The curve is based on the integers modulo the prime p given by:
p = 2^(224) - 2^(96) + 1

Group prime (in hexadecimal):
p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 00000000
00000001

The equation for the elliptic curve is:
y^2 = x^3 + ax + b (mod p)

Group curve parameter A (in hexadecimal):
a = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
FFFFFFFE

Group curve parameter B (in hexadecimal):
b = B4050A85 0C04B3AB F5413256 5044B0B7 D7BFD8BA 270B3943
2355FFB4

The generator for this group is given by: g=(gx,gy) where

gx = B70E0CBD 6BB4BF7F 321390B9 4A03C1D3 56C21122 343280D6
115C1D21

gy = BD376388 B5F723FB 4C22DFE6 CD4375A0 5A074764 44D58199
85007E34

Group Order (in hexadecimal):
n = FFFFFFFF FFFFFFFF FFFFFFFF FFFF16A2 E0B8F03E 13DD2945
5C5C2A3D

2.6. 256-bit Random ECP Group

The curve is based on the integers modulo the prime p given by:
p = 2^(256)-2^(224)+2^(192)+2^(96)-1

Group prime (in hexadecimal):
p = FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF
FFFFFFFF FFFFFFFF

The equation for the elliptic curve is:
y^2 = x^3 + ax + b (mod p)

Group curve parameter A (in hexadecimal):
a = FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF
FFFFFFFF FFFFFFFC

Group curve parameter B (in hexadecimal):
b = 5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6
3BCE3C3E 27D2604B

The generator for this group is given by: g=(gx,gy) where

gx = 6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0
F4A13945 D898C296

gy = 4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16 2BCE3357 6B315ECE
CBB64068 37BF51F5

Group Order (in hexadecimal):
n = FFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84
F3B9CAC2 FC632551

2.7. 384-bit Random ECP Group

The curve is based on the integers modulo the prime p given by:
p = 2^(384)-2^(128)-2^(96)+2^(32)-1

Group prime (in hexadecimal):
p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFE FFFFFFFF 00000000 00000000 FFFFFFFF

The equation for the elliptic curve is:
y^2 = x^3 + ax + b (mod p)

Group curve parameter A (in hexadecimal):
a = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFE FFFFFFFF 00000000 00000000 FFFFFFFC

Group curve parameter B (in hexadecimal):
b = B3312FA7 E23EE7E4 988E056B E3F82D19 181D9C6E FE814112
0314088F 5013875A C656398D 8A2ED19D 2A85C8ED D3EC2AEF

The generator for this group is given by: g=(gx,gy) where

gx = AA87CA22 BE8B0537 8EB1C71E F320AD74 6E1D3B62 8BA79B98
59F741E0 82542A38 5502F25D BF55296C 3A545E38 72760AB7

gy = 3617DE4A 96262C6F 5D9E98BF 9292DC29 F8F41DBD 289A147C
E9DA3113 B5F0B8C0 0A60B1CE 1D7E819D 7A431D7C 90EA0E5F

Group Order (in hexadecimal):
n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
C7634D81 F4372DDF 581A0DB2 48B0A77A ECEC196A CCC52973

2.8. 521-bit Random ECP Group

The curve is based on the integers modulo the prime p given by:
p = 2^(521)-1

Group Prime (in hexadecimal):
```
p = 000001FF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
    FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
    FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
```

The equation for the elliptic curve is:
y^2 = x^3 + ax + b (mod p)

Group curve parameter A (in hexadecimal):
```
a = 000001FF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
    FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
    FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFC
```

Group curve parameter B (in hexadecimal):
```
b = 00000051 953EB961 8E1C9A1F 929A21A0 B68540EE A2DA725B
    99B315F3 B8B48991 8EF109E1 56193951 EC7E937B 1652C0BD
    3BB1BF07 3573DF88 3D2C34F1 EF451FD4 6B503F00
```

The generator for this group is given by: g=(gx,gy) where

```
gx = 000000C6 858E06B7 0404E9CD 9E3ECB66 2395B442 9C648139
     053FB521 F828AF60 6B4D3DBA A14B5E77 EFE75928 FE1DC127
     A2FFA8DE 3348B3C1 856A429B F97E7E31 C2E5BD66

gy = 00000118 39296A78 9A3BC004 5C8A5FB4 2C7D1BD9 98F54449
     579B4468 17AFBD17 273E662C 97EE7299 5EF42640 C550B901
     3FAD0761 353C7086 A272C240 88BE9476 9FD16650
```

Group Order (in hexadecimal):
```
n = 000001FF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
    FFFFFFFF FFFFFFFF FFFFFFFA 51868783 BF2F966B 7FCC0148
    F709A5D0 3BB5C9B8 899C47AE BB6FB71E 91386409
```

3. Using These Groups with IETF Standards

3.1. X.509 Certificates

Representation of both MODP and Elliptic Curve Diffie-Hellman public keys (and associated parameters) in X.509 certificates is defined in [RFC3279]. The MODP groups defined above MUST be represented via the syntax defined in Section 2.3.3, and the elliptic curve groups via

the syntax defined in Section in 2.3.5 of that RFC. When a Diffie-Hellman public key is encoded in a certificate, if the KeyUsage extension is present, the keyAgreement bits MUST be asserted, and encipherOnly or decipherOnly (but not both) MAY be asserted.

3.2. IKE

Use of MODP Diffie-Hellman groups with IKEv2 is defined in [RFC4306], and the use of MODP groups with IKEv1 is defined in [RFC2409]. However, in the case of ECP Diffie-Hellman groups, the format of key exchange payloads and the derivation of a shared secret has thus far been specified on a group-by-group basis. For the ECP Diffie-Hellman groups defined in this document, the key exchange payload format and shared key derivation procedure specified in [RFC4753] MUST be used (with both IKEv2 and IKEv1).

In order to use a Diffie-Hellman group with IKE, it is required that a transform ID for the group be registered with IANA. The following table provides the Transform IDs of each Diffie-Hellman group described in this document, as registered in both [IANA-IKE] and [IANA-IKE2].

NAME	NUMBER
1024-bit MODP Group with 160-bit Prime Order Subgroup	22
2048-bit MODP Group with 224-bit Prime Order Subgroup	23
2048-bit MODP Group with 256-bit Prime Order Subgroup	24
192-bit Random ECP Group	25
224-bit Random ECP Group	26
256-bit Random ECP Group	19
384-bit Random ECP Group	20
521-bit Random ECP Group	21

3.3. TLS

Use of MODP Diffie-Hellman groups in TLS 1.1 is defined in [RFC4346]. TLS 1.0, the widely deployed predecessor of TLS 1.1, is specified in [RFC2246] and is the same as TLS 1.1 with respect to the use of (MODP) Diffie-Hellman to compute a pre-Master secret. (Currently, the TLS working group is in the process of producing a specification for TLS 1.2. It is unlikely that TLS 1.2 will make significant changes to the use of Diffie-Hellman, and thus the following will likely also be applicable to TLS 1.2).

A server may employ a certificate containing (fixed) Diffie-Hellman parameters, and likewise for a client using a certificate. Thus, the relevant PKIX RFCs (see 3.1 above) are applicable. Alternatively, a server may send ephemeral Diffie-Hellman parameters in the server key exchange message, where the message signature is verified using an RSA- or DSS-signed server certificate. The details for accomplishing this for MODP Diffie-Hellman groups are provided in [RFC2246].

Use of Elliptic Curve Diffie-Hellman in TLS 1.1 (and 1.0) is defined in [RFC4492]. The elliptic curves in this document appear in the IANA EC Named Curve Registry [IANA-TLS], although the names in the registry are taken from the Standards for Efficient Cryptography Group (SECG) specification [SECG] and differ from the names appearing in NIST publications. The following table provides the EC Named Curve ID for each of the elliptic curves along with both the NIST name and the SECG name for the curve.

NAME (NIST)	NUMBER	NAME (SECG)
192-bit Random ECP Group	19	secp192r1
224-bit Random ECP Group	21	secp224r1
256-bit Random ECP Group	23	secp256r1
384-bit Random ECP Group	24	secp384r1
521-bit Random ECP Group	25	secp521r1

3.4. SSH

Use of Diffie-Hellman with SSH was defined initially in [RFC4253]. That RFC defined two MODP Diffie-Hellman groups, and called for the registration of additional groups via an IANA registry. However, [RFC4419] extended the original model to allow MODP Diffie-Hellman parameters to be transmitted as part of the key exchange messages. Thus, using RFC 4419, no additional specifications (or IANA registry actions) are required to enable use of the MODP groups defined in this document. At this time, no RFC describes the use of Elliptic Curve Diffie-Hellman with SSH. However, [SSH-ECC] provides a description of how to make use of Elliptic Curve Diffie-Hellman with SSH.

3.5. SMIME

Use of Diffie-Hellman in SMIME is defined via a discussion of Cryptographic Message Syntax (CMS) enveloped data [RFC3852]. For MODP Diffie-Hellman, the appropriate reference is [RFC2631]. This specification calls for a sender to extract the Diffie-Hellman (MODP) parameters from a recipient’s certificate, and thus the PKIX specifications for representation of Diffie-Hellman parameters suffice. The sender transmits his public key via the

OriginatorIdentifierorKey field, or via a reference to the sender's certificate.

Use of Elliptic Curve Diffie-Hellman in CMS is defined in [RFC3278]. As with use of MODP Diffie-Hellman in the CMS context, the sender is assumed to acquire the recipient's public key and parameters from a certificate. The sender includes his Elliptic Curve Diffie-Hellman public key in the KeyAgreeRecipientInfo originator field. (See Section 8.2 of RFC 3278 for details of the ECC-CMS-SharedInfo).

4. Security Considerations

The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator used. The groups defined in this document were chosen to make the work factor for solving the discrete logarithm problem roughly comparable to an attack on the subgroup.

Using secret keys of an appropriate size is crucial to the security of a Diffie-Hellman exchange. For modular exponentiation groups, the size of the secret key should be equal to the size of q (the size of the prime order subgroup). For elliptic curve groups, the size of the secret key must be equal to the size of n (the order of the group generated by the point g). Using larger secret keys provides absolutely no additional security, and using smaller secret keys is likely to result in dramatically less security. (See [NIST80056A] for more information on selecting secret keys.)

When secret keys of an appropriate size are used, an approximation of the strength of each of the Diffie-Hellman groups is provided in the table below. For each group, the table contains an RSA key size and symmetric key size that provide roughly equivalent levels of security. This data is based on the recommendations in [NIST80057].

GROUP	SYMMETRIC	RSA
1024-bit MODP with 160-bit Prime Subgroup	80	1024
2048-bit MODP with 224-bit Prime Subgroup	112	2048
2048-bit MODP with 256-bit Prime Subgroup	112	2048
192-bit Random ECP Group	80	1024
224-bit Random ECP Group	112	2048
256-bit Random ECP Group	128	3072
384-bit Random ECP Group	192	7680
521-bit Random ECP Group	256	15360

5. IANA Considerations

IANA has taken the following actions:

Updated the IKE and IKEv2 registries to include the following five groups defined in this document: (Note that the other three ECP groups defined in this document have already been added to the IKE registry).

- 1024-bit MODP Group with 160-bit Prime Order Subgroup
- 2048-bit MODP Group with 224-bit Prime Order Subgroup
- 2048-bit MODP Group with 256-bit Prime Order Subgroup
- 192-bit Random ECP Group
- 224-bit Random ECP Group

Updated [IANA-IKE] and [IANA-IKE2] to reflect the above, which now appear as entries in the list of Diffie-Hellman groups given by Group Description. The descriptions are as stated above.

6. Acknowledgments

We wish to thank NIST for publishing the group definitions and providing test data to assist implementers in verifying that software or hardware correctly implements these groups. We also wish to thank Tero Kivinen and Sean Turner for providing helpful comments after reviewing an earlier version of this document.

Appendix A. Test Data

The test data in this appendix is a protocol-independent subset of the test data in [EX80056A]. In the test data for the three modular exponentiation groups, we use the following notation:

xA: The secret key of party A

yA: The public key of party A

xB: The secret key of party B

yB: The public key of party B

Z: The shared secret that results from the Diffie-Hellman computation

In the test data for the five elliptic curve groups, we use the following notation:

dA: The secret value of party A

x_qA: The x-coordinate of the public key of party A

y_qA: The y-coordinate of the public key of party A

dB: The secret value of party B

x_qA: The x-coordinate of the public key of party B

y_qA: The y-coordinate of the public key of party B

x_Z: The x-coordinate of the shared secret that results from the Diffie-Hellman computation

y_Z: The y-coordinate of the shared secret that results form the Diffie-Hellman computation

A.1. 1024-bit MODP Group with 160-bit Prime Order Subgroup

```
      xA =          B9A3B3AE 8FEFC1A2 93049650 7086F845 5D48943E

      yA =                                     2A853B3D 92197501
           B9015B2D EB3ED84F 5E021DCC 3E52F109 D3273D2B 7521281C
           BABE0E76 FF5727FA 8ACCE269 56BA9A1F CA26F202 28D8693F
           EB10841D 84A73600 54ECE5A7 F5B7A61A D3DFB3C6 0D2E4310
           6D8727DA 37DF9CCE 95B47875 5D06BCEA 8F9D4596 5F75A5F3
           D1DF3701 165FC9E5 0C4279CE B07F9895 40AE96D5 D88ED776

      xB =          9392C9F9 EB6A7A6A 9022F7D8 3E7223C6 835BBDDA

      yB =                                     717A6CB0 53371FF4
           A3B93294 1C1E5663 F861A1D6 AD34AE66 576DFB98 F6C6CBF9
           DDD5A56C 7833F6BC FDFF0955 82AD868E 440E8D09 FD769E3C
           ECCDC3D3 B1E4CFA0 57776CAA F9739B6A 9FEE8E74 11F8D6DA
           C09D6A4E DB46CC2B 5D520309 0EAE6126 311E53FD 2C14B574
           E6A3109A 3DA1BE41 BDCEAA18 6F5CE067 16A2B6A0 7B3C33FE

      Z  =                                     5C804F45 4D30D9C4
           DF85271F 93528C91 DF6B48AB 5F80B3B5 9CAAC1B2 8F8ACBA9
           CD3E39F3 CB614525 D9521D2E 644C53B8 07B810F3 40062F25
           7D7D6FBF E8D5E8F0 72E9B6E9 AFDA9413 EAFB2E8B 0699B1FB
           5A0CACED DEAEAD7E 9CFBB36A E2B42083 5BD83A19 FB0B5E96
           BF8FA4D0 9E345525 167ECD91 55416F46 F408ED31 B63C6E6D
```

A.2. 2048-bit MODP Group with 224-bit Prime Order Subgroup

```
      xA =                                               22E62601
           DBFFD067 08A680F7 47F361F7 6D8F4F72 1A0548E4 83294B0C

      yA =                   1B3A6345 1BD886E6 99E67B49 4E288BD7
           F8E0D370 BADDA7A0 EFD2FDE7 D8F66145 CC9F2804 19975EB8
           08877C8A 4C0C8E0B D48D4A54 01EB1E87 76BFEEE1 34C03831
           AC273CD9 D635AB0C E006A42A 887E3F52 FB8766B6 50F38078
           BC8EE858 0CEFE243 968CFC4F 8DC3DB08 4554171D 41BF2E86
           1B7BB4D6 9DD0E01E A387CBAA 5CA672AF CBE8BDB9 D62D4CE1
           5F17DD36 F91ED1EE DD65CA4A 06455CB9 4CD40A52 EC360E84
           B3C926E2 2C4380A3 BF309D56 849768B7 F52CFDF6 55FD053A
           7EF70697 9E7E5806 B17DFAE5 3AD2A5BC 568EBB52 9A7A61D6
           8D256F8F C97C074A 861D827E 2EBC8C61 34553115 B70E7103
           920AA16D 85E52BCB AB8D786A 68178FA8 FF7C2F5C 71648D6F

      xB =                                               4FF3BC96
           C7FC6A6D 71D3B363 800A7CDF EF6FC41B 4417EA15 353B7590
```

```
yB =                  4DCEE992 A9762A13 F2F83844 AD3D77EE
     0E31C971 8B3DB6C2 035D3961 182C3E0B A247EC41 82D760CD
     48D99599 970622A1 881BBA2D C822939C 78C3912C 6661FA54
     38B20766 222B75E2 4C2E3AD0 C7287236 129525EE 15B5DD79
     98AA04C4 A9696CAC D7172083 A97A8166 4EAD2C47 9E444E4C
     0654CC19 E28D7703 CEE8DACD 6126F5D6 65EC52C6 7255DB92
     014B037E B621A2AC 8E365DE0 71FFC140 0ACF077A 12913DD8
     DE894734 37AB7BA3 46743C1B 215DD9C1 2164A7E4 053118D1
     99BEC8EF 6FC56117 0C84C87D 10EE9A67 4A1FA8FF E13BDFBA
     1D44DE48 946D68DC 0CDD7776 35A7AB5B FB1E4BB7 B856F968
     27734C18 4138E915 D9C3002E BCE53120 546A7E20 02142B6C

Z  =                  34D9BDDC 1B42176C 313FEA03 4C21034D
     074A6313 BB4ECDB3 703FFF42 4567A46B DF75530E DE0A9DA5
     229DE7D7 6732286C BC0F91DA 4C3C852F C099C679 531D94C7
     8AB03D9D ECB0A4E4 CA8B2BB4 591C4021 CF8CE3A2 0A541D33
     994017D0 200AE2C9 516E2FF5 14577926 9E862B0F B474A2D5
     6DC31ED5 69A7700B 4C4AB16B 22A45513 531EF523 D7121207
     7B5A169B DEFFAD7A D9608284 C7795B6D 5A5183B8 7066DE17
     D8D671C9 EBD8EC89 544D45EC 061593D4 42C62AB9 CE3B1CB9
     943A1D23 A5EA3BCF 21A01471 E67E003E 7F8A69C7 28BE490B
     2FC88CFE B92DB6A2 15E5D03C 17C464C9 AC1A46E2 03E13F95
     2995FB03 C69D3CC4 7FCB510B 6998FFD3 AA6DE73C F9F63869
```

A.3. 2048-bit MODP Group with 256-bit Prime Order Subgroup

```
xA =                                       0881382C DB87660C
     6DC13E61 4938D5B9 C8B2F248 581CC5E3 1B354543 97FCE50E

yA =                  2E9380C8 323AF975 45BC4941 DEB0EC37
     42C62FE0 ECE824A6 ABDBE66C 59BEE024 2911BFB9 67235CEB
     A35AE13E 4EC752BE 630B92DC 4BDE2847 A9C62CB8 15274542
     1FB7EB60 A63C0FE9 159FCCE7 26CE7CD8 523D7450 667EF840
     E4919121 EB5F01C8 C9B0D3D6 48A93BFB 75689E82 44AC134A
     F544711C E79A02DC C3422668 4780DDDC B4985941 06C37F5B
     C7985648 7AF5AB02 2A2E5E42 F09897C1 A85A11EA 0212AF04
     D9B4CEBC 937C3C1A 3E15A8A0 342E3376 15C84E7F E3B8B9B8
     7FB1E73A 15AF12A3 0D746E06 DFC34F29 0D797CE5 1AA13AA7
     85BF6658 AFF5E4B0 93003CBE AF665B3C 2E113A3A 4E905269
     341DC071 1426685F 4EF37E86 8A8126FF 3F2279B5 7CA67E29

xB =                                       7D62A7E3 EF36DE61
     7B13D1AF B82C780D 83A23BD4 EE670564 5121F371 F546A53D
```

```
yB =                    575F0351 BD2B1B81 7448BDF8 7A6C362C
     1E289D39 03A30B98 32C5741F A250363E 7ACBC7F7 7F3DACBC
     1F131ADD 8E03367E FF8FBBB3 E1C57844 24809B25 AFE4D226
     2A1A6FD2 FAB64105 CA30A674 E07F7809 85208863 2FC04923
     3791AD4E DD083A97 8B883EE6 18BC5E0D D047415F 2D95E683
     CF14826B 5FBE10D3 CE41C6C1 20C78AB2 0008C698 BF7F0BCA
     B9D7F407 BED0F43A FB2970F5 7F8D1204 3963E66D DD320D59
     9AD9936C 8F44137C 08B180EC 5E985CEB E186F3D5 49677E80
     607331EE 17AF3380 A725B078 2317D7DD 43F59D7A F9568A9B
     B63A84D3 65F92244 ED120988 219302F4 2924C7CA 90B89D24
     F71B0AB6 97823D7D EB1AFF5B 0E8E4A45 D49F7F53 757E1913

Z  =                    86C70BF8 D0BB81BB 01078A17 219CB7D2
     7203DB2A 19C877F1 D1F19FD7 D77EF225 46A68F00 5AD52DC8
     4553B78F C60330BE 51EA7C06 72CAC151 5E4B35C0 47B9A551
     B88F39DC 26DA14A0 9EF74774 D47C762D D177F9ED 5BC2F11E
     52C879BD 95098504 CD9EECD8 A8F9B3EF BD1F008A C5853097
     D9D1837F 2B18F77C D7BE01AF 80A7C7B5 EA3CA54C C02D0C11
     6FEE3F95 BB873993 85875D7E 86747E67 6E728938 ACBFF709
     8E05BE4D CFB24052 B83AEFFB 14783F02 9ADBDE7F 53FAE920
     84224090 E007CEE9 4D4BF2BA CE9FFD4B 57D2AF7C 724D0CAA
     19BF0501 F6F17B4A A10F425E 3EA76080 B4B9D6B3 CEFEA115
     B2CEB878 9BB8A3B0 EA87FEBE 63B6C8F8 46EC6DB0 C26C5D7C
```

A.4. 192-bit Random ECP Group

```
dA   = 323FA316 9D8E9C65 93F59476 BC142000 AB5BE0E2 49C43426

x_qA = CD46489E CFD6C105 E7B3D325 66E2B122 E249ABAA DD870612

y_qA = 68887B48 77DF51DD 4DC3D6FD 11F0A26F 8FD38443 17916E9A

dB   = 631F95BB 4A67632C 9C476EEE 9AB695AB 240A0499 307FCF62

x_qB = 519A1216 80E00454 66BA21DF 2EEE47F5 973B5005 77EF13D5

y_qB = FF613AB4 D64CEE3A 20875BDB 10F953F6 B30CA072 C60AA57F

x_Z  = AD420182 633F8526 BFE954AC DA376F05 E5FF4F83 7F54FEBE

y_Z  = 4371545E D772A597 41D0EDA3 2C671112 B7FDDD51 461FCF32
```

A.5. 224-bit Random ECP Group

```
dA   =                                                  B558EB6C
       288DA707 BBB4F8FB AE2AB9E9 CB62E3BC 5C7573E2 2E26D37F

x_qA =                                                  49DFEF30
       9F81488C 304CFF5A B3EE5A21 54367DC7 833150E0 A51F3EEB

y_qA =                                                  4F2B5EE4
       5762C4F6 54C1A0C6 7F54CF88 B016B51B CE3D7C22 8D57ADB4

dB   =                                                  AC3B1ADD
       3D9770E6 F6A708EE 9F3B8E0A B3B480E9 F27F85C8 8B5E6D18

x_qB =                                                  6B3AC96A
       8D0CDE6A 5599BE80 32EDF10C 162D0A8A D219506D CD42A207

y_qB =                                                  D491BE99
       C213A7D1 CA3706DE BFE305F3 61AFCBB3 3E2609C8 B1618AD5

x_Z  =                                                  52272F50
       F46F4EDC 91515690 92F46DF2 D96ECC3B 6DC1714A 4EA949FA

y_Z  =                                                  5F30C6AA
       36DDC403 C0ACB712 BB88F176 3C3046F6 D919BD9C 524322BF
```

A.6. 256-bit Random ECP Group

```
dA   =                                  81426414 5F2F56F2
       E96A8E33 7A128499 3FAF432A 5ABCE59E 867B7291 D507A3AF

x_qA =                                  2AF502F3 BE8952F2
       C9B5A8D4 160D09E9 7165BE50 BC42AE4A 5E8D3B4B A83AEB15

y_qA =                                  EB0FAF4C A986C4D3
       8681A0F9 872D79D5 6795BD4B FF6E6DE3 C0F5015E CE5EFD85

dB   =                                  2CE1788E C197E096
       DB95A200 CC0AB26A 19CE6BCC AD562B8E EE1B5937 61CF7F41

x_qB =                                  B120DE4A A3649279
       5346E8DE 6C2C8646 AE06AAEA 279FA775 B3AB0715 F6CE51B0

y_qB =                                  9F1B7EEC E20D7B5E
       D8EC685F A3F071D8 37270270 92A84113 85C34DDE 5708B2B6

x_Z  =                                  DD0F5396 219D1EA3
       93310412 D19A08F1 F5811E9D C8EC8EEA 7F80D21C 820C2788
```

```
      y_Z  =                                          0357DCCD 4C804D0D
             8D33AA42 B848834A A5605F9A B0D37239 A115BBB6 47936F50
```

A.7. 384-bit Random ECP Group

```
      dA   = D27335EA 71664AF2 44DD14E9 FD126071 5DFD8A79 65571C48
             D709EE7A 7962A156 D706A90C BCB5DF29 86F05FEA DB9376F1

      x_qA = 793148F1 787634D5 DA4C6D90 74417D05 E057AB62 F82054D1
             0EE6B040 3D627954 7E6A8EA9 D1FD7742 7D016FE2 7A8B8C66

      y_qA = C6C41294 331D23E6 F480F4FB 4CD40504 C947392E 94F4C3F0
             6B8F398B B29E4236 8F7A6859 23DE3B67 BACED214 A1A1D128

      dB   = 52D1791F DB4B70F8 9C0F00D4 56C2F702 3B612526 2C36A7DF
             1F802311 21CCE3D3 9BE52E00 C194A413 2C4A6C76 8BCD94D2

      x_qB = 5CD42AB9 C41B5347 F74B8D4E FB708B3D 5B36DB65 915359B4
             4ABC1764 7B6B9999 789D72A8 4865AE2F 223F12B5 A1ABC120

      y_qB = E171458F EAA939AA A3A8BFAC 46B404BD 8F6D5B34 8C0FA4D8
             0CECA163 56CA9332 40BDE872 3415A8EC E035B0ED F36755DE

      x_Z  = 5EA1FC4A F7256D20 55981B11 0575E0A8 CAE53160 137D904C
             59D926EB 1B8456E4 27AA8A45 40884C37 DE159A58 028ABC0E

      y_Z  = 0CC59E4B 046414A8 1C8A3BDF DCA92526 C48769DD 8D3127CA
             A99B3632 D1913942 DE362EAF AA962379 374D9F3F 066841CA
```

A.8. 521-bit Random ECP Group

```
      dA   =              0113 F82DA825 735E3D97 276683B2 B74277BA
             D27335EA 71664AF2 430CC4F3 3459B966 9EE78B3F FB9B8683
             015D344D CBFEF6FB 9AF4C6C4 70BE2545 16CD3C1A 1FB47362

      x_qA =              01EB B34DD757 21ABF8AD C9DBED17 889CBB97
             65D90A7C 60F2CEF0 07BB0F2B 26E14881 FD4442E6 89D61CB2
             DD046EE3 0E3FFD20 F9A45BBD F6413D58 3A2DBF59 924FD35C

      y_qA =              00F6 B632D194 C0388E22 D8437E55 8C552AE1
             95ADFD15 3F92D749 08351B2F 8C4EDA94 EDB0916D 1B53C020
             B5EECAED 1A5FC38A 233E4830 587BB2EE 3489B3B4 2A5A86A4

      dB   =              00CE E3480D86 45A17D24 9F2776D2 8BAE6169
             52D1791F DB4B70F7 C3378732 AA1B2292 8448BCD1 DC2496D4
             35B01048 066EBE4F 72903C36 1B1A9DC1 193DC2C9 D0891B96
```

```
x_qB =              010E BFAFC6E8 5E08D24B FFFCC1A4 511DB0E6
       34BEEB1B 6DEC8C59 39AE4476 6201AF62 00430BA9 7C8AC6A0
       E9F08B33 CE7E9FEE B5BA4EE5 E0D81510 C24295B8 A08D0235

y_qB =              00A4 A6EC300D F9E257B0 372B5E7A BFEF0934
       36719A77 887EBB0B 18CF8099 B9F4212B 6E30A141 9C18E029
       D36863CC 9D448F4D BA4D2A0E 60711BE5 72915FBD 4FEF2695

x_Z  =              00CD EA89621C FA46B132 F9E4CFE2 261CDE2D
       4368EB56 56634C7C C98C7A00 CDE54ED1 866A0DD3 E6126C9D
       2F845DAF F82CEB1D A08F5D87 521BB0EB ECA77911 169C20CC

y_Z  =              00F9 A7164102 9B7FC1A8 08AD07CD 4861E868
       614B865A FBECAB1F 2BD4D8B5 5EBCB5E3 A53143CE B2C511B1
       AE0AF5AC 827F60F2 FD872565 AC5CA0A1 64038FE9 80A7E4BD
```

Normative References

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

Informative References

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, June 1999.

[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.

[RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, May 2003.

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

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.

[RFC4306] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4419] Friedl, M., Provos, N., and W. Simpson, "Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol", RFC 4419, March 2006.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006.

[RFC4753] Fu, D. and J. Solinas, "ECP Groups For IKE and IKEv2", RFC 4753, January 2007.

[SSH-ECC] Green, J. and D. Stebila, "Elliptic-Curve Algorithm Integration in the Secure Shell Transport Layer", Work in Progress, 2007.

[IANA-IKE] Internet Assigned Numbers Authority, Internet Key Exchange (IKE) Attributes. http://www.iana.org/assignments/ipsec-registry

[IANA-IKE2] IKEv2 Parameters. http://www.iana.org/assignments/ikev2-parameters

[IANA-TLS] Internet Assigned Numbers Authority, Transport Layer Security (TLS) Attributes. http://www.iana.org/assignments/tls-parameters

[ISO-14888-3] International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 14888-3:2006, Information Technology: Security Techniques: Digital Signatures with Appendix: Part 3 - Discrete Logarithm Based Mechanisms.

[DSS] National Institute for Standards and Technology, Digital Signature Standard (DSS), FIPS PUB 186-2, January 2000. http://csrc.nist.gov/publications/fips/index.html

[NIST80056A] National Institute of Standards and Technology, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography," NIST Special Publication 800-56A, March 2006. http://csrc.nist.gov/CryptoToolkit/KeyMgmt.html

[EX80056A] National Institute for Standards and Technology, "Examples for NIST 800-56A," May 2007. http://csrc.nist.gov/groups/ST/toolkit/examples.html

[NIST80057] National Institute of Standards and Technology, "Recommendation for Key Management - Part 1", NIST Special Publication 800-57.

[SECG] SECG, "Recommended Elliptic Curve Domain Parameters", SEC 2, 2000, <http://www.secg.org/>.

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

Author's Addresses

Matt Lepinski
BBN Technologies
10 Moulton St.
Cambridge, MA 02138

EMail: mlepinski@bbn.com

Stephen Kent
BBN Technologies
10 Moulton St.
Cambridge, MA 02138

EMail: kent@bbn.com

Full Copyright Statement

Intellectual Property