Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 24, 2015

B. Black
Microsoft
J. Bos
NXP Semiconductors
C. Costello
Microsoft Research
A. Langley
Google Inc
P. Longa
M. Naehrig
Microsoft Research
December 21, 2014

Rigid Parameter Generation for Elliptic Curve Cryptography
draft-black-rpgecc-01

Abstract

This memo describes algorithms for deterministically generating parameters for elliptic curves over prime fields offering high practical security in cryptographic applications, including Transport Layer Security (TLS) and X.509 certificates. The algorithms can generate domain parameters at any security level for modern (twisted) Edwards curves.

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 http://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 June 24, 2015.

Table of Contents

1. Introduction

Since the initial standardization of elliptic curve cryptography (ECC) in [SEC1] there has been significant progress related to both efficiency and security of curves and implementations. Notable examples are algorithms protected against certain side-channel attacks, different 'special' prime shapes which allow faster modular arithmetic, and a larger set of curve models from which to choose. There is also concern in the community regarding the generation and potential weaknesses of the curves defined in [NIST].

This memo describes a deterministic algorithm for generation of elliptic curves for cryptography. The constraints in the generation process produce curves that support constant-time, exception-free scalar multiplications that are resistant to a wide range of side-channel attacks including timing and cache attacks, thereby offering high practical security in cryptographic applications. The deterministic algorithm operates without any hidden parameters, reliance on randomness or any other processes offering opportunities for manipulation of the resulting curves. The selection between curve models is determined by choosing the curve form that supports the fastest (currently known) complete formulas for each modularity option of the underlying field prime. Specifically, the Edwards curve x^2 + y^2 = 1 + dx^2y^2 is used with primes p with p = 3 mod 4, and the twisted Edwards curve -x^2 + y^2 = 1 + dx^2y^2 is used for primes p with p = 1 mod 4.

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].

2. Scope and Relation to Other Specifications

This document specifies a deterministic algorithm for generating elliptic curve domain parameters over prime fields GF(p), with p having a length of twice the desired security level in bits, in (twisted) Edwards form.

3. Security Requirements

For each curve at a specific security level:

1. The domain parameters SHALL be generated in a simple, deterministic manner, without any secret or random inputs. The derivation of the curve parameters is defined in Section 5.

2. The trace of Frobenius MUST NOT be in {0, 1} in order to rule out the attacks described in [Smart], [AS], and [S], as in [EBP].

3. MOV Degree: the embedding degree k MUST be greater than (r - 1) / 100, as in [EBP].

4. CM Discriminant: discriminant D MUST be greater than 2^100, as in [SC].

4. Notation

Throughout this document, the following notation is used:

- p: Denotes the prime number defining the base field.
- GF(p): The finite field with p elements.
- d: An element in the finite field GF(p), different from -1,0.
- Ed: The elliptic curve Ed/GF(p): x^2 + y^2 = 1 + dx^2y^2 in Edwards form, defined over GF(p) by the parameter d.
- tEd: The elliptic curve tEd/GF(p): -x^2 + y^2 = 1 + dx^2y^2 in twisted Edwards form, defined over GF(p) by the parameter d.
- rd: The largest odd divisor of the number of GF(p)-rational points on Ed or tEd.
- td: The trace of Frobenius of Ed or tEd such that #Ed(GF(p)) = p + 1 - td or #tEd(GF(p)) = p + 1 - td, respectively.
- rd': The largest odd divisor of the number of GF(p)-rational points on the non-trivial quadratic twist Ed' or tEd'.
- hd: The index (or cofactor) of the subgroup of order rd in the group of GF(p)-rational points on Ed or tEd.
- hd': The index (or cofactor) of the subgroup of order rd' in the group of GF(p)-rational points on the non-trivial quadratic twist of Ed or tEd.
- P: A generator point defined over GF(p) of prime order rd on Ed or tEd.
- X(P): The x-coordinate of the elliptic curve point P.
- Y(P): The y-coordinate of the elliptic curve point P.

5. Parameter Generation

This section describes the generation of the curve parameters, namely the curve parameter d, and a generator point P of the prime order subgroup of the elliptic curve. Best practice is to use primes with p = 3 mod 4. For compatibility with some deployed implementations, a generation process for primes with p = 1 mod 4 is also provided.

5.1. Deterministic Curve Parameter Generation

5.1.1. Edwards Curves

For a prime p = 3 mod 4, the elliptic curve Ed in Edwards form is determined by the non-square element d from GF(p), different from -1,0 with smallest absolute value such that #Ed(GF(p)) = hd * rd, #Ed'(GF(p)) = hd' * rd', hd = hd' = 4, and both subgroup orders rd and rd' are prime. In addition, care must be taken to ensure the MOV degree and CM discriminant requirements from Section 3 are met.

```
Input: a prime p, with p = 3 mod 4
Output: the parameter d defining the curve Ed
1. Set d = 0
2. repeat
     repeat
       if (d > 0) then
         d = -d
       else
         d = -d + 1
       end if
     until d is not a square in GF(p)
     Compute rd, rd', hd, hd' where #Ed(GF(p)) = hd * rd,
     #Ed'(GF(p)) = hd' * rd', hd and hd' are powers of 2 and rd, rd'
     are odd
   until ((hd = hd' = 4) and rd is prime and rd' is prime)
3. Output d
```

GenerateCurveEdwards

5.1.2. Twisted Edwards Curves

For a prime p = 1 mod 4, the elliptic curve tEd in twisted Edwards form is determined by the non-square element d from GF(p), different from -1,0 with smallest absolute value such that #tEd(GF(p)) = hd * rd, #tEd'(GF(p)) = hd' * rd', hd = 8, hd' = 4 and both subgroup orders rd and rd' are prime. In addition, care must be taken to ensure the MOV degree and CM discriminant requirements from Section 3 are met.

```
Input: a prime p, with p = 1 mod 4
Output: the parameter d defining the curve tEd
1. Set d = 0
2. repeat
     repeat
       if (d > 0) then
         d = -d
       else
         d = -d + 1
       end if
     until d is not a square in GF(p)
     Compute rd, rd', hd, hd' where #tEd(GF(p)) = hd * rd,
     #tEd'(GF(p)) = hd' * rd', hd and hd' are powers of 2 and rd, rd'
     are odd
   until (hd = 8 and hd' = 4 and rd is prime and rd' is prime)
3. Output d
```

GenerateCurveTEdwards

6. Generators

The generator points P = (X(P),Y(P)) for all curves are selected by taking the smallest positive value x in GF(p) (when represented as an integer) such that (x, y) is on the curve and such that (X(P),Y(P)) = 8 * (x, y) has large prime order rd.

```
Input: a prime p and curve parameters non-square d and
       a = -1 for twisted Edwards (p = 1 mod 4) or
       a = 1 for Edwards (p = 3 mod 4)
Output: a generator point P = (X(P), Y(P)) of order rd
1. Set x = 0 and found_gen = false
2. while (not found_gen) do
    x = x + 1
    while ((1 - a * x^2) * (1 - d * x^2) is not a quadratic
           residue mod p) do
      x = x + 1
    end while
    Compute an integer s, 0 < s < p, such that
           s^2 * (1 - d * x^2) = 1 - a * x^2 mod p
    Set y = min(s, p - s)

    (X(P), Y(P)) = 8 * (x, y)

    if ((X(P), Y(P)) has order rd on Ed or tEd, respectively) then
      found_gen = true
    end if
  end while
3. Output (X(P),Y(P))
```

GenerateGen

7. Isogenies from the (twisted) Edwards to the Montgomery model

For applications requiring Montgomery curves, such as x-only point format for elliptic curve Diffie-Hellmann (ECDH) key exchange, isogenies from the generated (twisted) Edwards curves can be produced as described in the following sections.

7.1. Edwards to Montgomery for p = 3 (mod 4)

For a prime p = 3 mod 4, and a given Edwards curve Ed: x^2 + y^2 = 1 + d x^2 y^2 over GF(p) with non-square parameter d, let A = -(4d - 2). Then the Montgomery curve

EM: v^2 = u^3 + Au^2 + u

is isogenous to Ed over GF(p). The following map is a 4-isogeny from Ed to EM over GF(p):

```
    phi:   Ed -> EM, (x,y) -> (u,v), where
        u =  y^2 / x^2,
        v =  -y(x^2 + y^2 - 2) / x^3.
```

The neutral element (0,1) and the point of order two (0,-1) on Ed are mapped to the point at infinity on EM. The dual isogeny is given by

```
 phi_d: EM -> Ed, (u,v) -> (x,y), where
     x = 4v(u - 1)(u + 1) / (u^4 - 2u^2 + 4v^2 + 1),
     y = (u^2 + 2v - 1)(u^2 - 2v - 1) / (-u^4 + 2uv^2 + 2Au + 4u^2 + 1).
```

It holds phi_d(phi((x,y))) = [4](x,y) on Ed and phi(phi_d((u,v))) = [4](u,v) on EM.

7.2. Twisted Edwards to Montogmery for p = 1 (mod 4)

For a prime p = 1 mod 4, and a given twisted Edwards curve tEd: -x^2 + y^2 = 1 + d x^2 y^2 over GF(p) with non-square parameter d, let A = 4d + 2. Then the Montgomery curve

```
    EM: v^2 = u^3 + Au^2 + u
```

is isogenous to tEd over GF(p). Let s in GF(p) be a fixed square root of -1, i.e. s is a solution to the equation s^2 + 1 = 0 over GF(p). Then, the following map is a 4-isogeny from tEd to EM over GF(p):

```
    phi:   tEd -> EM, (x,y) -> (u,v), where
        u =  -y^2 / x^2,
        v =  -ys(x^2 - y^2 + 2) / x^3.
```

The neutral element (0,1) and the point of order two (0,-1) on tEd are mapped to the point at infinity on EM. The dual isogeny is given by

```
 phi_d: EM -> tEd, (u,v) -> (x,y), where
     x = 4sv(u - 1)(u + 1) / (u^4 - 2u^2 + 4v^2 + 1),
     y = (u^2 + 2v - 1)(u^2 - 2v - 1) / (-u^4 + 2uv^2 + 2Au + 4u^2 + 1).
```

It holds phi_d(phi((x,y))) = [4](x,y) on tEd and phi(phi_d((u,v))) = [4](u,v) on EM.

8. Recommended Curves

The following figures give parameters for recommended twisted Edwards and Edwards curves at the 128 and 192 bit security levels generated using the algorithms defined in previous sections. All integer values are unsigned.

```
    p = 0x7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
          FFFFFFFFFFED
    d = 0x1DB41
    r = 0x1000000000000000000000000000000014DEF9DEA2F79CD65812
          631A5CF5D3ED
 x(P) = 0x5C88197130371C6958E48E7C57393BDEDBA29F9231D24B3D4DA2
          242EC821CDF1
 y(P) = 0x6FEC03B956EC4A0E51A838029242F8B107C27399CC7840C34B95
          5E478A8FB7A5
    h = 0x8
```

p = 2^255 - 19, twisted Edwards

The isogenous Montgomery curve for p = 2^255 - 19 is given by A = 0x76D06.

```
    p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
          FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEC3
    d = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
          FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD19F
    r = 0x3FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE2471A1
          CB46BE1CF61E4555AAB35C87920B9DCC4E6A3897D
 x(P) = 0x61B111FB45A9266CC0B6A2129AE55DB5B30BF446E5BE4C005763FFA
          8F33163406FF292B16545941350D540E46C206BDE
 y(P) = 0x82983E67B9A6EEB08738B1A423B10DD716AD8274F1425F56830F98F
          7F645964B0072B0F946EC48DC9D8D03E1F0729392
    h = 0x4
```

p = 2^384 - 317, Edwards

The isogenous Montgomery curve for p = 2^384 - 317 is given by A = 0xB492.

9. TLS NamedCurve Types

As defined in [RFC4492], the name space NamedCurve is used for the negotiation of elliptic curve groups for key exchange during TLS session establishment. This document adds new NamedCurve types for the elliptic curves defined in this document:

```
enum {
    ietfp255t1(TBD1),
    ietfp255x1(TBD2),
    ietfp384e1(TBD3),
    ietfp384x1(TBD4)
} NamedCurve;
```

These curves are suitable for use with Datagram TLS [RFC6347].

10. Use with ECDSA

The (twisted) Edwards curves generated by the procedure defined in this draft are suitable for use in signature algorithms such as ECDSA. In compliance with [RFC5480], which only supports named curves, namedCurve OIDs must be defined for the generated curves and points must be represented as (x,y) in either uncompressed or compressed format.

10.1. Object Identifiers

The following object identifiers represent the (twisted) Edwards domain parameter sets defined in this draft:

```
ietfp255t1 OBJECT IDENTIFIER ::= {[TBDOID] 1}

ietfp384e1 OBJECT IDENTIFIER ::= {[TBDOID] 2}
```

11. Acknowledgements

The authors would like to thank Tolga Acar, Karen Easterbrook and Brian LaMacchia for their contributions to the development of this draft.

12. Security Considerations

TBD

13. Intellectual Property Rights

The authors have no knowledge about any intellectual property rights that cover either the generation algorithms or the usage of the domain parameters defined herein.

14. IANA Considerations

IANA is requested to assign numbers for the curves listed in Section 9 in the "EC Named Curve" [IANA-TLS] registry of the "Transport Layer Security (TLS) Parameters" registry as follows:

Value	Description	DTLS-OK	Reference
TBD1	ietfp255t1	Y	this doc
TBD2	ietfp255x1	Y	this doc
TBD3	ietfp384e1	Y	this doc
TBD4	ietfp384x1	Y	this doc

Table 1

15. References

15.1. Normative References

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

15.2. Informative References

[AS] Satoh, T. and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", 1998.

[EBP] ECC Brainpool, "ECC Brainpool Standard Curves and Curve Generation", October 2005, <http://www.ecc-brainpool.org/download/Domain-parameters.pdf>.

[ECCP] Bos, J., Halderman, J., Heninger, N., Moore, J., Naehrig, M., and E. Wustrow, "Elliptic Curve Cryptography in Practice", December 2013, <https://eprint.iacr.org/2013/734>.

[FPPR] Faugere, J., Perret, L., Petit, C., and G. Renault, 2012, <http://dx.doi.org/10.1007/978-3-642-29011-4_4>.

[IANA-TLS] IANA, "EC Named Curve Registry", 2014, <https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-8>.

[MSR] Bos, J., Costello, C., Longa, P., and M. Naehrig, "Selecting Elliptic Curves for Cryptography: An Efficiency and Security Analysis", February 2014, <http://eprint.iacr.org/2014/130.pdf>.

[NIST] National Institute of Standards, "Recommended Elliptic Curves for Federal Government Use", July 1999, <http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf>.

[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.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC4050] Blake-Wilson, S., Karlinger, G., Kobayashi, T., and Y. Wang, "Using the Elliptic Curve Signature Algorithm (ECDSA) for XML Digital Signatures", RFC 4050, April 2005.

[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.

[RFC4754] Fu, D. and J. Solinas, "IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 4754, January 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

[RFC5753] Turner, S. and D. Brown, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 5753, January 2010.

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

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[S] Semaev, I., "Evaluation of discrete logarithms on some elliptic curves", 1998.

[SC] Bernstein, D. and T. Lange, "SafeCurves: choosing safe curves for elliptic-curve cryptography", June 2014, <http://safecurves.cr.yp.to/>.

[SEC1] Certicom Research, "SEC 1: Elliptic Curve Cryptography", September 2000, <http://www.secg.org/collateral/sec1_final.pdf>.

[Smart] Smart, N., "The discrete logarithm problem on elliptic curves of trace one", 1999.

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

Authors' Addresses

Benjamin Black
Microsoft
One Microsoft Way
Redmond, WA 98115
US

Email: benblack@microsoft.com

Joppe W. Bos
NXP Semiconductors
Interleuvenlaan 80
3001 Leuven
Belgium

Email: joppe.bos@nxp.com

Craig Costello
Microsoft Research
One Microsoft Way
Redmond, WA 98115
US

Email: craigco@microsoft.com

Adam Langley
Google Inc

Email: agl@google.com

Patrick Longa
Microsoft Research
One Microsoft Way
Redmond, WA 98115
US

Email: plonga@microsoft.com

Michael Naehrig
Microsoft Research
One Microsoft Way
Redmond, WA 98115
US

Email: mnaehrig@microsoft.com