Abstract

Elliptic curves with pairings are useful tools for constructing cryptographic primitives. In this memo, we specify domain parameters of Barreto-Naehrig curves (BN-curves) [8]. The BN-curve is an elliptic curve suitable for pairings and allows us to achieve high security and efficiency of cryptographic schemes. This memo specifies domain parameters of four 254-bit BN-curves [1] [2] [5] which allow us to obtain efficient implementations.

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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 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
1. Introduction

Elliptic curves with a special map called a pairing or bilinear map allow cryptographic primitives to achieve functions or efficiency which cannot be realized by conventional mathematical tools. There are identity-based encryption (IBE), attribute-based encryption (ABE), ZSS signature, broadcast encryption (BE) as examples of such primitives. IBE realizes powerful management of public keys by allowing us to use a trusted identifier as a public key. ABE

Table of Contents

1. Introduction ....................................................... 2
2. Requirements Terminology ......................................... 3
3. Preliminaries .......................................................... 3
  3.1. Elliptic Curve ..................................................... 3
  3.2. Bilinear Map ....................................................... 4
4. Domain Parameter Specification ..................................... 5
  4.1. Notation for Domain Parameters and Types of Sextic Twists ..... 5
  4.2. Efficient Domain Parameters for 254-Bit-Curves ................ 7
     4.2.1. Domain Parameters by Beuchat et al. ....................... 7
     4.2.2. Domain Parameters by Nogami et al. / Aranha et al. .... 9
     4.2.3. Domain Parameters Scott .................................... 11
     4.2.4. Domain Parameters by BCMNPZ ............................... 13
5. Object Identifiers .................................................. 15
6. Security Considerations ............................................ 16
   6.1. Subgroup Security (OPTIONAL requirement) ..................... 17
7. Acknowledgements ................................................... 18
8. Change log ........................................................... 18
9. References ............................................................ 18
   9.1. Normative References ........................................... 18
   9.2. Informative References ........................................ 19
Appendix A. Domain Parameters Based on ISO Document ................. 21
   A.1. Specific ISO domain parameters ................................ 21
      A.1.1. Domain Parameters for 224-Bit Curves .................... 21
      A.1.2. Domain Parameters for 256-Bit Curves .................... 21
      A.1.3. Domain Parameters for 384-Bit Curves .................... 22
      A.1.4. Domain Parameters for 512-Bit Curves .................... 22
      A.1.5. Security of ISO curves ................................... 22
Authors’ Addresses ................................................... 23
provides a rich decryption condition based on boolean functions and attributes corresponding to a secret key or a ciphertext. The ZSS signature gives a shorter size of signature than that of ECDSA. BE provides an efficient encryption procedure in a broadcast setting.

Some of these cryptographic schemes based on elliptic curves with pairings were proposed in the IETF (e.g. [9], [10], and [11]) and used in some protocols (e.g. [12], [13], [14], [15], and [16]). These cryptographic primitives will be used actively more in the IETF due to their functions or efficiency.

We need to choose an appropriate type of elliptic curve and parameters for the pairing-based cryptographic schemes, because the choice has great impact on security and efficiency of these schemes. However, an RFC on elliptic curves with pairings has not yet been provided in the IETF.

In this memo, we specify domain parameters of Barreto-Naehrig curve (BN-curve) [8]. The BN-curve allows us to achieve high security and efficiency with pairings due to an optimum embedding degree for 128-bit security. This memo specifies domain parameters of four 254-bit BN-curves ([1] and [2]) because of these efficiencies ([5]). These BN-curves are known as efficient curves in academia and particularly provide efficient pairing computation which is generally slowest operation in pairing-based cryptography. There are optimized source codes of ([1] and [2]) as open source software ([20], [21], and [23]), respectively. This memo describes domain parameters of 224, 256, 384, and 512-bit curves which are compliant with ISO document [3] and organizes differences between types of elliptic curves which are compliant with ISO document [3] in Appendix A.

2. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this memo are to be interpreted as described in [4].

3. Preliminaries

In this section, we introduce the definition of elliptic curve and bilinear map, notation used in this memo.

3.1. Elliptic Curve

Throughout this memo, let \( p > 3 \) be a prime, \( q = p^n \), and \( n \) be a natural number. Also, let \( F_q \) be a finite field. The curve defined by the following equation \( E \) is called an elliptic curve.
E : \( y^2 = x^3 + A \times x + B \) such that A, B are in \( F_q \), 
\( 4 \times A^3 + 27 \times B^2 \neq 0 \) mod \( F_q \)

Solutions \((x, y)\) for an elliptic curve \( E \), as well as the point at 
infinity, are called \( F_q \)-rational points. The additive group is 
constructed by a well-defined operation in the set of \( F_q \)-rational 
points. Typically, the cyclic additive group with prime order \( r \) and 
the base point \( G \) in its group is used for the cryptographic 
applications. Furthermore, we define terminology used in this memo 
as follows.

\[ O_E: \] the point at infinity over elliptic curve \( E \).

\[ \#E(F_q) \]: number of points on an elliptic curve \( E \) over \( F_q \).

\[ \text{cofactor } h: \ h = \#E(F_p)/r. \]

\[ \text{embedding degree } k: \] minimum integer \( k \) such that \( r \) is a divisor of 
\( q^k - 1 \)

### 3.2. Bilinear Map

Let \( G_1 \) be an additive group of prime order \( r \) and let \( G_2 \) and \( G_T \) be 
additive and multiplicative groups, respectively, of the same order. 
Let \( P, Q \) be generators of \( G_1, G_2 \) respectively. We say that \((G_1, \ G_2, G_T)\) are asymmetric bilinear map groups if there exists a 
bilinear map \( e: (G_1, G_2) \rightarrow G_T \) satisfying the following 
properties:

1. Bilinearity: for any \( S \) in \( G_1 \), for any \( T \) in \( G_2 \), for any \( a, b \) in 
   \( Z_r \), we have the relation \( e([a]S, [b]T) = e(S, T)^{a \times b}. \)

2. Non-degeneracy: for any \( S \) in \( G_1 \), \( e(S, T) = 1 \) if and only if \( S = 0_E \). 
   Similarly, for any \( S \) in \( G_1 \), \( e(S, T) = 1 \) if and only if \( T = O_E \).

3. Computability: for any \( S \) in \( G_1 \), for any \( T \) in \( G_2 \), the bilinear 
   map is efficiently computable.

For \( \text{BN-curves} \), \( G_1 \) is a \( r \)-order cyclic subgroup of \( E(F_p) \) and \( G_2 \) is 
a subgroup of \( E(F_{p^k}) \), where \( k \) is the embedding degree of the 
curve. The group \( G_T \) is the set of \( r \)-th roots of unity in the finite 
field \( F_{p^k} \).
4. Domain Parameter Specification

In this section, this memo specifies the domain parameters for four 254-bit elliptic curves which allow us to efficiently compute the operation of a pairing at high levels of security.

4.1. Notation for Domain Parameters and Types of Sextic Twists

Here, we define notations for specifying domain parameters and explain types of pairing friendly curves.

The BN-curves $E$ over $\mathbb{F}_p$ satisfy following equation.

$$y^2 = x^3 + B$$

for $B$ in $\mathbb{F}_p$

The values $p$ and $r$ are computed from a suitable integer $t$.

- $p$ is a characteristic of a prime field $\mathbb{F}_p$: $p = 36 \times t^4 + 36 \times t^3 + 24 \times t^2 + 6 \times t + 1$.
- $r$ is order of group $E$ over $\mathbb{F}_p$: $r = 36 \times t^4 + 36 \times t^3 + 18 \times t^2 + 6 \times t + 1$.

Also, the value $b$ is the constant of the irreducible field polynomial $u^2 + b$ in $\mathbb{F}_{p^2}$.

Domain parameters of the elliptic curve $E(\mathbb{F}_p)$ and $E(\mathbb{F}_{(p^12)})$ are needed for computation of the pairing. In the pairing over BN-curves, we usually use a sextic twist curve group $E'(\mathbb{F}_{(p^2)})$ and a map $I$ from the sextic twist $E'(\mathbb{F}_{(p^2)})$ to $E(\mathbb{F}_{(p^12)})$ instead of $E(\mathbb{F}_{(p^12)})$. Hence, this memo follows the group and the map. For the details of the group and the map, refer to [8].

The sextic twist curves are classified in two types, which are called D-type and M-type respectively [22]. The D-type sextic twist curve is defined by equation $E'$: $y'^2 = x'^3 + B/s$ when elliptic curve $E(\mathbb{F}_p)$ is set to be $y^2 = x^3 + B$ and represent of $\mathbb{F}_{(p^12)}$ is set to be $\mathbb{F}_{(p^2)}[u]/(u^6 - s)$, where $s$ is in $\mathbb{F}_{(p^2)}^*$. Let $z$ be a root of $u^6 - s$, where $z$ is in $\mathbb{F}_{(p^12)}$. The corresponding map $I$: $E'(\mathbb{F}_{(p^2)}) \rightarrow E(\mathbb{F}_{(p^12)})$ is $(x', y') \rightarrow (z^2 * x', z^3 * y')$. The M-type sextic twist curve is defined by equation $E'$: $y'^2 = x'^3 + B * s$ when elliptic curve $E(\mathbb{F}_p)$ is set to be $y^2 = x^3 + B$ and represent of $\mathbb{F}_{(p^12)}$ is set to be $\mathbb{F}_{(p^2)}[u]/(u^6 - s)$, where $s$ is in $\mathbb{F}_{(p^2)}^*$. The corresponding map $I$: $E'(\mathbb{F}_{(p^2)}) \rightarrow E(\mathbb{F}_{(p^12)})$ is $(x', y') \rightarrow (x' * s^{(-1)} * z^4, y' * s^{(-1)} * z^3)$, with $z^6 = s$.

For the pairing, the group $G_1$ is defined as the subgroup of order $r$ in $E(\mathbb{F}_p)$. Then, the group $G_2$ is defined as the subgroup of order $r$
in \( E'(\mathbb{F}_{p^2}) \). The group \( G_T \) is subgroup of order \( r \) in the multiplicative group \( \mathbb{F}_{p^{12}}^* \). The output of pairing is an element on \( G_T \). The order of \( \mathbb{F}_{p^{12}}^* \) can be decomposed into \( (p^{12} - 1) = (p^6 - 1) \times (p^2 + 1) \times (p^4 - p^2 + 1)/r \). Let the cofactor \( h'' \) of \( r \) on \( \mathbb{F}_{p^{12}}^* \) be \( h''_1 \times h''_2 \), where \( h''_1 = (p^4 - p^2 + 1)/r \) and \( h''_2 = (p^6 - 1) \times (p^2 + 1) \).

These domain parameters are described in the following way.

For elliptic curve \( E(\mathbb{F}_p) \)

- \( \text{G1-Curve-ID} \) is an identifier of the \( G_1 \) curve with which the curve can be referenced.
- \( p_b \) is a prime specifying a base field \( \mathbb{F}_p \).
- \( B \) is the coefficient of the equation \( y^2 = x^3 + B \mod p \) defining \( E \).
- \( G = (x, y) \) is the base point, i.e., a point with \( x \) and \( y \) being its \( x- \) and \( y- \) coordinates in \( E \), respectively.
- \( r \) is the prime order of the group generated by \( G \).
- \( h \) is the cofactor of \( G \) in \( E(\mathbb{F}_p) \)

For twisted curve \( E'(\mathbb{F}_{p^2}) \)

- \( \text{G2-Curve-ID} \) is an identifier of the \( G_2 \) curve with which the curve can be referenced.
- \( p_b \) is a prime specifying a base field.
- \( e2 \) is the constant of an irreducible polynomial specifying extension field \( \mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 - e2) \).
- \( B' \) is the coefficient of the equation \( y'^2 = x'^3 + B' \mod \mathbb{F}_{p^2} \) defining \( E' \).
- \( G' = (x', y') \) is the base point, i.e., a point with \( x' \) and \( y' \) being its \( x'- \) and \( y'- \) coordinates in \( E' \), respectively.
- \( r' \) is the prime order of the group generated by \( G' \).
- \( h' \) is the cofactor of \( r' \) in \( \#E'(\mathbb{F}_{p^2}) \)

For \( \mathbb{F}_{p^{12}}^* \)
GT-Field-ID is an identifier of the \( F_{p^{12}}^* \).

\( p_b \) is a prime specifying base field.

\( r'' \) is the prime order of the group.

\( e2 \) is the constant of the irreducible polynomial of \( F_{p^2}^* = F_p[u]/(u^2 - e2) \).

\( e6 \) is the constant of the irreducible polynomial of \( F_{p^6}^* = F_{p^2}[v]/(v^3 - e6) \).

\( e12 \) is the constant of the irreducible polynomial of \( F_{p^{12}}^* = F_{p^6}[w]/(w^2 - e12) \).

\( h'' \) is the cofactor of \( r \) in \( F_{p^{12}}^* \) s.t. \( h'' = h''_1 \cdot h''_2 \)

\( h''_1 \) is the part of cofactor of \( r \) in \( F_{p^{12}}^* \) s.t. \( h''_1 = (p^4 - p^2 + 1)/r \)

\( h''_2 \) is the part of cofactor of \( r \) in \( F_{p^{12}}^* \) s.t. \( h''_2 = (p^6 - 1) * (p^2 + 1) \)

For the definition of the pairing parameter

Pairing-Param-ID is the set of the identifiers G1-Curve-ID, G2-Curve-ID and GT-Field-ID.

4.2. Efficient Domain Parameters for 254-Bit-Curves

This section specifies the domain parameters for four 254-bit elliptic curves. All twisted domain parameters specified in this section are D-type.

4.2.1. Domain Parameters by Beuchat et al.

The domain parameters by Beuchat et al. [1] generated by \( t = 3fc0100000000000 \).

The domain parameters described in this subsection are defined by elliptic curve \( E(F_p) : y^2 = x^3 + 5 \) and sextic twist \( E'(F_{p^2}) : x'^3 + 5/s = x^3 - u \), where \( F_{p^2} = F_p[u]/(u^2 + 5) \), \( F_{p^6} = F_{p^2}[v]/(v^3 - u) \), \( F_{p^{12}} = F_{p^6}[w]/(w^2 - v) \), \( s = -5/u \). We describe domain parameters of elliptic curves \( E \) and \( E' \). The parameter \( p_b \) is 1 mod 8. For the details of these parameters, refer to [1].

G1-Curve-ID: Fp254BNa
\[ p_b = 0x2370fb049d410fbe4e761a9886e502417d023f40180000017e80600000000001 \]
\[ x = 1 \]
\[ y = 0xd45589b158faaf6ab0e4ad38d998e9982e7ff63964e1460342a592677cccb0 \]
\[ r = 0x2370fb049d410fbe4e761a9886e502411d1af70120000017e80600000000001 \]
\[ h = 1 \]

G2-Curve-ID: Fp254n2BNa

\[ \text{p}_b = 0x2370fb049d410fbe4e761a9886e502417d023f40180000017e80600000000001 \]
\[ e2 = -5 \text{ in } F_p \]
\[ B' = -u \]
\[ x' = 0x19b0be4afe4c330da93cc3533da38a9f430b471c6f8a536e81962ed967909b5 + (0xa1cf585585a61c6e9880b1f2a5c539f7d906ff238fa6341e1de1a2e45c3f72) u \]
\[ y' = 0x17abd366ebbd65333e49c711a80a0cf6d24adf1b9b3990eedcc91731384d267 + (0x0e97d6de9902a27d00e952232a78700863bc9a9be960c32f5bf9fd0a32d345) u \]
\[ r' = r \]
\[ h' = 0x2370fb049d410fbe4e761a9886e50241dc42cf101e0000017e80600000000001 \]

GT-Field-ID: Fp254n12a

\[ p_b = 0x2370fb049d410fbe4e761a9886e502417d023f40180000017e80600000000001 \]
\[ r'' = r \]
\[ e2 = -5 \text{ in } F_p \]
\[ e6 = u \text{ in } F_\{(p^2)\} \]
\[ e12 = v \text{ in } F_\{(p^6)\} \]
h'' = 0x189b459262d16204423a54bb8427aba5530e63254675b78ca28b1f810476f6b3c53ed0e0c24dc30fa0db96f3d713f4343a4870545018ff4ea2c361c594bb
b9798ce81c80fd1d1cc1cdde274c80f3345359b79069f453e128c1502c0939bbcc7c5cd822ab539b98c5bd283a3377cf7638d91a23167c510e5bbf53609af49c01b990678cc10f11cc8620188ffca977741390b5093031edc0f806a7301b26b23c97ea03430a6512a4d56df697e761baaf6604e724be4ff5afdf675954131f2c785e364e09256e04dbdc5eb89733e8ad6a1adacffbb082f399a0d0e0a0ab73d647a964221656337a97192a7a42920fccc1c32eb1a2ab7225b5bf4c7c56d697e0481c6b23808f99ac23c352660b6923ab5347121765223970a69ad73439371870bd0fd163e4596afed9c6f7e9f73082070596e8c495b49f4b1bed21ac7b33b5d084c7ed91d1ae8c38a69d0fa48b800011ee048000000000
h''_1 = 0xa6c5ef7e1002629c65ca37294ca9149f129cbcb50212575b3d18098a
dac4072302e88c14ab0564ad9b2b719304c9ef7c907850461e1ece3a37da640be2032e03c8c7623b30af0d6da963854a4aca504a90ee000017e806000000001
h''_2 = 0x24396d2e7daaf102f72fc17484da5601e50a8e4fe4101271d84f063993031fae7dbc4b6f6a4ba9bbcb86f632e0a8295222ce92ab1fddba857b84b13025fd1c6e9be93d0ab69be21cc330e997025161babcc1d0eb55df3939c59d02e002fde269f16c3785ae71f0e1c256be269fde36925bd420043d390638c02e46f220bf63c0c938ab7e73ad426b32f383084672ea9f0e34df03d5a61b648d2e25c2efd50313acaed74538e40cd7c1827e79a8f14eac80410182feda2e06e801f007882b548ea54dc60e3b32a1790906c7395bd58bed4000e5193fae081e0b4dae5650bb8707a73b116f88a87c70800011ee0480000000000

Pairing-Param-ID: Beuchat = {
  GI-Curve-ID: Fp254BNa
  G2-Curve-ID: Fp254BNb
  GT-Field-ID: Fp254Bn12a
}

4.2.2. Domain Parameters by Nogami et al. / Aranha et al.

The domain parameters by Nogami et al. [2] generated by $t = -0x40800000000000001$. Aranha et al. presented an open source library of the pairing using this parameter $t$.

The domain parameters described in this subsection are defined by elliptic curve $E(F_p): y^2 = x^3 + 2$ and sextic twist $E'(F_{p^2}): x'^3 + 2/s = x'^3 + 1 - u$, where $F_p[p^2] = F_{p}[u]/(u^2 + 1), F_{p}[p^6] = F_{p}[p^2][v]/(v^3 - (1 + u)), F_{p}[p^12] = F_{p}[p^6]/(w^2 - v), 1/s = 1/(1 + u)$. We describes domain parameters of elliptic curves $E$ and $E'$. The parameter $p_b$ is 3 mod 4. For the details of these parameters, refer to [2].

GI-Curve-ID: Fp254BNa

p_b = 0x2523648240000001ba344d800000008612100000000013a700000000013

B = 2

x = 0x2523648240000001ba344d800000008612100000000013a700000000012

y = 1

r = 0x2523648240000001ba344d8000000007ff9f80000000010a100000000000d

h = 1

G2-Curve-ID: Fp254BNb

p_b = 0x2523648240000001ba344d800000008612100000000013a700000000013

e2 = -1 in F_p

B' = 1 + (-1) u

x' = 0x061a10bb519eb62feb8d8c7e8c61edb6a4648bb4898bf0d91ee4224c80
3fb2b + (0x0516aaf9ba737833310aa78c5982aa5b1f4d746baf3784b70d8c34c
e7d54cf3) u

y' = 0x021897a06baf93439a90e096698c822329bd0ae6b3be09bd19f0e07891c
d2b9a + (0x0ebb2b0e7c8b15268f6d4456f5f38d37b09006ff739c578a2d1ae
c6b3ace9b) u

r' = r

h' = 0x2523648240000001ba344d8000000008c2a280000000016ad000000000019

GT-Field-ID: Fp254n12b

p_b = 0x2523648240000001ba344d800000008612100000000013a700000000013

r'' = r

e2 = -1 in F_p

e6 = 1 + u in F_{p^2}
e12 = v in F_2[p^6]

h'' = 0x2928fb3b2b39159ec3fe4cbe857330da83e46fed04d235a4a8daf5ff
96f6abcbe43f20a06f0a0d96b24f9a0cbbce750f56127dcbf5ec9139b8f1c46
c86b49bf8a202af26e4504f2c0f56570e9b5d94c40f385d19085648e62b39
6ddc2df1d3d0652f84fe8e82ccb07b7423fc1ef4e8cc73d605e3e87c0a75f45
ea7f6756d9846ce35d5a34f30396938818ad41914b799c289a7259b5d2e09477
a77bd3c409b19f19e893f8ade90b0aed1b5fc8a07a3ceeb41d4e9ee96b21a832d
db1e93e113edfb704fa532848c18593c0deee9044aab349908017ea38bd6ee6
cic5191f2b6bbee449722f98d2173ad330775452ad10347e125a56fd40f0f6e9a4e
62a3d36a72c8b202ac3c147373b93d93d0c795ca0ca39226e7b4c1b92f99248e
cco0806e0ad7074e9f22387367690f518eac4c70808442a1d530c6cc5d55a69738
676e6c73599bbd020bbe1053c9c86b5c009ad8946cd6f0

h''_1 = 0xc816ed457c4f0cbba598fbf85279d6a283736855af2828a32ad1c29a
142223e6281b946847fde6b69c50d19a04e83b02b91083fe87011a78b30ec3c0
4f5235bd893d80083e82c02278000099261da280000006fd67100000000027
0d

h''_2 = 0x34a94d3d1f0dc12947911459f9c7e1caebcb74609938a7cd37a11adf
6b9bd99ba488c257f6664b18eaf5e6fd52eac7666c59eefee0438bd28494fddd8
88b53a9f06cd9c46fcaead14f6a422f3f96db68ff3d696b0842defd02b7aе853d9
cb6eae194a2457251af4e714ce395c6oae4852c28303971c94054144476d3c3ad8
a7f0cb78a53125d893e87ac3969ecbf474dd99f9e644f7e086c6b07840f2b9a2
5c9f64bfb87e61b01a5427a37f30701029b0b18e809a4938363753b991d19
196b045707176d197dfe87a21127b476fde3e7ec276b2c848552cc2534391bfb5
420df1026219e849c1f94a3d35e0020c9d8849b5c000003f7a17b0

Pairing-Param-ID: Nogami-Aranha = {
  G1-Curve-ID: Fp254BNb
  G2-Curve-ID: Fp254n2BNb
  GT-Field-ID: Fp254n12b
}

4.2.3. Domain Parameters Scott

The domain parameters by Scott generated by t = -0x400080600000004081
[6].

The domain parameters described in this subsection are defined by
eccentric curve E(F_p) : y^2 = x^3 + 2 and sextic twist E'(F_2[p^2]) :

x'^3 + 2/s = x'^3 + 1 - u, where F_2[p^2] = F_p[u]/(u^2 + 1), F_2[p^6] =
F_2[p^2][v]/(v^3 + (1 + u)), F_2[p^12] = F_2[p^6][w]/(w^2 + v), l/s =
1/(1 + u). We describes domain parameters of eccentric curves E and
E'. The parameter p_b is 3 mod 4. For the details of these
parameters, refer to [2].
G1-Curve-ID: Fp254BNc

\[ p_b = 0x240120db6517014efa0bab3696f8d5f06e8a555614f464babe9dbbfeeeb4a713 \]

\[ B = 2 \]

\[ x = 0x240120db6517014efa0bab3696f8d5f06e8a555614f464babe9dbbfeeeb4a712 \]

\[ y = 1 \]

\[ r = 0x240120db6517014efa0bab3696f8d5f00e88d43492b2cb363a75777e8d30210d \]

\[ h = 1 \]

G2-Curve-ID: Fp254n2BNc

\[ p_b = 0x240120db6517014efa0bab3696f8d5f06e8a555614f464babe9dbbfeeeb4a713 \]

\[ e_2 = -1 \text{ in } F_p \]

\[ B' = 1 + (-1) u \]

\[ r' = r \]

\[ x' = 0x0571af2ea9666eb2a53f3fb837172bdd809c03a95c5870f34a8cb340220bf9c0 + (0x0f71abb712a9e6e12c07b58bc01f2f994c3b5a1531cf96609b838e5ccf05bc71) u \]

\[ y' = 0x0b88822fe134c1695b21419bblab9732f707701046a2e6ff3ad10f3c70284b93 + (0x1659b732676b5af5231fb045b3d822c0de6fcaab171bad9c8951afcc800a26775) u \]

\[ h' = 0x240120db6517014efa0bab3696f8d5f0ce8bd6779735fe3f42c6007f50392d19 \]

GT-Field-ID: Fp254n12c

\[ p_b = 0x240120db6517014efa0bab3696f8d5f06e8a555614f464babe9dbbfeeeb4a713 \]

\[ r'' = r \]

\[ e_2 = -1 \text{ in } F_p \]
e6 = 1 + u in F_2(p^2)
e12 = v in F_6(p^6)
h'' = 0x1d43e8fcd92a8e7d54f5820d5a3701e694bad5ec9201a8a58128e0908b
cb747b9c941f2c7713cf91dc9a01561342e892b37c0bccc7873897da12bed5e
ec3246e008c92b4e3e5a56498bb1b44874b164fc2f88cb2e02847eb2550ef4f6b7
ebba59d2d7bbf6a6b348d432b009168f8afdf5ec31daed9dc0c9790d7640fd2085e
d6f6796b5637409896c13aaabbc8aad817ce596a31e581258ed88895978f27e6b4
b5adaadbe327cbe2fd02200fb61a1e9dc7f78e06ed67a0c1f6dca9b1d839e046
ecb9573b030322f4ab9826241af8aad8c1d8f97661f7d6660f1660b845948d1ca4
d92203cbb5798ba37248a67f2f5f2b01dd03efbadd89232efeb54f7f23b
583c0df642183a0066080a3e938fd763eefee80a6a5a1792ce5e4bf740c9442
5a83e7b6f6e685fb5a801664f766722508261c7fda904ac0d5fc9e0068f9b5f79b6e69c21797de3707e7a0f22ae9afbb90f232f0
h''_1 = 0x651238d914d66e916c6f4c59202389fb75a267e7c7feabf4a5ee9ef
5aa05b88f60d6f5d737b92988f3253f3d3c8aa439f07432d8102d47dec7eb0ff07
f7e1282739c9d5a3236579d81733eaf2966bb184134d7ac2c082e05e60634f918
0d
h''_2 = 0x2917c05fa90fae306d470d85d3f04e9265a1736bc281349adbabf
8e5c4b6129d260e89f96240137b86473a62a614754354738767777a255874c9
16f826d23df351380749423add88352eb9838833969e3fccc2b61bfa62642308
509c7ef4dddc267f1f9db3804737b4618a64d77a9c3067cd2d5114c50915e9a6
fd49e3049860c56da205a066dafa99472a91a225abcaaa40s7ee0f8c811889
384be0388717565e7a4e13d320e4d4397c9f1273cf507a6552123e1c30d6
0e0d4536a32d372a3d42d1d904f5d1a0ffff853ab24a4fa6604bc224c04e916
90d6050db8e366a4b78b4b6af9b7cface675844fd40ed13d2b0
Pairing-Param-ID: Scott =
G1-Curve-ID: Fp254BNc
G2-Curve-ID: Fp254n2BNc
GT-Field-ID: Fp254n12c
}

4.2.4. Domain Parameters by BCMNPZ

The domain parameters by BCMNPZ generated by \( t = -0x400002010060205 \) [7].

The domain parameters described in this subsection are defined by elliptic curve \( E(F_p) : y^2 = x^3 + 2 \) and sextic twist \( E'(F_p^2) : x'^3 + 2/s = x'^3 + 1 - u, \) where \( F_p = F_p[1] = (u^2 + 1), F_{p^2} = F_{p^2}[1] = (u^2 + 1), F_{p^6} = F_{p^6}[1] = (u^2 + 1), F_{p^{12}} = F_{p^{12}}[1] = (u^2 + 1). \) We describes domain parameters of elliptic curves E and E'. The parameter p_b is 3 mod 4. For the details of these parameters, refer to [2].
G1-Curve-ID: Fp254BNd

\[ p_b = 0x24000482410f5aadb74e200f3b89d00081cf93e428f0d651e8b2dc2bb460a48b \]

\[ x = 0x24000482410f5aadb74e200f3b89d00081cf93e428f0d651e8b2dc2bb4a48a \]

\[ y = 1 \]

\[ B = 2 \]

\[ r = 0x24000482410f5aadb74e200f3b89d00021cf8de127b73833d7fb71a511a2bfb5 \]

\[ h = 1 \]

G2-Curve-ID: Fp254BNd

\[ p_b = 0x24000482410f5aadb74e200f3b89d00081cf93e428f0d651e8b2dc2bb460a48b \]

\[ e2 = -1 \text{ in } F_p \]

\[ B' = 1 + (-1) \text{ } u \]

\[ r' = r \]

\[ x' = 0x20cfe8b965fc444008a21b12cd2a55f843c1dd68ba12a8bb1f1ddee3533b91a32 + (0x0176f822a5ee7ada449f8f876ee001508dd43b5413e03c8f4ad3e3b38dadaf51) \text{ } u \]

\[ y' = 0x02b27f22c2920fee3b4af218b6d92421780a9bdc66155142fececf3af7f58e872 + (0x14e9c62a36ebce710810576b5401fd0b28126ad2d563bf5043be3347646dfb4) \text{ } u \]

\[ h' = 0x24000482410f5aadb74e200f3b89d000e1cf99e72a2a746ff96a46b257171d21 \]

GT-Field-ID: Fp254n12d

\[ p_b = 0x24000482410f5aadb74e200f3b89d00081cf93e428f0d651e8b2dc2bb460a48b \]

\[ r'' = r \]

\[ e2 = -1 \text{ in } F_p \]
\[ e_6 = 1 + u \in \text{F}_p^2 \]
\[ e_{12} = v \in \text{F}_p^6 \]
\[ h'' = 0x1d39fc2421c459d1f0de7cde7c1285648918cd045a503063f111e3aaabaa83df215962969c6fceb6f999c3747dc0fb3eb830701566b2e2b206368ba4f04ebebcdf9c008c23935547b5a46e37a5f1f6e2674b5f3219c8b446c4fbc2615960004d5f42547d6b9a867244929fd958b2f962fb35d58f0225a524e4199f3e961c67e9b1618141cbe93892841e90048054c324d828c3cabba01c45b1c8d62829192d222d2fa7281370c28fe749df33a45af6bf04c8fc54e271bd28c671b5ef06591044fcede0613d7a0fb7a9f4467428dcdf071e85f86b6097ec6dd14b794aa94a1d189b2227ae75851160753faac94c2cb2c2c15d5be5e68fc316683ac92cf07b030c91b25e4ddd04f8a6fc9c128f52b060f4be0c33dd22007c9df38874bf66ce8f217366bce5b2d0a69d020b0ef5d3a05fe0fa939f27db66812f89bff4c3852044c18aa3059b63505ec878753497904916ce2ede93dd267ccd69fc2f6c50 
\[ h''_1 = 0xb640447a44acc2b50912a1528832c5f4358315c85cd27dc4629b83ad23ca6447537784d1adc703cf92a3z2bf736604c22f7fc113e080bd1a0f04061cc8alcc4df3b0317a331d6cb9e0fbb55404de8fbd905999f354e0ca9d80c9d6b6c6ca35 
\[ h''_2 = 0x290d9d32167d7406812204488b22639b77897f44694c058dd022c21816fc3e82f8b7222ac3b8f9a7a347184422c7278b0d501d0de0374429d873e7ef5c866ca749bc6b55607d2f6dc47f8f1a1bfb770d434104836d6e95f2e72ece6b0ace36bbdc8d94be2d4c7ca44f2312b12932ca02c795a69a34e7ce26ae7afbb2fd599e43ace676bc1566aadd101c07a0965098616e46806838413f3eb842d1d4b6dcd261a852bce85e2b39d159198a82e7e794fe5310fe0c08ec3521b101bfc4d9d49204f2489d162489f3b2c5c07251ae6da0e0b7df86f8464ccb6df13439cd25d90d22d3514c1824b5917c5713a224c08c208f2e9fc510 

Pairing-Param-ID: BCMNPZ = {
  G1-Curve-ID: Fp254BNd
  G2-Curve-ID: Fp254n2BNd
  GT-Field-ID: Fp254n12d
}

5. Object Identifiers

We need to define the following object identifiers. Which organization is suitable for the allotment of these object identifiers?

Beuchat OBJECT IDENTIFIER ::= { TBD }

Nogami-Aranha OBJECT IDENTIFIER ::= { TBD }

Scott OBJECT IDENTIFIER ::= { TBD }

BCMNPZ OBJECT IDENTIFIER ::= {TBD}

6. Security Considerations

For above sections, $G_1$ is a $r$-order cyclic subgroup of $E(F_p)$ and $G_2$ is a subgroup of $E'(F_{p^2})$, where $k$ is the embedding degree of the curve and the group $G_T$ is the set of $r$-th roots of unity in the finite field $F_{p^{12}}^*$. In this section, $G_1$, $G_2$ and $G_T$ imply $E(F_p)$, $E'(F_{p^2})$ and $F_{p^{12}}^*$ respectively.

Pairing-based cryptographic primitives are often based on the hardness of the following problems, so when the elliptic curves from this document are used in such schemes, these problems would apply.

- The elliptic curve discrete logarithm problem in $G_1$ and $G_2$ (ECDLP)
- The finite field discrete logarithm problem in $G_T$ (FFDLP)
- The elliptic curve computational Diffie-Hellman (CDH) problem in $G_1$ and $G_2$
- The elliptic curve computational co-Diffie-Hellman problem in $G_1$ and $G_2$
- The elliptic curve decisional Diffie-Hellman (DDH) problem in $G_1$
- The bilinear Diffie-Hellman (BDH) problem

Algorithms to efficiently solve the problems above, aside from special cases, are unknown. Mainly, there are Pollard-rho algorithm [18] against point of an elliptic curve $G_1$ and $G_2$, and Number Field Sieve method [17] against $G_T$ which is output of pairing as generic attacks against elliptic curve with pairing.

$G_T$ to be larger than $G_1$ and $G_2$, because FFDLP can be computed more efficiently than ECDLP in most cases. Security level of schemes based on pairing depends most weak level for each problems. Thus implementors should necessary to ensure adequate security level for both of problems.

Table 1 shows the security level of elliptic curves described in this memo. Schemes based on the elliptic curves (i.e. $G_1$ and $G_2$) and the finite fields (i.e. $G_T$) must be combined with cryptographic primitives which have similar or greater security level than the scheme.
<table>
<thead>
<tr>
<th>Pairing-Param-ID</th>
<th>Security Level for ECDLP in $G_1$, $G_2$ (bits)</th>
<th>Security Level for FFDLP in $G_T$ (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beuchat</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Nogami-Aranha</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>Scott</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>BCMNPZ</td>
<td>128</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 1: security level of elliptic curves and finite field specified in this memo

6.1. Subgroup Security (OPTIONAL requirement)

For BN-curves, $G_1$ is cryptographic group of large prime order and cofactor $h$ is always 1. On the other hand, $G_2$, $G_T$ are consisted of subgroup of order $h'$ and $h''$ that are not equal to 1 in addition to subgroup of order $r$, resp. Thus implementors who provided groups in $G_2$ and $G_T$, MUST check element of those groups included in subgroup of order $r$ (see [7]).

The order check of $G_T$ can be performed by exponentiation of $h''_1$ and $h''_2$. The exponentiation of $h''_2$ can be easily computed by using Frobenius map. Whereas the exponentiation of $h''_1$ is complicated.

For simplification of the order check which is the smallest prime factor of $h'$ and $h''_1$ will be greater than $r$, of element, we define OPTIONAL security $G_2$-strong and $G_T$-strong security. $G_2$-strong and $G_T$-strong means those order of cryptographic group MUST have the smallest prime factor greater than $r$. Therefore implementors could not check of order, $G_2$-strong and $G_T$-strong cryptographic group will not be insecure.

Table 2 shows the $G_2$, $G_T$-strong security of parameters described in this memo.
Table 2: G2, G3-strong security

<table>
<thead>
<tr>
<th>Pairing-Param-ID</th>
<th>Have G_2-Strong?</th>
<th>Have G_T-Strong?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beuchat</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Nogami-Aranha</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Scott</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>BCMNPZ</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

7. Acknowledgements

This memo was inspired by the content and structure of [19].

8. Change log

NOTE TO RFC EDITOR: Please remove this section in before final RFC publication.

9. References

9.1. Normative References


9.2. Informative References


Appendix A. Domain Parameters Based on ISO Document

We describe the domain parameters for 224, 256, 384, and 512-bit elliptic curves which are compliant with the ISO document and are based on M-type twisted curve. The domain parameters described in below subsections are defined by Elliptic curve $E(F_p): y^2 = x^3 + 3$ and sextic twist $E'(F_{p^2}): y'^2 = x'^3 + 3 * s$, where $F_{p^2} = F_p[u]/(u^2 + 1)$, $F_{p^12} = F_{p^2}[w]/(w^6 - s)$, $s = 1 + u$. We describe domain parameters of elliptic curves $E$. Detailed information on these domain parameters is given in [3].

A.1. Specific ISO domain parameters

A.1.1. Domain Parameters for 224-Bit Curves

G1-Curve-ID: Fp224BN

$p_b = \text{0xfffffffffff107288ec29e602c4520db42180823bb907d1287127833}$

$B = 3$

$x = 1$

$y = 2$

$r = \text{0xfffffffffff107288ec29e602c4420db4218082b36c2accff76c58ed}$

$h = 1$

A.1.2. Domain Parameters for 256-Bit Curves

G1-Curve-ID: Fp256BN

$p_b = \text{0xfffffffffffc0cd46e5f25eee71a49f0cdc65fb12980a82d3292d1abae}$

$B = 3$

$x = 1$

$y = 2$

$r = \text{0xfffffffffffc0cd46e5f25eee71a49e0cdc65fb1299921af62d536cd10b}$

$h = 1$
A.1.3. Domain Parameters for 384-Bit Curves

G1-Curve-ID: Fp384BN

\[ p_b = 0xfffffffffffffffffff2a96823d5920d2a127e3f6fbca024c8fbe29531892c79534f9d306328261550a7cabd7cccd10b \]

\[ B = 3 \]

\[ x = 1 \]

\[ y = 2 \]

\[ r = 0xfffffffffffffffffff2a96823d5920d2a127e3f6fbca023c8fbe29531892c79536487d8ac63e4f4db17384341a5775 \]

\[ h = 1 \]

A.1.4. Domain Parameters for 512-Bit Curves

G1-Curve-ID: Fp512BN

\[ p_b = 0xfffffffffffffffffffffffffff9ec7f01c60ba1d8cb5307c0bbe3c111b0ef455146cf1eacbe98b8e48c65deab236fe1916a55ce5f4c6467b4eb280922adef33 \]

\[ B = 3 \]

\[ x = 1 \]

\[ y = 2 \]

\[ r = 0xfffffffffffffffffffffffffff9ec7f01c60ba1d8cb5307c0bbe3c111b0ef455146cf1eacbe98b8e48c65deab2679a34a10313e04f9a2b406a64a5f519a09ed \]

\[ h = 1 \]

A.1.5. Security of ISO curves

In this section, this memo describes ECDLP on G_1 and G_2, FFDLP on G_T and subgroup security over G_2 and G_T, for ISO curves.

Table 3 shows the security level of ISO curves.
### Table 3: security level of ISO elliptic curves and finite field specified in this memo

<table>
<thead>
<tr>
<th>Pairing-Param-ID</th>
<th>Security Level for ECDLP in $G_1$, $G_2$ (bits)</th>
<th>Security Level for FFDLP in $G_T$ (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO-Fp224</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>ISO-Fp256</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>ISO-Fp384</td>
<td>192</td>
<td>128</td>
</tr>
<tr>
<td>ISO-Fp512</td>
<td>256</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 3 shows the $G_2$, $G_T$-strong security of ISO curves.

### Table 4: $G_2$, $G_T$-strong security of ISO curves

<table>
<thead>
<tr>
<th>Pairing-Param-ID</th>
<th>Have $G_2$-Strong?</th>
<th>Have $G_T$-Strong?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO-Fp224</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ISO-Fp256</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ISO-Fp384</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ISO-Fp512</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 4 shows the $G_2$, $G_T$-strong security of ISO curves.
Abstract

This draft proposes an identity-based authenticated key exchange protocol following the extended Canetti-Krawczyk (id-eCK) model. The protocol is currently the most efficient among the id-eCK protocols.

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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 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.

Table of Contents

1. Introduction ................................. 3
2. Requirements Terminology ..................... 3
3. Notation ...................................... 3
4. Data Type and Its Conversions ................ 5
   4.1. BitString-to-OctetString Conversion (BS2OSP) .... 6
   4.2. OctetString-to-BitString Conversion (OS2BSF) .... 6
   4.3. FieldElement-to-Integer Conversion (FE2IP) ....... 6
   4.4. Integer-to-FieldElement Conversion (I2FEP) ....... 6
   4.5. FieldElement-to-OctetString Conversion (FE2OSP) .... 6
   4.6. OctetString-to-FieldElement Conversion (OS2FEP) .... 7
   4.7. EllipticCurvePoint-to-OctetString Conversion (ECP2OSP) .... 7
   4.8. OctetString-to-EllipticCurvePoint Conversion (OS2ECPP) .... 7
5. Building Block of FSU Key Exchange ............ 7
   5.1. Key Derivation Function ................... 7
   5.2. Hashing to Point ......................... 8
      5.2.1. IHF1 ................................ 9
      5.2.2. OS2FQE .............................. 10
   5.3. Group Membership Test Function ............. 11
6. FSU Key Exchange ............................... 12
   6.1. System Parameter Setup .................... 12
   6.2. Key Distribution by KGC ................... 13
   6.3. FSU Key Exchange Protocol ................. 13
7. Security Considerations ....................... 15
8. Acknowledgements .............................. 15
9. Algorithm Identifiers .......................... 15
10. Change log .................................... 16
11. Test Vectors .................................. 16
12. References .................................... 16
   12.1. Normative References ..................... 16
   12.2. Informative References ................... 16
Appendix A. Construction of Data Conversion ........ 18
   A.1. Construction of BS2OSP .................... 18
   A.2. Construction of OS2BSF .................... 18
   A.3. Construction of FE2IP ..................... 19
   A.4. Construction of I2FEP ..................... 19
   A.5. Construction of FE2OSP .................... 20
   A.6. Construction of OS2FEP .................... 21
   A.7. Construction of ECP2OSP ................... 21
   A.8. Construction of OS2ECPP ................... 23
Authors’ Addresses .............................. 24
1. Introduction

Authenticated key exchange (AKE) is a core security function within many deployed systems today. It is a foundational function that allows end-users and systems alike to be authenticated prior to access to resource and services. Over the past two decades key exchange schemes have been proposed, based on symmetric and asymmetric key cryptography.

A more recent approach to AKE protocol has been the introduction of identity binding to the exchange [7] [8], obviating the need to rely on a public key infrastructure in which digital certificates need to be exchanged by users or end-points that wish to communicate signed and/or encrypted messages.

Identity-based AKE (ID-AKE) schemes rely on the use of the trusted intermediary referred to as the Key Generation Center (KGC). The role of the KGC, among others, is to generate a pair of master public and secret keys based on the user’s identity and to extract a user’s secret key corresponding to his or her identity.

In a 2-pass ID-AKE scheme, an "initiator" entity wishing to share a key with a second entity (referred to as the "responder") sends ephemeral public information to the responder. In its turn, the responder sends another ephemeral public information to the initiator entity. Following this, each entity would then generate a session from a number of parameters, notably their respective secret keys (given by the KGC), their own secret values of the ephemeral information, the identity of the peer they’re communicating with, and the ephemeral information they received from that peer.

We propose a provably secure ID-AKE scheme called "FSU" [4] [5] [6] based on the previous model of [9] and which builds on the previous efforts in [10] [11]. The model underlying the FSU was chosen due to the merit of provable security based on an adversarial model in which the adversary has the freedom to choose keys reveal.

2. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this memo are to be interpreted as described in [1].

3. Notation

This section shows notation used in this memo.
Let $\mathbb{F}_q$ be a finite field with $q = p^n$ elements for a prime $p$ and an integer $n$ and let $E(\mathbb{F}_q)$ be an elliptic curve with an order $r$ and an embedding degree $k$ defined over $\mathbb{F}_q$. An embedding degree $k$ is defined as a minimum integer $k$ such that $r$ is a divisor of $q^k - 1$.

Let $G_1$ (resp. $G_2$) be an additive group with an order $r$ generated by $E(\mathbb{F}_q)$ (resp. $E'(\mathbb{F}_q)$). Let $G_T$ be multiplicative groups with the same order $r$. Let $P_1, P_2$ be generators of $G_1, G_2$ respectively. We say that $(G_1, G_2, G_T)$ are bilinear map groups if there exists a pairing $e: (G_1, G_2) \rightarrow G_T$ satisfying the following properties:

1. Bilinearity: for any $Q_1$ in $G_1$, for any $Q_2$ in $G_2$, for any $a, b$ in $\mathbb{Z}_r$, we have the relation $e(aQ_1, bQ_2) = e(Q_1, Q_2)^{ab}$.

2. Non-degeneracy: for any $Q_1$ in $G_1$, $e(Q_1, Q_2) = 1$ only if $Q_2 = O_{G_2}$ and for any $Q_2$ in $G_2$, $e(Q_1, Q_2) = 1$ only if $Q_1 = O_{G_1}$.

3. Computability: for any $Q_1$ in $G_1$, for any $Q_2$ in $G_2$, the bilinear map is efficiently computable.

This pairing is described in specification of optimal ate pairing specification[3]. It is defined by Pairing-Parm-ID following way.

Pairing-Param-ID = { 
  G1-Curve-ID, 
  G2-Curve-ID 
  GT-Field-ID
}

G1-Curve-ID and G2-Curve-ID is an identifiers of elliptic curve. And GT-Field-ID is an identifier of the G_T range of finite field. G1-Curve-ID, G2-Curve-ID and GT-Field-ID are described in [2] the following way.
G1-Curve-ID = {
  p_b    : A prime specifying base field F_p.
  A, B   : The coefficients of the equation \(y^2 = x^3 A \times x + B\)
           defining \(E\).
  G = (x, y) : The base point, i.e., a point with \(x\) and \(y\)
              being its \(x\)- and \(y\)-coordinates in \(E\), respectively.
  r      : The prime order of the group generated by \(G\).
  h      : The cofactor of \(G\) in \(E\).
}

G2-Curve-ID = {
  p_b    : A prime specifying base field \(F_p\).
  e2     : The constant of an irreducible polynomial specifying
          extension field \(F_{p^2} = F_p[u] / (u^2 - e2)\).
  A', B' : The coefficients of the equation \(y^2 = x^3 A' \times x + B'\)
           defining \(E'\).
  G' = (x', y') : The base point, i.e., a point with \(x'\) and \(y'\)
                being its \(x\)- and \(y\)-coordinates in \(E'\), respectively.
  r'     : The prime order of the group generated by \(G'\).
  h'     : The cofactor of \(G'\) in \(E'\).
}

GT-Filed-ID = {
  p_b    : A prime specifying base field.
  r      : The prime order of the subgroup of \(F_{p^{12}}\).
  e2     : The constant of the irreducible polynomial of \(F_{p^{12}} =
           F_{p^2}[u] / (u^2 - e2)\).
  e6     : The constant of the irreducible polynomial of \(F_{p^6} =
           F_{p^2}[v] / (v^3 - e6)\).
  e12    : The constant of the irreducible polynomial of \(F_{p^{12}} =
           F_{p^6}[w] / (w^2 - e12)\).
  h''    : The cofactor of \(G_T\)
}

In addition, this memo uses the following functions.

floor(x) : The function returning an integer such that \(\max\{x' \in Z | x' \leq x\}\).

ceil(x) : The function returning an integer such that \(\min\{x' \in Z | x' \geq x\}\).

O_E : The point at infinity over elliptic curve \(E\).

4. Data Type and Its Conversions

This section describes data type and its conversion used in this memo.
4.1. BitString-to-OctetString Conversion (BS2OSP)

This memo uses conversion from bit strings to octet strings. Informally, the idea is to pad the bit string with 0’s on the left to make its length a multiple of 8, then chop the result up into octets. Formally, the conversion routine, BS2OSP(B), is specified in Appendix A.1.

4.2. OctetString-to-BitString Conversion (OS2BSP)

This memo uses conversion from octet strings to bit strings. Informally, the idea is simply to view the octet string as a bit string. Formally, the conversion routine, OS2BSP(M), is specified in Appendix A.2.

4.3. FieldElement-to-Integer Conversion (FE2IP)

This memo uses conversion from field elements to integers. An finite field element should be represented as a polynomial with subfield coefficients, which can be represented as a sequence of the coefficients. Informally, the idea is simply to view the sequence of the coefficients as the radix-\(p^m\) representation of the base field elements, where \(p^m\) is the number of the subfield elements. Formally, the conversion routine, FE2IP(a), is specified in Appendix A.3.

4.4. Integer-to-FieldElement Conversion (I2FEP)

This memo uses conversion from integers to field elements. A field element should be represented as a polynomial with subfield coefficients, and it can be represented as a sequence of the coefficients. Informally, the idea is to represent the integer with radix-\(p^m\) positional number system where \(p^m\) is the number of the subfield element, and then convert the each digit to the each coefficient of the polynomial. Formally, the conversion routine, I2FEP(x), is specified in Appendix A.4.

4.5. FieldElement-to-OctetString Conversion (FE2OSP)

This memo uses conversion from field elements to octet strings. This conversion is constructed by using FE2IP and I2SOP conversions. Formally, the conversion routine, FE2OSP(a), is specified in Appendix A.5.
4.6. OctetString-to-FieldElement Conversion (OS2FEP)

This memo uses conversion from octet strings to field elements. This conversion is constructed by using OS2IP and I2FEP conversions. Formally, the conversion routine, OS2FEP(M), is specified in Appendix A.6.

4.7. EllipticCurvePoint-to-OctetString Conversion (ECP2OSP)

This memo uses conversion from elliptic curve points to octet strings. Informally the idea is that, if point compression is being used, the compressed y-coordinate is placed in the leftmost octet of the octet string along with an indication that point compression is on, and the x-coordinate is placed in the remainder of the octet string; otherwise if point compression is off, the leftmost octet indicates that point compression is off, and remainder of the octet string contains the x-coordinate followed by the y-coordinate. Formally, the conversion routine, ECP2OSP(P,R), is specified in Appendix A.7.

4.8. OctetString-to-EllipticCurvePoint Conversion (OS2ECP)

This memo uses conversion from octet strings to elliptic curve points. Informally, the idea is that, if the octet string represents a compressed point, the compressed y-coordinate is recovered from the leftmost octet, the x-coordinate is recovered from the remainder of the octet string, and then the point compression process is reversed; otherwise the leftmost octet of the octet string is removed, the x-coordinate is recovered from the left half of the remaining octet string, and the y-coordinate is recovered from the right half of the remaining octet string. Formally, the conversion routine, OS2ECP(M), is specified in Appendix A.8.

5. Building Block of FSU Key Exchange

This section describes building block for constructing FSU Key Exchange.

5.1. Key Derivation Function

MGF1 is a mask generation function, parameterized by a hash function. MGF1(M,n) is defined as follows:

System parameters:

- Hash : a hash function
- hashLen : the length in octets of the hash function output
Input:
- M: a seed from which a mask is generated, an octet string
- n: the octet length of the output, a positive integer

Output:
- mask: a mask, an octet string of length n

Method:
1. Let n_0 be the octet length of M. If n_0 + 4 is greater than the input limitation for the hash function, output INVALID and stop.
2. Set cThreshold = ceil(n / hashLen)
3. If cThreshold > 2^32, output INVALID and stop
4. Let M' be the empty octet string
5. Set counter = 0
6. B = B_{0}, ..., B_{31} such that counter = B_{31} + B_{30}*2 + ... + B_{0}*2^{31}
7. Compute C = BS2OSP(B)
8. Compute H = Hash(M||C)
9. Set M' = M'||H
10. Set counter = counter + 1
11. If counter < cThreshold, go back to step 6.
12. Set mask = M'_0M'_1...M'_{n-1} where M' = M'_0M'_1M'_2...
13. Output mask

5.2. Hashing to Point

Hashed value should be converted to elliptic curve point as described in this section. Formally, the conversion routine, HASHINGTOPOINT(Curve-ID, Hash, M), is specified as follows:

Input:
o Curve-ID : an elliptic curve parameter
o Hash : a hash function
o M : an octet string

Output:
o P : an elliptic curve point

Method:
1. Set i = 0
2. B = B_0, ..., B_{15} such that counter = B_{15} + B_{14} * 2 + ...
   + B_0 * 2^{15}
3. Compute C = BS2OSP(B)
4. x_0 = OS2FQE(C|M, Hash, F_{p^m}) in F_{p^m}
5. t = x_0^3 + A * x_0 + B
6. If t=0, set P = (x_0, 0) and output h' * P
7. If t is not square in F_{p^m}, set i = i + 1 and go back to step 2
8. Set alpha be one of square roots of t. Then, -alpha is another
   square root of t.
9. Set y_1 = FE2IP(alpha)
10. Set y_2 = FE2IP(-alpha)
11. If y_1 > y_2, set y_0 = -alpha
12. Else (i.e. y_1 <= y_2), set y_0 = alpha
13. Set P = (x_0, y_0)
14. Output h * P

5.2.1. IHF1

Bit string should be converted to hashed non-negative integer less
than an assigned integer as described in this section. Formally, the
conversion routine, IHF1(s,n,Hash) is defined as follows:
Input:
  o $s$: an octet string
  o $n$ : an integer
  o $\text{Hash}$ : a hash function

Output:
  o $v$ in $\mathbb{Z}_n$

Method:
1. Set $\text{hashLen}$ be the length of the output of the hash function $\text{Hash}$
2. Set $h_0$ be the zero string of length $\text{hashLen}$
3. $h_1 = \text{Hash}(h_0 \ || \ s)$
4. $B = B_0, \ldots, B_{l-1} = \text{OS2BSP}(h_1)$
5. $a_1 = \sum_{i=0}^{l-1} 2^{l-1-i} \times B_i$
6. $h_2 = \text{Hash}(h_1 \ || \ s)$
7. $B = B_0, \ldots, B_{l-1} = \text{OS2BSP}(h_2)$
8. $a_2 = \sum_{i=0}^{l-1} 2^{l-1-i} \times B_i$
9. $v = 2^{\text{hashLen}} \times a_1 + a_2 \mod n$
10. Output $v$

5.2.2. **OS2FQE**

Octet string should be converted to hashed finite field element as described in this section. Formally, the conversion routine, $\text{OS2FQE}(s, \text{Hash}, F_{p^m})$ is defined as follows:

Input:
  o $s$: an octet string
  o $\text{Hash}$ : a hash function
o  $F_{p^m}$ : a finite field with $p^m$ elements where $p$ is a prime, and $m > 0$ is an integer

Output:

o  $a$: an element in $F_{p^m}$

Method:

1. Set $i = 0$
2. $B = B_0, \ldots, B_{31}$ such that counter $= B_{31} + B_{30} \cdot 2 + \ldots + B_0 \cdot 2^{31}$
3. Compute $C = BS2OSP(B)$
4. Compute $t_i = IHF1(C|s,p,Hash)$
5. If $i < m$, set $i = i + 1$ and go back to step 2
6. Compute $a = \sum_{i=0}^{m-1} t_i \cdot \beta^i$ where $\beta$ is the variable of the polynomial
7. Output $a$

5.3. Group Membership Test Function

GROUPMEMBERSHIPTEST(Curve-ID, $P$) is a test function that an elliptic curve point is on the correct curve and group. GROUPMEMBERSHIPTEST is defined as follows:

Input:

o  Curve-ID : an elliptic curve identifier
o  $P = (x,y)$ : an elliptic curve point

Output:

o  boolean : an integer in $\{0,1\}$

Method:

1. If $P = O_E$, then output 1
2. If $y^2 \neq x^3 + A \cdot x + B$, then output 0
3. If $h \neq 1$ && $r \cdot P \neq O_E$, then output 0
6. FSU Key Exchange

This section provides the specification of ID-based authenticated key exchange protocol FSU [4] that is an extension of FSU (Fujioka-Suzuki-Ustaoglu) protocol standardized in ISO/IEC11770-3 [5] [6].

6.1. System Parameter Setup

Key Generation Center (KGC) defines the following system parameters in FSU:

- **Pairing-Param-ID**: An identifier for showing asymmetric pairing. i.e., G1-Curve-ID, G2-Curve-ID and GT-Field-ID.
- **G1-Curve-ID**: An identifier for showing an elliptic curve which defines cyclic groups $G_1$ with prime $p_{b_1}$, coefficients $A_1$ and $B_1$, generator $P_1$, order $r$, and cofactor $h_1$.
- **G2-Curve-ID**: An identifier for showing an elliptic curve which defines cyclic groups $G_2$ with prime $p_{b_2}$, irreducible polynomial $e_{2,2}$, coefficients $A_2$ and $B_2$, generator $P_2$, order $r$, and cofactor $h_2$.
- **GT-Field-ID**: An identifier for showing a pairing co-domain group which is subgroup of order $r$ in $G_{\phi_{12}(p)}$. $G_{\phi_{12}(p)}$ is the 12-th cyclotomic subgroup of order $p^4-p^2+1$ in $F_{p^{12}}^*$.
- **HASH-ID**: An identifier for showing a hash function, i.e., $Hash : \{0,1\}^* \rightarrow \{0,1\}^{hashLen}$.
- **hashLen**: Length of output by Hash.
- **KDF-ID**: An identifier for showing key derivation function, i.e., $MGF1: \{0,1\}^* \rightarrow \{0,1\}^n$.
- **n**: Length of output by key derivation function.
- **R**: A point compression type of conversion between elliptic curve point and octet string specifically "Compressed", "Uncompressed", or "Hybrid".

KGC generates the master secret key MSK and master public key MPK from system parameters as following.

1. KGC selects a random integer $z$ in $Z_r$. 

2. KGC computes \( Z_v = z \cdot P_v \) for \( v \) is in \( \{1, 2\} \).

3. KGC sets \( \text{MSK} = z \) and \( \text{MPK} = (Z_1, Z_2) \).

Hash function \( H_v \) are defined as \( H_v(M) = \text{HASHINGTOPOINT}(G_v, \text{Curve-ID}, \text{Hash}, \text{"FSU"}|ECP2OSP(Z_1, R)|ECP2OSP(Z_2, R)||M) \) for \( v \) in \( \{1, 2\} \).
Hash function \( H \) is defined as \( H(M) = \text{MGF1}(\text{"FSU"}|ECP2OSP(Z_1, R)|ECP2OSP(Z_2, R)||M, n) \).

6.2. Key Distribution by KGC

This subsection explains operations of key distribution by KGC. There are two types of static secret key in FSU Key Exchange, respectively static secret key based on cyclic groups in \( G_1 \) and in \( G_2 \). FSU Key Exchange requires that an initiator and a responder use static secret key with different types, respectively. Hence, KGC needs to define a rule for key distribution for users. For example, clients use static secret keys in \( G_1 \) and servers use them in \( G_2 \).

KGC generates static secret key \( D_{i, v} \) for an identifier \( ID_i \) for \( i \) in \( \{A, B\} \) of user in \( G_v \) as following.

1. Let \( \text{MPK} = (Z_1, Z_2) \) and \( \text{MSK} = z \).
2. KGC Compute \( D_{i, v} = z \cdot H_v(ID_i) \).
3. Distribute \( D_{i, v} \) to a user with \( ID_i \).

6.3. FSU Key Exchange Protocol

This subsection describes FSU Key Exchange Protocol in an initiator \( U_A \) with an identifier \( ID_A \) and static secret key \( D_{A,1} \) and a responder \( U_B \) with an identifier \( ID_B \) and static secret key \( D_{B,2} \).

Computation of ephemeral public key by \( U_A \)

1. \( U_A \) selects a random integer \( x_A \) in \( Z_r \).
2. \( U_A \) computes the ephemeral public key \( X_{A,v} = x_A \cdot P_v \) for \( v \) in \( \{1, 2\} \).
3. \( U_A \) computes \( \text{XOS}_{A,v} = \text{ECP2OSP}(X_{A,v}, R) \) for \( v \) in \( \{1, 2\} \).
4. \( U_A \) sends \( (ID_A, ID_B, \text{XOS}_{A,1}, \text{XOS}_{A,2}) \) to \( U_B \).

Computation of ephemeral public key by \( U_B \)

1. \( U_B \) receives \( (ID_A, ID_B, \text{XOS}_{A,1}, \text{XOS}_{A,2}) \).
2. \( U_B \) computes \( X_{(A,v)} = \text{OS2ECPP}(X_{OS_{(A,v)}) \text{ for } v \in \{1,2\}. \)

3. If \( \text{GROUPMEMBERSHプTEST}(G1\text{-Curve-ID, } X_{(A,1)}) = 0 \) \( || \) \( \text{GROUPMEMBERSHプTEST}(G2\text{-Curve-ID, } X_{(A,2)}) = 0 \) \( || \) \( e(X_{(A,1)}, P_2) \) \( \neq e(P_1, X_{(A,2)}) \), then abort.

4. \( U_B \) selects a random ephemeral secret key \( x_B \in Z_r \).

5. \( U_B \) computes the ephemeral public key \( X_{(B,v)} = x_B * P_v \text{ for } v \in \{1,2\}. \)

6. \( U_B \) computes \( XOS_{(B,v)} = \text{ECP2OSP}(X_{(B,v)}, R) \text{ for } v \in \{1,2\}. \)

7. \( U_B \) sends \((ID_B, ID_A, XOS_{(B,1)}, XOS_{(B,2)})\) to \( U_A \).

Computation of session key by \( U_B \)

1. \( U_B \) computes \( \sigma_1 = e(H_1(ID_A), D_{(B,2)}) \).

2. \( U_B \) computes \( \sigma_2 = e(H_1(ID_A) + X_{(A,1)}, D_{(B,2)} + x_B * Z_2) \).

3. \( U_B \) computes \( \sigma_3 = x_B * X_{(A,1)} \).

4. \( U_B \) computes \( \sigma_4 = x_B * X_{(A,2)} \).

5. \( U_B \) computes \( \sigma_{OS_j} = \text{FE2OSP}(\sigma_j) \text{ for } j \in \{1,2\}. \)

6. \( U_B \) computes \( \sigma_{OS_{j'}} = \text{ECP2OSP}(\sigma_{j'}, R) \text{ for } j' \in \{3,4\}. \)

7. Set \( \text{sid} = (ID_A \| ID_B \| XOS_{(A,1)} \| XOS_{(A,2)} \| XOS_{(B,1)} \| XOS_{(B,2)}). \)

8. \( U_B \) computes session key \( K = H(\sigma_{OS_1} \| \sigma_{OS_2} \| \sigma_{OS_3} \| \sigma_{OS_4} \| \text{sid}). \)

Computation of session key by \( U_A \)

1. \( U_A \) computes \( X_{(B,v)} = \text{OS2ECPP}(XOS_{(B,v)}) \text{ for } v \in \{1,2\}. \)

2. If \( \text{GROUPMEMBERSHプTEST}(G1\text{-Curve-ID, } X_{(B,1)}) = 0 \) \( || \) \( \text{GROUPMEMBERSHプTEST}(G2\text{-Curve-ID, } X_{(B,2)}) = 0 \) \( || \) \( e(X_{(B,1)}, P_2) \) \( \neq e(P_1, X_{(B,2)}) \), then abort.

3. \( U_A \) computes \( \sigma_1 = e(D_{(A,1)}, H_2(ID_B)). \)

4. \( U_A \) computes \( \sigma_2 = e(D_{(A,1)} + x_A * Z_1, H_2(ID_B) + X_{(B,2)}). \)

5. \( U_A \) computes \( \sigma_3 = x_A \cdot X_{(B,1)} \).

6. \( U_A \) computes \( \sigma_4 = x_A \cdot X_{(B,2)} \).

7. \( U_A \) computes \( \sigma_{OS,j} = FE2OSP(\sigma_j) \) for \( j \in \{1,2\} \).

8. \( U_A \) computes \( \sigma_{OS,j'} = ECP2OSP(\sigma_{j'},R) \) for \( j' \in \{3,4\} \).

9. Set \( sid = (ID_A||ID_B||XOS_{(A,1)}||XOS_{(A,2)}||XOS_{(B,1)}||XOS_{(B,2)}) \).

10. \( U_A \) computes session key \( K = H(\sigma_{OS_1}||\sigma_{OS_2}||\sigma_{OS_3}||\sigma_{OS_4}||sid) \).

7. Security Considerations

This memo specifies identity-based authenticated key exchange protocol FSU [4] [6] [5] which is secure in the id-eCK(id-based extended Canetti-Krawczyk) security model under the GBDH(gap bilinear DH) assumption [4].

id-eCK security model is the most strong security model in the meaning of that it ensures the safety of session key if any non-trivial combinations of master key, static key, and ephemeral key are leaked.

And id-eCK security model guarantees following 4 security notions:

- MitM(resistance to man in the middle attacks),
- wPFS(weak perfect forward security),
- KCI(resistance to key compromise impersonation attacks),
- RLE(resilience to leakage of ephemeral private keys).

8. Acknowledgements

TBD

9. Algorithm Identifiers

TBD
10. Change log

NOTE TO RFC EDITOR: Please remove this section in before final RFC publication.

11. Test Vectors

TBD

12. References

12.1. Normative References


12.2. Informative References


Appendix A. Construction of Data Conversion

A.1. Construction of BS2OSP

Concrete construction of BS2OSP(B) is specified as follows:

Input:
- B = B_0 B_1 ... B_{l-1} : a bit string of length l

Output:
- M = M_0 M_1 ... M_{n-1} : an octet string of length n = ceil(l/8).

Method:
1. If l = 0, then output empty octet string and stop.
2. For j in {0,...,8n-1}, if j >= 8n - l, set B'_j = B_{j-(8n-l)}, otherwise set B'_j = 0.
3. For i in {0,...,n-1}, set M_i = B'_{8i} B'_{8i+1}...B'_{8i+7}.
4. Output M = M_0 M_1 ... M_{n-1}.

A.2. Construction of OS2BSP

Concrete construction of OS2BSP(M) is specified as follows:

Input:
- M = M_0M_1...M{n-1} : an octet string of length n.

Output:
- B = B_0B_1...B_{l-1} : a bit string of length l = 8*n

Method:
1. If l = 0, then output empty octet string and stop.
2. For i in {0, ..., n-1}, j in {0,...,7}, set B_{8i + j} in {0,1} as M_i = B_{8i} B_{8i+1}...B_{8i+7}.
3. Output B = B_0 B_1 ... B_{l-1}.
A.3. Construction of FE2IP

Concrete construction of FE2IP(a) is specified as follows:

System parameters:

- $\mathbb{F}_{p^{m_2}}/\mathbb{F}_{p^{m_1}}$: a field extension with an irreducible polynomial $\text{Irr}(\mathbb{F}_{p^{m_2}} / \mathbb{F}_{p^{m_1}}; \beta)$

Input:

- $a$: a field element in $\mathbb{F}_{p^{m_2}}$

Output:

- $x$: an integer in $\{0, \ldots, p^{m_2} - 1\}$

Method:

1. If $m_2 = 1$ (i.e. $\mathbb{F}_{p^{m_2}}$ is prime field)
   
   A field element of $\mathbb{F}_{p^{m_2}}$ must be represented as an integer in $\{0, \ldots, p-1\}$
   
   (A) Set $x = a$
   
   (B) Output $x$

2. Else (i.e. $m_2 > 1$)

   (A) Let the coefficients $a_i$ in $\mathbb{F}_{p^{m_1}}$ for $i$ in $\{0, \ldots, m_2 / m_1 - 1\}$ such that $a = \sum_{i=0}^{m_2 / m_1 - 1} a_i \beta^i$

   (B) Compute $x = \sum_{i=0}^{m_2 / m_1 - 1} \text{FE2IP}(a_i) \cdot (p^{m_1})^i$

   (C) Output $x$

A.4. Construction of I2FEP

Concrete construction of I2FEP(x) is specified as follows:

System parameters:

- $\mathbb{F}_{p^{m_2}}/\mathbb{F}_{p^{m_1}}$: a field extension with an irreducible polynomial $\text{Irr}(\mathbb{F}_{p^{m_2}} / \mathbb{F}_{p^{m_1}}; \beta)$
Input:
- \( x \): an integer in \( \{0, \ldots, p^{m_2} - 1\} \)

Output:
- \( a \): a field element in \( \mathbb{F}_{p^{m_2}} \)

Method:

1. If \( m_2 = 1 \) (i.e. \( \mathbb{F}_{p^{m_2}} \) is prime field)
   - A field element of \( \mathbb{F}_{p^{m_2}} \) must be represented as an integer in \( \{0, \ldots, p-1\} \)
   - (A) Set \( a = x \)
   - (B) Output \( a \)

2. Else (i.e. \( m_2 > 1 \))
   - (A) Let \( x_i \) be an element in \( \{0, \ldots, p^{m_1} - 1\} \) for \( i \) in \( \{0, \ldots, m_2 / m_1 - 1\} \) such that \( x = \sum_{i=0}^{m_2 / m_1 - 1} x_i \cdot p^{m_1 \cdot i} \)
   - (B) Compute \( a = \sum_{i=0}^{m_2 / m_1 - 1} I2FEP(x_i) \cdot \beta^{i} \)
   - (C) Output \( a \)

A.5. Construction of FE2OSP

System parameter:
- \( \mathbb{F}_{p^m} \): a finite field with \( p^m \) elements where \( p \) is a prime, and \( m > 0 \) is an integer
- \( n \): an integer equivalent to \( \lceil m \cdot \log_2 p / 8 \rceil \)

Input:
- \( a \): a field element in \( \mathbb{F}_{p^m} \)

Output:
- \( M \): an octet string

Method:
1. Compute $I = \text{FE2IP}(a)$

2. Compute $X = x_0, \ldots, x_{n-1}$ such that $I = x_{n-1} + x_{n-2} \cdot 2 + \ldots + x_{1} \cdot 2^{n-2} + x_{0} \cdot 2^{n-1}$

3. Compute $M = \text{BS2OSP}(X)$

4. Output $M$

A.6. Construction of OS2FEP

System parameter:

- $F_{p^m}$ : a finite field with $p^m$ elements where $p$ is a prime, and $m > 0$ is an integer
- $n$ : an integer equivalent to $\lceil m \cdot \log_2 p / 8 \rceil$

Input:

- $M$ : an octet string

Output:

- $a$ : a field element in $F_{p^m}$

Method:

1. Compute $X = \text{OS2BSP}(M)$

2. Let $X$ be $x_0, \ldots, x_{l-1}$

3. Compute $I = \sum_{i=0}^{l-1} 2^{l-1-i} \cdot x_i$

4. Compute $a = \text{I2FEP}(I)$

5. Output $a$

A.7. Construction of ECP2OSP

Concrete construction of ECP2OSP(P,R), is specified as follows:

System parameters:

- Curve-ID : an elliptic curve parameter

Input:
o  $P$: a point on an elliptic curve over $F_{p^m}$

o  $R$: compression type specifically "Compressed", "Uncompressed", or "Hybrid"

Output:

o  $M$: an octet string of length $n$

Method:

1. If $P = O_E$
   
   (A) Compute $M = \text{BS2OSP}(00000000)$
   
   (B) Output $M$

2. If $P = (x, y) \neq O_E \& \& R = \text{Compressed}$
   
   (A) Set $X = \text{FE2OSP}(x)$
   
   (B) If $p$ is odd $\& \& y = 0$, set $y' = 0$
   
   (C) Else if $p$ is odd $\& \& y \neq 0$, set $y' = y_i \mod 2$ such that $y = y_{m-1} \cdot \beta^{m-1} + \ldots + y_1 \cdot \beta + y_0$ and $i$ is the smallest integer such that $y_i \neq 0$
   
   (D) If $y' = 0$, compute $L = \text{BS2OSP}(00000100)$
   
   (E) If $y' = 1$, compute $L = \text{BS2OSP}(00000101)$
   
   (F) Output $M = L || X$

3. If $P = (x, y) \neq O_E \& \& R = \text{Uncompressed}$
   
   (A) Set $X = \text{FE2OSP}(x)$
   
   (B) Set $Y = \text{FE2OSP}(y)$
   
   (C) Compute $L = \text{BS2OSP}(00000100)$
   
   (D) Output $M = L || X || Y$

4. If $P = (x, y) \neq O_E \& \& R = \text{Hybrid}$
   
   (A) Set $X = \text{FE2OSP}(x)$
   
   (B) Set $Y = \text{FE2OSP}(y)$
(C) If y = 0, set \( y' = 0 \)

(D) Else (i.e. \( y \neq 0 \)) \( y' = y_i \mod 2 \) such that \( y = y_{m-1} \cdot \beta^{m-1} + \ldots + y_1 \cdot \beta + y_0 \) and \( i \) is the smallest integer such that \( y_i \neq 0 \)

(E) If \( y' = 0 \), compute \( L = \text{BS2OSP}(00000010) \)

(F) If \( y' = 1 \), compute \( L = \text{BS2OSP}(00000011) \)

(G) Output \( M = L || X || Y \)

A.8. Construction of OS2ECPP

Concrete construction of OS2ECPP(\( M \)), is specified as follows:

System parameters

- Curve-ID: an elliptic curve parameter

Input:

- \( M \): an octet string

Output:

- \( P \): an elliptic curve point

Method:

1. If \( M = \text{BS2OSP}(00000000) \), output \( P = O_E \)

2. If \( M \) has length \( \lceil m \cdot \log_2(p) / 8 \rceil + 1 \)

   (A) Let \( M \) be \( L || X \) where \( L \) is a single octet

   (B) Compute \( x = \text{OS2FEP}(X) \)

   (C) If \( L = \text{BS2OSP}(00000010) \), then set \( y' = 0 \)

   (D) Else if \( L = \text{BS2OSP}(00000011) \), then set \( y' = 1 \)

   (E) Else output INVALID and stop

   (F) Compute \( w = x^3 + A \cdot x + B \)

   (G) Compute \( \gamma = \text{square}(w) \)
(H) If there is no gamma in $F_{p^m}$, then output INVALID and stop

(I) Else if gamma = 0, then set $y = 0$

(J) Else if $\gamma_i = \gamma' \mod 2$ where $\gamma = \gamma_{m-1} \beta^{m-1} + \ldots + \gamma_1 \beta + \gamma_0$ and $i$ is the smallest integer such that $\gamma_i \neq 0$

(K) Else if $\gamma_i \neq \gamma' \mod 2$, set $y = -\gamma$ where $\gamma = \gamma_{m-1} \beta^{m-1} + \ldots + \gamma_1 \beta + \gamma_0$ and $i$ is the smallest integer such that $\gamma_i \neq 0$

(L) Output $P = (x,y)$

3. If $M$ has length $2 \times \lfloor m \times \log_2 p / 8 \rfloor + 1$

(A) Let $M$ be $L || X || Y$ where $L$ is a single octet, $X$ is $\lfloor m \times \log_2 p / 8 \rfloor$ octets, and $Y$ is $\lfloor m \times \log_2 p / 8 \rfloor$ octets

(B) Unless $L$ is BS2OSP(00000100), BS2OSP(00000110) or BS2OSP(00000111), output INVALID and stop.

(a) Compute $x = \text{OS2FEP}(X)$

(b) Compute $y = \text{OS2FEP}(Y)$

(c) If $(x,y)$ does not satisfy the equation of elliptic curve, then output INVALID and stop

(d) Output $P = (x,y)$

Authors' Addresses

Akihiro Kato
NTT Software Corporation
EMail: kato.akihiro-at-po.ntts.co.jp

Thomas Hardjono
MIT
EMail: hardjono-at-mit.edu
Tetsutaro Kobayashi
NTT
EMail: kobayashi.tetsutaro-at-lab.ntt.co.jp

Tsunekazu Saito
NTT
EMail: saito.tsunekazu-at-lab.ntt.co.jp

Koutarou Suzuki
NTT
EMail: suzuki.koutarou-at-lab.ntt.co.jp
Abstract

Pairing is a special map from two elliptic curve that called Pairing-friendly curves to a finite field and is useful mathematical tools for constructing cryptographic primitives. It allows us to construct powerful primitives. (e.g. [3] and [4])

There are some types of pairing and its choice has an impact on the performance of the primitive. For example, Tate Pairing [3] and Ate Pairing [4] are specified in IETF. This memo focuses on Optimal Ate Pairing [2] which is an improvement of Ate Pairing.

This memo defines Optimal Ate Pairing for any pairing-friendly curve. We can obtain concrete algorithm by deciding parameters and building blocks based on the form of a curve and the description in this memo. It enables us to reduce the cost for specifying Optimal Ate Pairing over additional curves. Furthermore, this memo provides concrete algorithm for Optimal Ate Pairing over BN-curves [7] and its test vectors.

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."
1. Introduction .................................................. 3
2. Requirements Terminology ................................... 3
3. Preliminaries .................................................. 3
   3.1. Elliptic Curve ............................................ 3
   3.2. Bilinear Map .............................................. 4
4. Optimal Ate Pairing .......................................... 4
   4.2. Miller Loop ................................................ 6
   4.3. Straight Line Function ................................... 7
5. Optimal Ate Pairing over BN-curves ......................... 7
   5.1. Straight Line Function over BN-curves .................... 8
   5.2. Doubling Step of Miller Loop over BN-Curves .............. 9
   5.3. Addition Step of Miller Loop over BN-Curves ............. 10
6. Algorithm Identifiers ......................................... 11
7. Security Considerations ....................................... 11
8. Acknowledgements ............................................. 11
9. Change log ..................................................... 11
10. References ................................................... 11
    10.1. Normative References .................................. 11
    10.2. Informative References ................................. 11
Appendix A. Test Vectors of Optimal Ate Pairing over BN-curves 13
    A.1. 254-Bit-Curves by Beuchat et al. ........................ 13
    A.2. 254-Bit-Curves by Nogami et al. / Aranha et al. ......... 14
    A.3. 254-Bit-Curves by Scott ................................. 15
    A.4. 254-Bit-Curves by BCMNPZ ................................ 16
Authors’ Addresses .............................................. 16

1. Introduction

Pairing is a special map from two elliptic curve that called Pairing-friend curves (PFCs) to a finite field and is useful mathematical tools for constructing cryptographic primitives. It allows us to construct powerful primitives like Identity-Based Encryption (IBE) [5] and Functional Encryption (FE) [6]. The IBE and FE provide a rich decryption condition. Some Pairing-Based Cryptography is specified in IETF. (e.g. [3] and [4])

There are some types of pairing and its choice has an impact on the performance of the primitive. For example, primitives by using Tate Pairing [3] and Ate Pairing [4] are specified in IETF. This memo focuses on Optimal Ate Pairing which is an improvement of Ate Pairing. Optimal Ate Pairing allows us to construct Pairing-Based Cryptography with high performance and is implemented in some open source softwares. ([8], [9], and [10])

This memo defines Optimal Ate Pairing [2] for any PFC. We can obtain concrete algorithm by deciding parameters and two building blocks based on the form of a curve. It enables us to reduce the cost for describing the body of Optimal Ate Pairing when Optimal Ate Pairing is specified over additional curves in IETF. Furthermore, this memo provides concrete algorithm for Optimal Ate Pairing over BN-curves [7] and its test vectors. This memo is expected to use by combining Optimal Ate Pairing with a suitable PFC for a primitive in order to realize same functional structure of ECDSA and ECDH. (i.e. DSA over elliptic curve and DH over elliptic curve)

2. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this memo are to be interpreted as described in [1].

3. Preliminaries

In this section, we introduce the definition of elliptic curve and bilinear map, notation used in this memo.

3.1. Elliptic Curve

Throughout this memo, let \( p > 3 \) be a prime, \( q = p^n \), and \( n \) be a natural number. Also, let \( F_q \) be a finite field. The curve defined by the following equation \( E \) is called an elliptic curve.

\[
E : y^2 = x^3 + A * x + B \text{ such that } A, B \text{ are in } F_q,
4 * A^3 + 27 * B^2 \neq 0 \text{ mod } F_q
\]
Solutions \((x, y)\) for an elliptic curve \(E\), as well as the point at infinity, are called \(F_q\)-rational points. The additive group is constructed by a well-defined operation in the set of \(F_q\)-rational points. Typically, the cyclic additive group with prime order \(r\) and the base point \(G\) in its group is used for the cryptographic applications. Furthermore, we define terminology used in this memo as follows.

- \(O_E\): the point at infinity over elliptic curve \(E\).
- \(#E(F_q)\): number of points on an elliptic curve \(E\) over \(F_q\).
- Cofactor \(h\): \(h = \#E(F_p)/r\).
- Embedding degree \(k\): minimum integer \(k\) such that \(r\) is a divisor of \(q^k - 1\).

### 3.2. Bilinear Map

Let \(G_1\) be an additive group of prime order \(r\) and let \(G_2\) and \(G_T\) be additive and multiplicative groups, respectively, of the same order. Let \(P, Q\) be generators of \(G_1, G_2\) respectively. We say that \((G_1, G_2, G_T)\) are asymmetric bilinear map groups if there exists a bilinear map \(e: (G_1, G_2) \rightarrow G_T\) satisfying the following properties:

1. **Bilinearity:** for any \(S\) in \(G_1\), for any \(T\) in \(G_2\), for any \(a, b\) in \(\mathbb{Z}_r\), we have the relation \(e([a]S, [b]T) = e(S, T)^{a \cdot b}\).

2. **Non-degeneracy:** for any \(T\) in \(G_2\), \(e(S, T) = 1\) if and only if \(S = O_E\). Similarly, for any \(S\) in \(G_1\), \(e(S, T) = 1\) if and only if \(T = O_E\).

3. **Computability:** for any \(S\) in \(G_1\), for any \(T\) in \(G_2\), the bilinear map is efficiently computable.

### 4. Optimal Ate Pairing

This section specifies Optimal Ate Pairing \(e\) for \(c_0, \ldots, c_l\) and \(s_i = \sum_{j=i}^l c_j \cdot q^j\) with following conditions:

1. \(c_1\) is not 0
2. \(r\) is a divisor of \(s_0\)
3. \(r^2\) is not a divisor of \(s_0\)
4. \( r \) does not divide \( s_0 \cdot k \cdot q^{(k-1)} - (q^k - 1)/r \cdot \sum_{i=0}^l i \cdot c_i \cdot q^{i-1} \)

Section 4.1 shows a guide to decide these parameters \( c_0, \ldots, c_l \). Optimal Ate Pairing is specified below and Miller Loop \( f \) which are its building blocks are introduced in Section 4.2. Straight Line Function \( l \) which is building blocks of Optimal Ate Pairing and Miller Loop are defined in Section 4.3. Section 4.3 only show the definitions because its descriptions are based on the form (of the PFC?). Practically, concrete algorithms need to be specified for a form of PFC.

Input:
- A point \( P \) in \( G_1 \)
- A point \( Q \) in \( G_2 \)

Output:
- The value \( e(P, Q) \) in \( G_T \)

Method:
1. \( f = 1 \)
2. \( ln = 1 \)
3. for \( i = 0 \) to \( l \)
   (a) \( f = f \cdot f_{c_i, Q}^{q^i}(P) \)
end for
4. for \( i = 0 \) to \( l - 1 \)
   (a) \( ln = ln \cdot l_{[s_i + 1]Q, [c_i \cdot q^i]Q}(P) \)
end for
5. return \( (f \cdot ln)^{(q^k - 1)/r} \)

4.1. Guide for Decision on Parameters for Optimal Ate Pairing

This subsection shows a guide for decision on parameters \( c_0, \ldots, c_l \) for Optimal Ate Pairing. According to [2], a way is to choice coefficients of short vector of the following lattice \( L \) with a minimal number of coefficients as parameters \( c_0, \ldots, c_l \).
L = (v_1, ..., v_{\phi(k)}) where
o v_1 is column vector t(r, -q, -q^2, ..., -q^{\phi(k) - 1})
o v_i is column vector whose i component is 1 and other components
is 0 for i = 2, ..., \phi(k)

4.2. Miller Loop

In this subsection, we specify Miller Loop f which is building block
of Optimal Ate Pairing.

Input:
o A point P in G_1
o A point Q in G_2
o An integer s

Output:
o f_{s, Q}(P)

Method:

1. compute s_0, ..., s_L such that |s| = \sum_{j=0}^{L} s_j * 2^j with
   s_j is in \{0, 1\} and s_L = 1
2. T = Q
3. f = 1
4. for j = L - 1 down to 0
   (A) Doubling Step
      (a) ln = l_{T, T}(P)
      (b) T = 2 * T
   (B) f = f^2 * ln
   (C) if s_j = 1
      (a) Addition Step
         (i) ln = l_{T, Q}(P)
(ii) \( T = T + Q \)

(b) \( f = f' \ast \ln \)

end if

end for

5. if \( s < 0 \), then \( f = f^{-1} \)

6. return \( f \)

4.3. Straight Line Function

Straight Line Function \( l_{Q, Q'}(P) \) is calculated by a point \( P \) for linear equation defined as a line \( l \) though points \( Q, Q' \). Note that Straight Line Function \( l_{Q, Q'}(P) \) is calculated by a point \( P \) for linear equation defined as a tangent line to an elliptic curve \( E \) at a point \( Q \) of \( E \) on condition that \( Q = Q' \). The function is used for Optimal Ate Pairing in Section 4 and Miller Loop in Section 4.2

5. Optimal Ate Pairing over BN-curves

In this section, we specify Optimal Ate Pairing over BN-curves [7]. BN-curves define over a finite field \( F_p \), and have embedding degree \( k = 12, r(t) = 36 \ast t^4 + 36 \ast t^3 + 18 \ast t^2 + 6 \ast t + 1 \), and \( p(t) = 36 \ast t^4 + 36 \ast t^3 + 24 \ast t^2 + 6 \ast t + 1 \), where \( t \) is the specific integer in [7].

The extension fields are defined by following:

\[
\begin{align*}
F_{p^2} & \text{ is set to } F_p[u]/(u^2 - e2) \\
F_{p^6} & \text{ is set to } F_{p^2}[v]/(u^3 - e6) \\
F_{p^{12}} & \text{ is set to } F_{p^6}[w]/(w^2 - e12)
\end{align*}
\]

The constants \( e3, e6 \) and \( e6 \) which are varied by \( G_T \) are defined in [7].

Hence parameters for Optimal Ate Pairing over D-Type twisted curve are following by the method in Section 4.1:

1. \( l = 3 \)
2. \( c_0 = 6 \ast t + 2 \)
3. \( c_1 = 1 \)
4. \( c_2 = -1 \)
5. \( c_3 = 1 \)

These short vectors are specified in section 4. A of [2].

Algorithm of Optimal Ate Pairing by Miller Loop in Section 4.2 based on building blocks specified in Section 5.2 and Section 5.3 and Straight Line Function \( f \) in Section 5.1 over BN-curves is as following:

Input:
- A point \( P \) in \( G_1 \)
- A point \( Q \) in \( G_2 \)

Output:
- The value \( e(P, Q) \) in \( G_T \)

Method:
1. \( f_1 = f_{c_0, Q}(P) \)
2. \( l_1 = l_{[p^3]Q, -[p^2]Q}(P) \)
5. return \( (f_1 * l_1 * l_2 * l_3)^{(p^k - 1)/r} \)

5.1. Straight Line Function over BN-curves

This subsection shows an operation of Straight Line Function over BN-curves for Optimal Ate Pairing.

Input:
- A point \( Q = (x_1, y_1) \) in \( G_2 \)
- A point \( Q' = (x_2, y_2) \) in \( G_2 \)
- A point \( P = (x, y) \) in \( G_1 \)

Output:
Method:
1. If \( Q \neq \pm Q' \)
   (A) \( \lambda = (y_2 - y_1)/(x_2 - x_1) \)
   (B) \( t_0 = -\lambda x \)
   (C) \( t_1 = \lambda x_1 - y_1 \)
   (D) \( l_n = y + t_0 w + t_1 w^3 \)
2. If \( Q = Q' \)
   (A) \( \lambda = (3 \times x_1^2)/(2 \times y_1) \)
   (B) \( t_0 = -\lambda x \)
   (C) \( t_1 = \lambda x_1 - y_1 \)
   (D) \( l_n = y + t_0 w + t_1 w^3 \)
   (E) return \( l_n \)
3. If \( Q = -Q' \)
   (A) \( l_n = x - x_1 w^3 \)
4. return \( l_n \)

5.2. Doubling Step of Miller Loop over BN-Curves

This subsection shows an operation of Doubling Step of Miller Loop over BN-curves. (i.e. operation of method 4-(A) in Section 4.2 over BN-curves)

Input:
- A point \( P = (x, y) \) in \( G_1 \)
- A point \( Q = (x_1, y_1) \) in \( G_2 \)

Output:
- \( l_n \) such that \( l_{\{Q, Q'\}}(P) \)
A point \( T = (x_3, y_3) \) such that \([2]Q\)

Method:
1. \( \lambda = (3 \cdot x_1^2)/(2 \cdot y_1) \)
2. \( x_3 = \lambda^2 - 2 \cdot x_1 \)
3. \( y_3 = \lambda \cdot (x_1 - x_3) - y_1 \)
4. \( t0 = -\lambda \cdot x \)
5. \( t1 = \lambda \cdot x_1 - y_1 \)
6. \( \ln = y + t0 \cdot w + t1 \cdot w^3 \)
7. return \( \ln \) and \( T \)

5.3. Addition Step of Miller Loop over BN-Curves

This subsection shows an operation of Addition Step of Miller Loop over BN-curves. (i.e. operation of method 4-(C)-(a) in Section 4.2 over BN-curves)

Input:
- A point \( Q = (x_1, y_1) \) in \( G_2 \)
- A point \( Q' = (x_2, y_2) \) in \( G_2 \)
- A point \( P = (x, y) \) in \( G_1 \)

Output:
- \( \ln \) such that \( l_{Q, Q'}(P) \)
- A point \( T = (x_3, y_3) \) such that \( Q + Q' \)

Method:
1. \( \lambda = (y_2 - y_1)/(x_2 - x_1) \)
2. \( x_3 = \lambda^2 - x_1 - x_2 \)
3. \( y_3 = \lambda \cdot (x_1 - x_3) - y_1 \)
4. \( t0 = -\lambda \cdot x \)
5. \( t_1 = \lambda * x_1 - y_1 \)
6. \( l_n = y + t_0 w + t_1 w^3 \)
7. return \( ln \) and \( T \)

6. Algorithm Identifiers

TBD

7. Security Considerations

The security of cryptographic primitive which is constructed by pairing depends on pairing-friendly curves (PFC). PFC must satisfy computational assumption which the primitive requires at the level of security strength in system when the primitive is constructed by using Optimal Ate Pairing.

8. Acknowledgements

TBD

9. Change log

NOTE TO RFC EDITOR: Please remove this section in before final RFC publication.

10. References

10.1. Normative References


10.2. Informative References


Appendix A. Test Vectors of Optimal Ate Pairing over BN-curves

In this section, we specify test vectors of optimal ate pairing over BN-curves which are specified by [7] in the following way.

Parameter:

Pairing-Param-ID is an identifier with which the pairing parameter set can be referenced.

Input:

P is a point of E in G_1
Q is a point of E' in G_2

Output:

e(P, Q) is computation of pairing in G_T

A.1. 254-Bit-Curves by Beuchat et al.

This subsection shows test vector of 254-bit curves by Beuchat et al. [7] and reprints its parameters under $F_{p^2} = F_p[u]/(u^2 + 5)$, $F_{p^6} = F_{p^2}[v]/(v^3 - u)$, $F_{p^{12}} = F_{p^6}[w]/(w^2 - v)$ as a reference.

Parameter:

Pairing-Param-ID: Beuchat

Input:

P = (0x0A971735A70FD0DF94D706EBB815EA78D292A8510F3344038A41641
9ADB9, 0x094D5621754237447752A448282C0873785F724447E1299826F53AC55
6936D3F)

Q = (0x115231D7B49901BA97CB93B5227F7F438A346532893DD5FAFD5189509
24AA9 + 0x0DF12398FB7695A50BB3499B7E23BD09035989B91A76D13AF7BC643
74BF8A86 u, 0x051D0E087527BC9F41379FB0272EC91EEF8EE011B183EF7D671
2EF3FC9A166 + 0x0107E6654DC6C36E163B7867AE5B98E4046084734524DBB56
2E73E5A811F678A u)

Output:

e(P, Q) = (0x06A4E0DD1F7FD2F9E5DACABA2B2ECC9CE8254925C5DC6697E153F05A
242CBA8A8 + 0x22AOE22CE97AEC1187087B7632C9B963BOE779BC8D09848C44D
3EA95CD1CF8C u + 0x0751037182B5F93BABC31B115A2C0A0DCC09C6DB7602E0
A.2.  254-Bit-Curves by Nogami et al. / Aranha et al.

This subsection shows test vector of 254-bit curves by Nogami et al. / Aranha et al. [7] and reprints its parameters under $F_{p^2} = F_p[u]/(u^2 + 1), F_{p^6} = F_{p^2}[v]/(v^3 - (1 + u)), F_{p^12} = F_{p^6}[w]/(w^2 - v)$ as a reference.

Parameter:

Pairing-Param-ID: Nogami-Aranha

Input:

\[
\begin{align*}
P &= (0x2074A81D4402A0B63B947335C14B2FC3C28FEA2973860F686114EC470 E4EB7, 0x06A4110808B20038771FC89F94A82B2006034A6EB871B3BC284846 631CBEB) \\
Q &= (0x049EEDB108B71A87BFCFC9B65EB5CF1C2F89554E02DF8354E4A05F21 83C77 + 0x1FB93AB676140E87D97226185B059EC088A9CC76966697CFBFA 9A9A845D u, 0x0CD04A1ED4AD3C6F6A1FE4453DA2BE96E7A679BF3FE825736 44C1C4D2F208A + 0x11FF7795CF59D1A17D6EE3C32D5F7615E1A94EA264 E71B7630A3C41C u)
\end{align*}
\]

Output:

\[
\begin{align*}
e(P, Q) &= (0x03E1F2693AC654989C87897EB158490A4832296F888D3014050 0DB7BD3D12 + 0x1EBEC4A768E44EB5D532945226BF103DE9CE14F6C698B7FAA6 6EF8BA79D3ED u + 0x0A5A5045542F67384D683A48C281F367B6755ED5A17 00784169A0B47A5E4 v + 0x048B66DAFCFAEE86DB46A71A9FE84843F814 F88B036A72B39698CF7201 u + 0x142715D6482BC6FA7737C9CB2A51C047C 16DE8683D5A889C7EF4DF5F03CBDB v^2 + 0x11EE0C12164133041C3DCF312CE1 11C45B6009281F7B7285D4AF61427934 u^2 + 0x22371A9F75DAE562F686 9BCDDB02702C595BBBF43A1FB3C7532D07BE37A3A w + 0x04052CA96090068 4A1B25C434B2776A70736847C416208CC1A7C2792E19 u + 0x05D259DA3 F33AAA54A6A55FBE8272A5B79D7FE5BDBF3B5E3C815AD781113F7548 v^2 + 0x084 3C37BC5BDBF253E3BCE568F5905A6386D7836855B74CBA0C800D5DC41B71 uuvv
\end{align*}
\]
A.3.  254-Bit-Curves by Scott

This subsection shows test vector of 254-bit curves by Scott [7] and reprints its parameters under $F_{p^2} = F_p[u]/(u^2 + 1)$, $F_{p^6} = F_{p^2}[v]/(v^3 - (1 + u))$, $F_{p^{12}} = F_{p^6}[w]/(w^2 - v)$ as a reference.

Parameter:

Pairing-Param-ID: Scott

Input:

$$P = (0x8a9143801f541142f89e498a1c06ba0959b8f9713abda0881e5de80d8af$$
$$f1a + 0x17df54e2be5e8afeb9a42f412825f79c32841307471fb2b6a1e3a0f$$
$$c6e010f4)$$

$$Q = (0x21794a9da7b34b2c1614315d7d90a282c484c8fd49c0c8b75b079ae304$$
$$7d566 + 0xa9b474c4519e6fae5b32c7cb65547d8707137bca00c9c182d10b7$$
$$e3e305936 u, 0xb00d54bf5a298d0eacdefb0efdb741a7e744722f61cc884488$$
$$fcce20ff876 + 0x8a9143801f541142f89e498a1c06ba0959b8f9713abda0881e5$$
$$de80d8af)$$

Output:

$$e(P,Q) = (0x13d3127ba07feff8c8c1a608aaf58a33a25148176968ef0ecc0a2e09$$
$$b62344f984 + 0x1774dfc7361e1d4cd2de4bf62cd9b460f0a78487e75994f9e25$$
$$51fed2f9d2b78 u + 0x2c7888f053123b5a815125b2c40e3f986594f6c35585c$$
$$fb1ed1a1cb2d2e6a5 v + 0xe7e7af51c459f6e0ef489348664bc4277e023a5031$$
$$bee9856b5b357c07d7e8 uv + 0x8d0f0d3f0d31d3624dd9e179233a12f2f2d13c$$
$$c1869f2eb3933c3c67d75e0d v^2 + 0x63e676f8cc5be53e8718cc9e61a8c5a$$
$$18a4c7e3a6f83f4c403ec8ca130e v^2u + 0x1643c66c66c5f4a1970bfe1a9$$
$$c5e3a312eb5c825f8d31354200d29339d2ca61 w + 0xa4c7e3a6f83f4c403ec8ca130e v^2u + 0x1643c66c66c5f4a1970bfe1a9$$
$$c5e3a312eb5c825f8d31354200d29339d2ca61 w + 0xa4c7e3a6f83f4c403ec8ca130e v^2u + 0x1643c66c66c5f4a1970bfe1a9$$

A.4. 254-Bit-Curves by BCMNPZ

This subsection shows test vector of 254-bit curves by BCMNPZ [7] and reprints its parameters under $F_{p^2} = F_p[u]/(u^2 + 1), F_{p^6} = F_{p^2}[v]/(v^3 - (1 + u)), F_{p^{12}} = F_{p^6}[w]/(w^2 - v)$ as a reference.

Parameter:

Pairing-Param-ID: BCMNPZ

Input:

$P = (0x1bec8eae1f1d3959e394588e49d09f2d3070efda1f836640288cf2af5488765 + 0x2d148d39f9edf5325d9a1f4820774930675669a6fe2084e35f4bfe3d3273c)$

$Q = (0xd62cf33cd0e46fdcc38cfab52ca5cdebf1a9348e44605441584ff4fadcf275 + 0x22701025e0cd2bfed4518febe8e7fa973c7f3f2fdd280e24651be9d17d7a8, 0x1cc6cb005535e7f83be0cfccf439d4687588fc21dc465a508c4f6cc1f90 + 86ee46779f9e922a870137d033e484ec5c5ba979b75bba179064a bff0cfa2a u)$

Output:

$e(P, Q) = (0x20f263ae42e42cfd53cf99dc238ed7b61951c1c767af88a72ad3c19ca54cd2d + 0XA96b263aade350f17201808028c4ce11793dd84127d80525fa57f829d3043f6 u + 0x3a31ca4864d996d64181d9a0b25e7368d60bf53a8276a2c39e02a58b6636e v + 0x2301fe7eb607ff6dd63b72797973c962d32fd487f1167764884f86a83c837174 uv + 0xcbe52ab6e1c210cf80215816f38d8964c45347bd3802c66d8s5e616ca9786d2e v^2 + 0x1c039dee75146d8ae6812568e77d11c fa06e01224dce628606bfb14090650 v^2u + 0x2344f2b5dd57710d54458383c33bd8f928babfe67f7641887a56579ob88e24 w + 0xe848a543ca273cca42811a2feae2e79eb3e628e27e54a477b5e1652466629608 wu+ 0x96a84564f586eed59d8a9397830824b885818e93a3ce4bfae057682efc37aebw + 0x17260fa31ed89d4e90d7a1a2652379e4329927e61f15b1a2ce2a93c84050245 wvu+ 0x5bd83369435b63ba10384db8248dab3908ef2173e166129d0cc6d357c89dce6 wv^2 + 0x2a4dec6bbfe98df2c9169b06410c329d4c699747ca649e611d9960416d15 b5 wv^2u)$

Authors' Addresses

Akihiro Kato
NTT Software Corporation

EMail: kato.akihiro-at-po.ntts.co.jp