COSE
Internet-Draft
Intended status: Standards Track
Expires: 8 May 2026

T. Looker
Mattr
M. Jones
Self-Issued Consulting

4 November 2025

Barreto-Lynn-Scott Elliptic Curve Key Representations for JOSE and COSE
draft-ietf-cose-bls-key-representations-08

Abstract

This specification defines how to represent cryptographic keys for the pairing-friendly elliptic curves known as Barreto-Lynn-Scott (BLS), for use with the key representation formats of JSON Web Key (JWK) and COSE (COSE_Key).

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/tplooker/draft-ietf-cose-bls-key-representations.

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 https://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 8 May 2026.

Copyright Notice

Table of Contents

1. Introduction

This specification defines how to represent cryptographic keys for the pairing-friendly elliptic curves known as Barreto-Lynn-Scott [BLS], for use with the key representation formats of JSON Web Key (JWK) and COSE_Key. This specification registers the elliptic curves in appropriate IANA JOSE and COSE registries.

There are a variety of applications for pairing based cryptography including schemes already published as RFCs, such as Identity-Based Cryptography [RFC5091] Sakai-Kasahara Key Encryption (SAKKE) [RFC6508], and Identity-Based Authenticated Key Exchange (IBAKE) [RFC6539]. SAKKE is applied to Multimedia Internet KEYing (MIKEY) via [RFC6509] and IBAKE is applied for a similar application via [RFC6267].

This branch of cryptography has also been used to develop privacy-preserving cryptographic hardware attestations schemes, including the Elliptic Curve Direct Anonymous Attestation (ECDAA) in the Trusted Platform Modules [TPM] specified by the Trusted Computing Group. Further work on similar schemes has also occurred at the FIDO Alliance [ECDAA]. Similarly, Intel released [EPID] which provides a solution to remote hardware attestation for Intel Software Guard Extension (SGX) enabled environments.

More recently, applications of pairing based cryptography using the Barreto-Lynn-Scott curves include the standardization effort for BLS Signatures [id.draft.bls-signature], which are used extensively in multiple blockchain projects due to their unique signature aggregation properties, including [Ethereum] [DFINITY] [Algorand]. Additionally, efforts are under way to standardize the general purpose short group signature scheme of BBS Signatures [BBS], which features novel properties such as multi-message signing and selective disclosure alongside zero knowledge proving. It is intended that this draft will help with these efforts by standardizing the associated cryptographic key representation in the popular formats of JWK and COSE_Key.

Other relevant work to this draft includes [JWP] which is extending the JOSE family of specifications to provide support for representing a variety of new proof based cryptographic schemes such as [BBS] which as referred to above uses the Barreto-Lynn-Scott curves.

There are multiple different pairing-friendly curves in active use; however, this draft focuses on a definition for the Barreto-Lynn-Scott curves due to them being the most "widely used" and "efficient" whilst achieving 128-bit and 256-bit security (BLS12-381 and BLS48-581 respectively).

More extensive discussion on the broader application of pairing based cryptography and the assessment of various elliptic curves (including the BLS family) can be found in [id.draft.pairing-friendly-curves].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Point Coordinates Encoding

A point representing a public key will either be in the G1 or G2 subgroup of a curve. Both are encoded using the compressed serialized point format defined normatively in Appendix B.2 of [BBS] and in Appendix C.

2.2. Representation Definition

The following definitions apply to the pairing-friendly elliptic curves known as the Barreto-Lynn-Scott (BLS) curves.

2.2.1. JSON Web Key Representation

When expressing a cryptographic key for these curves in JSON Web Key (JWK) form, the following rules apply:

* The parameter "kty" MUST be present and set to "OKP".
* The parameter "crv" MUST be present and value MUST be one defined in Section 2.2.3.
* The parameter "x" MUST be present with its value being the base64url encoding of the compressed serialized point format defined normatively in Appendix B of [BBS].
* The parameter "d" MUST be present for private key representations whose value MUST contain the big-endian representation of the private key base64url encoded without padding as defined in [RFC7515] Appendix C. This parameter MUST NOT be present for public keys.

2.2.2. COSE_Key Representation

When expressing a cryptographic key for these curves in COSE_Key form, the following rules apply:

* The parameter "kty" (1) MUST be present and set to "OKP" (1).
* The parameter "crv" (-1) MUST be present and value MUST be one defined in Section 2.2.3.
* The parameter "x" (-2) MUST be present with its value being the compressed serialized point format defined normatively in Appendix B of [BBS].

* The parameter "d" (-4) MUST be present for private key representations whose value MUST contain the big-endian representation of the private key. This parameter MUST NOT be present for public keys.

2.2.3. Curve Parameter Registration

JWK "crv" value	COSE_Key "crv" value	Description
BLS12381G1	TBD (13 requested)	A cryptographic key on the Barreto-Lynn-Scott (BLS) curve featuring an embedding degree 12 with 381-bit p in the subgroup of G1 defined as E(GF(p)) of order r. The private key will be 32 bytes long. The public key will be 48 bytes long.
BLS12381G2	TBD (14 requested)	A cryptographic key on the Barreto-Lynn-Scott (BLS) curve featuring an embedding degree 12 with 381-bit p in the subgroup of G2 defined as E(GF(p^2)) of order r. The private key will be 32 bytes long. The public key will be 96 bytes long.
BLS48581G1	TBD (15 requested)	A cryptographic key on the Barreto-Lynn-Scott (BLS) curve featuring an embedding degree 48 with 581-bit p in the subgroup of G1 defined as E(GF(p)) of order r. The private key will be 65 bytes long. The public key will be 73 bytes long.
BLS48581G2	TBD (16 requested)	A cryptographic key on the Barreto-Lynn-Scott (BLS) curve featuring an embedding degree 48 with 581-bit p in the subgroup of G2 defined as E(GF(p^8)) of order r. The private key will be 65 bytes long. The public key will be 584 bytes long.

Table 1

3. Security Considerations

See [id.draft.pairing-friendly-curves] for additional details on security considerations for the curves used. Implementers should also consider the general guidance provided in Section 9 of [RFC7517] and Section 17 of [RFC8152] when using this specification.

Furthermore, because this specification only defines the cryptographic key representations and not the usage of these keys with specific algorithms, implementers should be aware to follow any guidance that may be provided around appropriate usage of the keys and or additional steps that may be required to validate the keys within the context of particular algorithms.

4. IANA Considerations

4.1. JSON Web Key (JWK) Elliptic Curve Registrations

This section registers the following values in the IANA "JSON Web Key Elliptic Curve" registry [IANA.JOSE.Curves].

BLS12381G1

* Curve Name: BLS12381G1
* Curve Description: 381 bit with an embedding degree of 12 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E(GF(p))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.1

BLS12381G2

* Curve Name: BLS12381G2
* Curve Description: 381 bit with an embedding degree of 12 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E’(GF(p^2))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.1

BLS48581G1

* Curve Name: BLS48581G1
* Curve Description: 581 bit with an embedding degree of 48 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E(GF(p))
* JOSE Implementation Requirements: Optional

* Change Controller: IESG
* Specification Document(s): Section 2.2.1

BLS48581G2

* Curve Name: BLS48581G2
* Curve Description: 581 bit with an embedding degree of 48 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E'(GF(p^8))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.1

4.2. COSE Elliptic Curve Registrations

This section registers the following value in the IANA "COSE Elliptic Curves" registry [IANA.COSE.Curves].

BLS12381G1

* Curve Name: BLS12381G1
* Value: TBD (13 requested)
* Key Type: OKP
* Curve Description: 381 bit with an embedding degree of 12 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E(GF(p))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.2
* Recommended: Yes

BLS12381G2

* Curve Name: BLS12381G2
* Value: TBD (14 requested)
* Key Type: OKP
* Curve Description: 381 bit with an embedding degree of 12 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E'(GF(p^2))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.2
* Recommended: Yes

BLS48581G1

* Curve Name: BLS48581G1
* Value: TBD (15 requested)

* Key Type: OKP
* Curve Description: 581 bit with an embedding degree of 48 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E(GF(p))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.2
* Recommended: Yes

BLS48581G2

* Curve Name: BLS48581G2
* Value: TBD (16 requested)
* Key Type: OKP
* Curve Description: 581 bit with an embedding degree of 48 Barreto-Lynn-Scott pairing-friendly curve using the r-order subgroup of E’(GF(p^8))
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): Section 2.2.2
* Recommended: Yes

5. References

5.1. Normative References

[BLS] Barreto, P., Lynn, B., and M. Scott, "Constructing Elliptic Curves with Prescribed Embedding Degrees", 2003, <https://link.springer.com/chapter/10.1007/3-540-36413-7_19>.

[IANA.COSE.Curves] IANA, "COSE Elliptic Curves", <https://www.iana.org/assignments/cose/cose.xhtml#elliptic-curves>.

[IANA.JOSE.Curves] IANA, "JOSE Elliptic Curves", <https://www.iana.org/assignments/jose/jose.xhtml#web-key-elliptic-curve>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

5.2. Informative References

[BBS] IRTF CFRG, "The BBS Signature Scheme", 7 July 2025, <https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-09.html>.

[ECDAA] FIDO Alliance, "ECDAA Algorithm", 2018, <https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-ecdaa-algorithm-v2.0-id-20180227.html>.

[EPID] Intel Corporation, "Intel (R) SGX: Intel (R) EPID Provisioning and Attestation Services", <https://software.intel.com/en-us/download/intel-sgx-intel-epid-provisioning-and-attestation-services>.

[JWP] Miller, J., Waite, D., and M. Jones, "JSON Web Proof", 21 October 2023, <https://www.ietf.org/archive/id/draft-ietf-jose-json-web-proof-02.html>.

[RFC5091] Boyen, X. and L. Martin, "Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems", RFC 5091, DOI 10.17487/RFC5091, December 2007, <https://www.rfc-editor.org/info/rfc5091>.

[RFC6267] Cakulev, V. and G. Sundaram, "MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 6267, DOI 10.17487/RFC6267, June 2011, <https://www.rfc-editor.org/info/rfc6267>.

[RFC6508] Groves, M., "Sakai-Kasahara Key Encryption (SAKKE)", RFC 6508, DOI 10.17487/RFC6508, February 2012, <https://www.rfc-editor.org/info/rfc6508>.

[RFC6509] Groves, M., "MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)", RFC 6509, DOI 10.17487/RFC6509, February 2012, <https://www.rfc-editor.org/info/rfc6509>.

[RFC6539] Cakulev, V., Sundaram, G., and I. Broustis, "IBAKE: Identity-Based Authenticated Key Exchange", RFC 6539, DOI 10.17487/RFC6539, March 2012, <https://www.rfc-editor.org/info/rfc6539>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[TPM] Trusted Computing Group, "Trusted Platform Module", <https://trustedcomputinggroup.org/>.

[id.draft.bls-signature]
IRTF CFRG, "BLS Signatures", 16 June 2022, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-bls-signature-05>.

[id.draft.pairing-friendly-curves]
IRTF CFRG, "Pairing-Friendly Curves", 10 May 2023, <https://www.ietf.org/archive/id/draft-irtf-cfrg-pairing-friendly-curves-11.html>.

Appendix A. JSON Web Key Examples

A.1. BLS12381 Key Pairs

The following examples showcase JWKs for both the G1 and G2 subgroups of the BLS12381 curve. Note, the examples also include the corresponding private key, expressed through the inclusion of the "d" parameter.

An example JWK for the BLS12381 curve where the public key is in the G1 subgroup.

```
{
  "kty": "OKP",
  "crv": "BLS12381G1",
  "x": "iQ5g10FLC8VIX4b0jjN1ofvjStLU1tL0xN-_CpcHNPiQMT6qtk8hYBmbyevZWu5y",
  "d": "csnGswuvtF4lLJ5g7xdlFRbOKI1N7XaPhFdLZc408JU"
}
```

Another example of a different JWK for the BLS12381 curve where the public key is in the G1 subgroup.

```
{
  "kty": "OKP",
  "crv": "BLS12381G1",
  "x": "q6GrMMvlJ46PKeaj-IoTBtr9MDpORjmE8rQjUNOgsYXIBRYZhMn0XWCHdNWyZos_",
  "d": "H--QT8IQCXMHlYHEVBy6Z2yU4jENSPcmB6eVvcOWDHI"
}
```


An example JWK for the BLS12381 curve where the public key is in the G2 subgroup.


```
{
  "kty": "OKP",
  "crv": "BLS12381G2",
  "x": "pVD25M3Ca0jOBmHizej_YwuVOEIadk44urQcQQD3uhITsWj5LdgRmjTkftCme9KQ
EReUf5yoxPi7pDDx4UdkmTXtzuaIKm9YY2cOpT5dO26ttBSzneQEUFhHpM3sdUmf",
  "d": "XdXO0OOc6YrVTKEPIR6JmmTSDDA5Y5pxCyY5TRI0k5c"
}
```


Another example of a different JWK for the BLS12381 curve where the public key is in the G2 subgroup.


```
{
  "kty": "OKP",
  "crv": "BLS12381G2",
  "x": "o-w6GPtbZuiG7pEZ7Jelw925pirHQIunTOnLy-F68HSs3A2ejcukZFeYkyWOsVyI
DZKkES69mX0UBhUeyHI_DaZMv3YbSs_9Q1YxtJVn4uaneEykAFtTJyCSh2A6H1S7",
  "d": "MpN9MF6G6pmiZaJN6WOjWM2LQt07Blgb7WeJQbsKxWY"
}
```


A.2. BLS48581 Key Pairs

The following examples showcase JWKs for both the G1 and G2 subgroups of the BLS48581 curve. As before, note that the examples also include the corresponding private key, expressed through the inclusion of the "d" parameter.

An example JWK for the BLS48581 curve where the public key is in the G1 subgroup.


```
{
  "kty": "OKP",
  "crv": "BLS48581G1",
  "x": "jKj8Qmvi52Lky3VXrVaK7rEjW5lFBimGpicaEXPcsYrKzTjV5rRXYwtUog3QqY8Ub
aE7cGD2ppQXtR2KWfK6DpHWXy2HaGWS4g",
  "d": "EKF6v4ZUUDPLp52MWzpmTUg-S_-e01R08TcSH_wSUFkD4QteRbC13LEJE0W7aGJIV
i1BoLLaAcuTbJxwI_1qAbs"
}
```

   Another example of a different JWK for the BLS48581 curve where the public key is in the G1 subgroup.


```
{
  "kty": "OKP",
  "crv": "BLS48581G1",
  "x": "q9xqInwvGl6wXUITFFQMUUP4WKdVfmuTSS8gXsoe9ds1R78KR2xMoodMY9iTrWcD
eTYlOiFaRCxjMKhdgEwO3XMbKAnqPbUyQA",
  "d": "DKrY4HjY_A9pER9o0-YZ2AFq7VbNFEjsnXhGV6eKzgotb2cND-8E5bRb8zahSSAN
JqXHSTka7RAswU-8fprn0v0"
}
```


   An example JWK for the BLS48581 curve where the public key is in the G2 subgroup.


```
{
  "kty": "OKP",
  "crv": "BLS48581G2",
  "x": "g7cSrDeOkRJ5WXJMzb5OsLSWaAeVe8yXBxprZRTl9I8722A19NXCS8iR3xbTk-1V
am8dY4ZBV2TzeIWJT79GZNC2aTsup-WvSwqtB8gyafGtIXd0VSYkA3ApQosFTJoqro2vMk5Y
AuFKMvDzVaKR66zmCU6eLPeiWKiUsoOxV8g7VIo5p6Kxb9-wr6_MVEA2LfUMj1ecZXVf17XI
kvAt14iDUWVr8bouQOTGD00WS6o1dzPpacffZTd39285o4sNpFtOD2RRzHSm_imsYM7B9c5d
p0FAree2Om_hC6bYHBzdginDi84nOFCv1WzRb57Hy0tos8BY_3J-Tk70XS_7ren7OUyuQd1G
WLwhgMvHIFb7mxBOpxYTbP4_OYqNlDkpB4nA5A7lT5cjIbLHnj94Vn-HwFfLa6imE6zwt7Fa
IF9bwo3RUdwOn_Af75afZsxZ6xbDndOCxuHK0rDc8TQ9jZ4mFOqk6QNKl69dq8cQoc6eZnlJ
0DiD1_2QUGytR8PDK-a74mXrzfY51xifhq6bRVq1YdlnTk_afFMNIf6hLF2p_LcowHJ902--
0kUNUNzmYyYw2IoUCIjrvnJE2qwFKD6AseoQ18m3i1jeG6_i9KXT8QH9s-Wnp7hwv49wBIr2
dXOo-IB3T5jLm2hKm_w5e2GJRKabm5nSdr5L2YCGAwrupLy3vo2KFPUTX2evpzIaU4a455Ny
gWNka59tht-QcB4s5JK6X8h-m99Nbn4wgJAuHo2g9TSDyUK7l7UJVkx2rEckQfPj2fs",
  "d": "DMEAsp5YBiZvmzxnmZaA4baSfqc5-UK_tJBlJCP2_ig5ZEq0C7XAhI6jhHZX0y2H
XZUk9_y9QI_68dAwugoguh0"
}
```


   Another example of a different JWK for the BLS48581 curve where the public key is in the G2 subgroup.

```
{
  "kty": "OKP",
  "crv": "BLS48581G2",
  "x": "pYOX8QwBD32Rs4fvEGskWN5Fxbm1QYWDlGOIMfDSIBr1lJF0qj4UnKZngrHvjIQe
HSHfEjM8-1Z5xvjoehD7nDps0JEVQvr2eg4EEG5aRnL8F5uIo2QNdExwadKjjNu8tIvpjF_l
Bnoqy-RZyMGSPpIMxHnJMmO2VgtDYUft1WuVyrIjiFIDBSCnchYa734IV7MDbbyDofBnQl4L
F7Qn5mKt-r-WRmAf0gh_xEUW5d1D9XiLE3goIqfwiKUo4AoM2AkwQFCm6dImzJRAf5OMBwN3
U2uo-LeCMKXZDDsyARadT_zzbhDwYyiznjHssxONiukY25dXHQ1NOW_4ow0YI1O30a8KGAW-
n6SNU4eLIXo4U5blqpu4189proxjUemeE_To9QMqDaaxx-nr_Hz4kbE2FdVPESqlepLGcGpR
N7M_BVOZ-G_9wPiWbBbOjKy0rweMd9eEs7FAt1kHtMFNvur2c4rnWMF_p-aZs7ALT2aSl2tU
VVOZcSm25wHs6mllSOqMfCfw_aFiiOnd5AovXIAPJChH0lJL5b-Ji0-KpiOYA92x0w9P-JBu
9TEtJhkTgh9qcDb7BwC43BJvfoFfF-xPNwC2ZYR_8-zajuwRgZwrIQ11LIIVLOuOyeGdbPG7
JMdaJrTubV6iDxkx9x42zD_Nvb-f0FTbk_uYuxT3KBBGbmD9Zz54OFvHel41dmBtFiHqUtxy
bb3d71OHeZyvxu8b6LMZ22JpjVzRja1l95CiLAfBAdMDHyxwYE8a_4j1p0Zp7KyHGB8",
  "d": "CHIVGUCPsLY0GIx9DgOZlxmJHIWYrupsXtuKLZmFLCu5evIxwxrKo0edTXuch7uc
N437IDzp4P5-WKYtVcWFUrU"
}
```


Appendix B. COSE_Key Examples

B.1. BLS12381 Key Pairs

The following examples showcase COSE_Key examples for both the G1 and G2 subgroups of the BLS12381 curve. Note, the examples also include the corresponding private key, expressed through the inclusion of the "d" (-4) parameter.

An example COSE_Key for the BLS12381 curve where the public key is in the G1 subgroup expressed as an octet string.


```
a40101200d215830890e60d7414b0bc5485f86f48e3375a1fbe34ad2d4d6d2f4c4dfbf0a
970734f890313eaab64f2160199bc9ebd95aee7223582072c9c6b30bafb45e252c9e60ef
17651516ce288d4ded768f84574b65ce34f095
```


Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 13,
  -2 => h'890e60d7414b0bc5485f86f48e3375a1fbe34ad2d4d6d2f4c4dfbf0a970734f8
          90313eaab64f2160199bc9ebd95aee72',
  -4 => h'72c9c6b30bafb45e252c9e60ef17651516ce288d4ded768f84574b65ce34f095',
}
```

Another example of a different COSE_Key for the BLS12381 curve where the public key is in the G1 subgroup expressed as an octet string.

```
a40101200d215830aba1ab30cbe5278e8f29e6a3f88a1306dafd303a4e463984f2b42350
d3a0b185c805161984c9f45d608774d5b2668b3f2358201fef904fc2100973079581c454
1cba676c94e2310d48f72607a795bdc3960c72
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 13,
  -2 => h'aba1ab30cbe5278e8f29e6a3f88a1306dafd303a4e463984f2b42350d3a0b185
          c805161984c9f45d608774d5b2668b3f',
  -4 => h'1fef904fc2100973079581c4541cba676c94e2310d48f72607a795bdc3960c72',
}
```

An example COSE_Key for the BLS12381 curve where the public key is in the G2 subgroup expressed as an octet string.

```
a40101200d215860a550f6e4cdc26b48ce0661e2cde8ff630b9538421a764e38bab41c41
00f7ba1213b168f92dd8119a34e47ed0a67bd2901117947f9ca8c4f8bba430f1e1476499
35edcee6882a6f5863670ea53e5d3b6eadb414b39de404505847a4cdec75499f2358205d
d5ced0e39ce98ad54ca10f211e899a64d20c3039639a710b26394d12349397
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 13,
  -2 => h'a550f6e4cdc26b48ce0661e2cde8ff630b9538421a764e38bab41c4100f7ba12
          13b168f92dd8119a34e47ed0a67bd2901117947f9ca8c4f8bba430f1e1476499
          35edcee6882a6f5863670ea53e5d3b6eadb414b39de404505847a4cdec75499f',
  -4 => h'5dd5ced0e39ce98ad54ca10f211e899a64d20c3039639a710b26394d12349397',
}
```

Another example of a different COSE_Key for the BLS12381 curve where the public key is in the G2 subgroup expressed as an octet string.

```
a40101200d215860a3ec3a18fb5b66e886ee9119ec97a5c3ddb9a62ac7408ba74ce9cbcb
e17af074acdc0d9e8dcba464579893258eb15c880d92a4112ebd997d1406151ec8723f0d
a64cbf761b4acffd435631b49567e2e6a7784ca4005b5327209287603a1f54bb23582032
937d305e86ea99a265a24de963a358cd8b42dd3b06581bed678941bb0ac566
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 13,
  -2 => h'a3ec3a18fb5b66e886ee9119ec97a5c3ddb9a62ac7408ba74ce9cbcbe17af074
          acdc0d9e8dcba464579893258eb15c880d92a4112ebd997d1406151ec8723f0d
          a64cbf761b4acffd435631b49567e2e6a7784ca4005b5327209287603a1f54bb',
  -4 => h'32937d305e86ea99a265a24de963a358cd8b42dd3b06581bed678941bb0ac566',
}
```

B.2. BLS48581 Key Pairs

The following examples showcase COSE_Key examples for both the G1 and G2 subgroups of the BLS48581 curve. Note, the examples also include the corresponding private key, expressed through the inclusion of the "d" (-4) parameter.

An example COSE_Key for the BLS48581 curve where the public key is in the G1 subgroup expressed as an octet string.

```
a40101200e2158498ca8fc426be2e762e4cb7557ad568aeeb1235b9945062986a6271a11
73dcb18acacd38d5e6b457630b54a20dd0a98f146da13b7060f6a69417b51d8a59f2ba0e
91d65f2d87686592e223584110a17abf86545033cba79d8c5b3a664d483e4bff9ed35474
f137121ffc12505903e10b5e45b0b5dcb1091345bb686248562d41a0b2da01cb936c9c70
23fd6a01bb
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 14,
  -2 => h'8ca8fc426be2e762e4cb7557ad568aeeb1235b9945062986a6271a1173dcb18a
          cacd38d5e6b457630b54a20dd0a98f146da13b7060f6a69417b51d8a59f2ba0e
          91d65f2d87686592e2',
  -4 => h'10a17abf86545033cba79d8c5b3a664d483e4bff9ed35474f137121ffc125059
          03e10b5e45b0b5dcb1091345bb686248562d41a0b2da01cb936c9c7023fd6a01
          bb',
}
```

Another example of a different COSE_Key for the BLS48581 curve where the public key is in the G1 subgroup expressed as an octet string.

```
a40101200e215849abdc6a227c2f1a5eb05d421314540c5143f858a7557e6b93492f205e
ca1ef5db3547bf0a476c4ca2874c63d893ad67037936253a215a442c6330a85d804c0edd
731b2809ea3db532402358410caad8e078d8fc0f69111f68d3e619d8016aed56cd1448ec
9d784657a78ace0a2d6f670d0fef04e5b45bf336a149200d26a5c749391aed102cc14fbc
7e9ae7d2fd
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 14,
  -2 => h'abdc6a227c2f1a5eb05d421314540c5143f858a7557e6b93492f205eca1ef5db
          3547bf0a476c4ca2874c63d893ad67037936253a215a442c6330a85d804c0edd
          731b2809ea3db53240',
  -4 => h'0caad8e078d8fc0f69111f68d3e619d8016aed56cd1448ec9d784657a78ace0a
          2d6f670d0fef04e5b45bf336a149200d26a5c749391aed102cc14fbc7e9ae7d2
          fd',
}
```

An example COSE_Key for the BLS48581 curve where the public key is in the G2 subgroup expressed as an octet string.

```
a40101200e2159024883b712ac378e91127959724ccdbe4eb0b4966807957bcc97071a6b
6514e5f48f3bdb6035f4d5c24bc891df16d393ed556a6f1d6386415764f37885894fbf46
64d0b6693b2ea7e5af4b0aad07c83269f1ad21774552624037029428b054c9a2aae8daf
324e5802e14a32f0f355a291ebace6094e9e2cf7a258a894b283b157c83b548a39a7a2b1
6fdfb0afafcc5440362df50c8f579c65755fd7b5c892f02dd7888351656bf1ba2e40e4c6
0f4d164baa357733e969c7df653777f76f39a38b0da45b4e0f6451cc74a6fe29ac60cec1
f5ce5da74140ade7b63a6fe10ba6d81c1cdd8229c38bce273850afd56cd16f9ec7cb4b68
b3c058ff727e4e4ef45d2ffbade9fb394cae41dd4658bc2180cbc72056fb9b104ea71613
6cfe3f398a8d9439290789c0e40ee54f972321b2c79e3f78567f87c057cb6ba8a613acf0
b7b15a205f5bc28dd151dc0e9ff01fef969f66cc59eb16c39dd382c6e1cad2b0dcf1343d
8d9e2614eaa4e9034a97af5dabc710a1ce9e667949d03883d7fd90506cad47c3c32be6bb
e265ebcdf639d7189f86ae9b455ab561d9674e4fda7c530d21fea12c5da9fcb728c0727d
d36fbed2450d50dce6632630d88a140888ebbe7244daac05283e80b1ea10d7c9b78b58de
1bafe2f4a5d3f101fdb3e5a7a7b870bf8f70048af67573a8f880774f98cb9b684a9bfc39
7b618944a69b9b99d276be4bd98086030aeea4bcb7be8d8a14f5135f67afa7321a5386b8
e793728163646b9f6d86df90701e2ce492ba5fc87e9bdf4d6e7e3080902e1e8da0f53483
c942bb97b509564c76ac472441f3e3d9fb2358410cc100b29e5806266f9b3c67999680e1
b6927ea739f942bfb490652423f6fe2839644ab40bb5c0848ea3847657d32d875d9524f7
fcbd408ffaf1d030ba0a20ba1d
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 14,
  -2 => h'83b712ac378e91127959724ccdbe4eb0b4966807957bcc97071a6b6514e5f48f
          3bdb6035f4d5c24bc891df16d393ed556a6f1d6386415764f37885894fbf4664
          d0b6693b2ea7e5af4b0aad07c83269f1ad21774552624037029428b054c9a2a
          ae8daf324e5802e14a32f0f355a291ebace6094e9e2cf7a258a894b283b157c8
          3b548a39a7a2b16fdfb0afafcc5440362df50c8f579c65755fd7b5c892f02dd7
          888351656bf1ba2e40e4c60f4d164baa357733e969c7df653777f76f39a38b0d
          a45b4e0f6451cc74a6fe29ac60cec1f5ce5da74140ade7b63a6fe10ba6d81c1c
          dd8229c38bce273850afd56cd16f9ec7cb4b68b3c058ff727e4e4ef45d2ffbad
          e9fb394cae41dd4658bc2180cbc72056fb9b104ea716136cfe3f398a8d943929
          0789c0e40ee54f972321b2c79e3f78567f87c057cb6ba8a613acf0b7b15a205f
          5bc28dd151dc0e9ff01fef969f66cc59eb16c39dd382c6e1cad2b0dcf1343d8d
          9e2614eaa4e9034a97af5dabc710a1ce9e667949d03883d7fd90506cad47c3c3
          2be6bbe265ebcdf639d7189f86ae9b455ab561d9674e4fda7c530d21fea12c5d
          a9fcb728c0727dd36fbed2450d50dce6632630d88a140888ebbe7244daac0528
          3e80b1ea10d7c9b78b58de1bafe2f4a5d3f101fdb3e5a7a7b870bf8f70048af6
          7573a8f880774f98cb9b684a9bfc397b618944a69b9b99d276be4bd98086030a
          eea4bcb7be8d8a14f5135f67afa7321a5386b8e793728163646b9f6d86df9070
          1e2ce492ba5fc87e9bdf4d6e7e3080902e1e8da0f53483c942bb97b509564c76
          ac472441f3e3d9fb',
  -4 => h'0cc100b29e5806266f9b3c67999680e1b6927ea739f942bfb490652423f6fe28
          39644ab40bb5c0848ea3847657d32d875d9524f7fcbd408ffaf1d030ba0a20ba
          1d',
}
```

Another example of a different COSE_Key for the BLS48581 curve where the public key is in the G2 subgroup expressed as an octet string.

```
a40101200e21590248a58397f10c010f7d91b387ef106b2458de45c5b9b5418583946388
31f0d2201af5949174aa3e149ca66782b1ef8c841e1d21df12333cfb5679c6f8e87a10fb
9c3a6cd0911542faf67a0e04106e5a4672fc179b88a3640d744c7069d2a38cdbbcb48be9
8c5fe5067a2acbe459c8c1923e920cc479c93263b6560b436147edd56b95cab223885203
0520a772161aef7e0857b3036dbc83a1f067425e0b17b427e662adfabf9646601fd2087f
c44516e5dd43f5788b13782822a7f088a528e00a0cd809304050a6e9d226cc94407f938c
070377536ba8f8b78230a5d90c3b3201169d4ffcf36e10f06328b39e31ecb3138d8ae918
db97571d0d4d396ff8a30d182353b7d1af0a1805be9fa48d53878b217a385396e5aa9bb8
d7cf69ae8c6351e99e13f4e8f5032a0da6b1c7e9ebfc7cf891b13615d54f112aa57a92c6
706a5137b33f055399f86ffdc0f8966c16ce8cacb4af078c77d784b3b140b75907b4c14d
beeaf6738ae758c17fa7e699b3b00b4f6692976b545553997129b6e701ecea696548ea8c
7c27f0fda16288e9dde40a2f5c800f242847d2524be5bf898b4f8aa6239803ddb1d30f4f
f8906ef5312d261913821f6a7036fb0700b8dc126f7e815f17ec4f3700b665847ff3ecda
8eec11819c2b210d752c82152ceb8ec9e19d6cf1bb24c75a26b4ee6d5ea20f1931f71e36
cc3fcdbdbf9fd054db93fb98bb14f72810466e60fd673e78385bc77a5e3576606d1621ea
52dc726dbdddef5387799cafc6ef1be8b319db62698d5cd18dad65f790a22c07c101d303
1f2c70604f1aff88f5a74669ecac87181f235841087215194O8fb0b634188c7d0e039997
19891c8598aeea6c5edb8a2d99852c2bb97af231c31acaa3479d4d7b9c87bb9c378dfb20
3ce9e0fe7e58a62d55c58552b5
```

Below is the above CBOR represented as CDDL.

```
{
  1 => 1,
  -1 => 14,
  -2 => h'a58397f10c010f7d91b387ef106b2458de45c5b9b541858394638831f0d2201a
          f5949174aa3e149ca66782b1ef8c841e1d21df12333cfb5679c6f8e87a10fb9c
          3a6cd0911542faf67a0e04106e5a4672fc179b88a3640d744c7069d2a38cdbbc
          b48be98c5fe5067a2acbe459c8c1923e920cc479c93263b6560b436147edd56b
          95cab2238852030520a772161aef7e0857b3036dbc83a1f067425e0b17b427e6
          62adfabf9646601fd2087fc44516e5dd43f5788b13782822a7f088a528e00a0c
          d809304050a6e9d226cc94407f938c070377536ba8f8b78230a5d90c3b320116
          9d4ffcf36e10f06328b39e31ecb3138d8ae918db97571d0d4d396ff8a30d1823
          53b7d1af0a1805be9fa48d53878b217a385396e5aa9bb8d7cf69ae8c6351e99e
          13f4e8f5032a0da6b1c7e9ebfc7cf891b13615d54f112aa57a92c6706a5137b3
          3f055399f86ffdc0f8966c16ce8cacb4af078c77d784b3b140b75907b4c14dbe
          eaf6738ae758c17fa7e699b3b00b4f6692976b545553997129b6e701ecea6965
          48ea8c7c27f0fda16288e9dde40a2f5c800f242847d2524be5bf898b4f8aa623
          9803ddb1d30f4ff8906ef5312d261913821f6a7036fb0700b8dc126f7e815f17
          ec4f3700b665847ff3ecda8eec11819c2b210d752c82152ceb8ec9e19d6cf1bb
          24c75a26b4ee6d5ea20f1931f71e36cc3fcdbdbf9fd054db93fb98bb14f72810
          466e60fd673e78385bc77a5e3576606d1621ea52dc726dbdddef5387799cafc6
          ef1be8b319db62698d5cd18dad65f790a22c07c101d3031f2c70604f1aff88f5
          a74669ecac87181f',
  -4 => h'08721519408fb0b634188c7d0e03999719891c8598aeea6c5edb8a2d99852c2b
          b97af231c31acaa3479d4d7b9c87bb9c378dfb203ce9e0fe7e58a62d55c58552
          b5',
}
```

Appendix C. BLS48581 point encoding

Appendix B.2 of [BBS] defines point encoding and decoding procedures for BLS12-381. This section analogously extends the definition with encoding and decoding procedures for BLS48-581.

In this section we will use the notation defined in Appendix B.2 of [BBS] as well as the following notation,

* For an octet string x, x[i:j] will denote the substring beginning with the i-th octet and ending just before the j-th octet, where indices begin at 0. For example, x[0:3] denotes the first three octets (i.e., 24 most significant bits) of x.

We first have to define the following utility operations. sign_GF_p^8(y) returns one bit corresponding to the sign of an element in GF(p^8). The procedure sign_GF_p is defined in Appendix B.2 of [BBS].

```
   res = sign_GF_p^8(y)

   Inputs:

   - y (REQUIRED), point of the GF(p^8) group

   Outputs:

   - res, either 0 or 1

   Procedure:

   1. return sign_GF_p^8_i(y, 7)


   res = sign_GF_p^8_i(y, i)

   Inputs:

   - y (REQUIRED), point of the GF(p^8) group
   - i (REQUIRED), integer in the range [0, 7].
       Index of the component to evaluate next.

   Outputs:

   - res, either 0 or 1

   Procedure:

   1. (y_0, ..., y_i, ..., y_7) = y
   2. if i is 0, return sign_GF_p(y_0)
   3. if y_i is 0, return sign_GF_p^8_i(y_0, i - 1)
   4. return sign_GF_p(y_i)
```

C.1. Point Serialization

The point serialization procedure is the same as defined in Appendix B.2.1 of [BBS], with the following differences:

* The expression sign_GF_p^2(y) is replaced with sign_GF_p^8(y).
* The expression I2OSP(0, 48) is replaced with I2OSP(0, 73).
* The expression I2OSP(x, 48) is replaced with I2OSP(x, 73).
* The expression I2OSP(0, 96) is replaced with I2OSP(0, 584).
* Step 4 of the x_string definition is replaced with the following: If P is a point on E2 and P != Identity_E2, then let x_0, ..., x_7 elements of GF(p) such that x = (x_0, ..., x_7) and set x_string = I2OSP(x_7, 73) || ... || I2OSP(x_0, 73).

C.2. Point De-serialization

The point de-serialization procedure is the same as defined in Appendix B.2.2 of [BBS], with the following differences:

* The first two conditions in step 1 are:
  - If s_string has length 73 octets, the encoded point is on the curve E1.
  - If s_string has length 584 octets, the encoded point is on the curve E2.
* Step 4 is deleted.
* The following sub-step is added at the beginning of step 5:
  - Let x = OS2IP(s_string).
* The expression x^3 + 4 is replaced with x^3 + 1 in step 5.
* The following sub-steps are added at the beginning of step 6:
  - Let x_7, ..., x_0 = OS2IP(s_string[0:73]), OS2IP(s_string[73:146]), ..., OS2IP(s_string[511:584]).
  - Let x = (x_0, ..., x_7).
* The expression x^3 + 4 * (I + 1) is replaced with x^3 - 1 / w in step 6.

Appendix D. Acknowledgments

The authors would like to acknowledge the work of Kyle Den Hartog, which was used as the foundation for this draft. We would also like to thank Emil Lundberg and David Waite for their contributions to the specification.

Appendix E. Document History

-08

* Use ZCash compressed point format defined normatively in Appendix B of [BBS].
* Use "kty": "OKP" instead of "EC"/"EC2".
* Added Emil Lundberg to the acknowledgements.

-07

* Reference draft-irtf-cfrg-bbs-signatures-09.

-06

* Updated draft-irtf-cfrg-bbs-signatures reference.
* Made draft-irtf-cfrg-bls-signature and draft-irtf-cfrg-pairing-friendly-curves references informative.

-05

* Replaced instances of "Bls" with "BLS" since B., L., and S. are people’s initials, just like "RSA" is three people’s initials.

-04

* Changed the key type from OKP to EC/EC2 since these keys use "x" and "y" values.

-03

* Updated references.

-02

* Update COSE_Key and JWK examples.

-01

* Added JWK examples.

-00

* Created draft-ietf-cose-bls-key-representations-00 from draft-looker-cose-bls-key-representations-00 following working group adoption.

Authors’ Addresses

Tobias Looker
Mattr
Email: tobias.looker@mattr.global

Michael B. Jones
Self-Issued Consulting
Email: michael_b_jones@hotmail.com
URI: https://self-issued.info/